

# Contents

## Dynamics 365 Commerce

### Dynamics 365 Commerce

[Dynamics 365 Commerce home page](#)

[Get started](#)

[Before you buy](#)

[One Version service updates](#)

[One Version service updates overview](#)

[One Version service updates FAQ](#)

[Manage updates and the custom code lifecycle, including Dynamics 365 Commerce](#)

[Software lifecycle policy - Cloud](#)

[Software lifecycle policy - On-premises](#)

[Service update availability](#)

[Apply updates to cloud environments](#)

[Apply updates to on-premises deployments](#)

[Configure service updates](#)

[Pause service updates](#)

[Get notified about service updates](#)

[Data task automation](#)

[Delivering ISV solutions using One Version](#)

[Create and automate user acceptance tests](#)

[Set up a Dynamics 365 Commerce evaluation environment](#)

[Dynamics 365 Commerce evaluation environment overview](#)

[Provision a Dynamics 365 Commerce evaluation environment](#)

[Configure a Dynamics 365 Commerce evaluation environment](#)

[Configure BOPIS in a Dynamics 365 Commerce evaluation environment](#)

[Configure optional features for a Dynamics 365 Commerce evaluation environment](#)

[Dynamics 365 Commerce evaluation environment FAQ](#)

## Deployment considerations

- Deployment options

- Comparison of cloud and on-premises features

- Cloud deployment

- System requirements for cloud deployments

- On-premises deployment

- System requirements for on-premises deployments

- Installation steps for Retail channel components in an on-premises environment

- Buy Finance + Operations (on-premises)

- Commerce Data Exchange implementation guidance

- Finance and Operations apps operated by 21Vianet in China

- Finance and Operations apps in France

## Get support

### Upgrade and N-1 support

- Upgrade and N-1 support for Commerce

- Phased Rollout (N-1) installation, configuration, and cutover

- Upgrade the Retail channel extension to the latest Retail SDK

## What's new or changed

- What's new or changed in Finance and Operations apps

- What's new or changed in Commerce

  - What's new or changed in Commerce home page

  - Commerce 10.0.17 (April 2021)

  - Commerce 10.0.16 (February 2021)

  - Commerce 10.0.15 (January 2021)

- Removed or deprecated features

  - Removed or deprecated features in Finance and Operations

  - Removed or deprecated features in Dynamics 365 Commerce

- AX 2012 features that were postponed

- End of mainstream support for Microsoft Dynamics AX 2009 and 2012

## Documentation resources

- Help system

- Connect the Help system

[Custom help overview](#)

[View and export field descriptions](#)

[Task recorder](#)

[Create documentation or training with Task recorder](#)

[Videos](#)

[Glossary](#)

[Learning catalog](#)

[Learning catalog home page](#)

[Business decision makers](#)

[Business users](#)

[Implementation project managers](#)

[Administrators](#)

[Developers](#)

[Partner sales and marketing](#)

[Functional consultants](#)

[Solution architects](#)

[Explore the user interface](#)

[User interface elements](#)

[Accessibility features](#)

[Feature management overview](#)

[Client FAQ](#)

[Action search](#)

[Advanced filtering and query syntax](#)

[Configure and filter workspaces](#)

[Show pages side-by-side by using the Open in new window feature](#)

[Keyboard shortcuts](#)

[Change the banners or logo images for legal entities](#)

[Navigation search](#)

[Personalize the user experience](#)

[Saved views](#)

[Grid capabilities](#)

[Create and work with custom fields](#)

[Embed canvas apps from Power Apps](#)

[Find information by using lookups](#)

[Change the date for a session](#)

[Set a user's preferred time zone](#)

[Understand Lifecycle Services](#)

[Implementation lifecycle management](#)

[Implementation lifecycle management home page](#)

[Architecture for Finance and Operations apps](#)

[Microsoft FastTrack](#)

[Onboard an implementation project](#)

[Environment planning](#)

[Prepare for go-live](#)

[Go-live FAQ](#)

[Submit a service request](#)

[Subscriptions, LCS projects, and Azure AD tenants FAQ](#)

[Move environments between data centers](#)

[Move licenses between agreement types](#)

[Move LCS implementation projects to different Azure AD tenants](#)

[Move LCS implementation projects from on-premises to cloud](#)

[Multiple LCS projects and production environments on one Azure AD tenant](#)

[Implement Finance and Operations Commerce projects](#)

[Set up new environments, Azure DevOps, and branches for projects](#)

[Update code and environments for Commerce projects](#)

[Testing and performance issues](#)

[Architectural overview](#)

[Architecture](#)

[Dynamics 365 Commerce architecture overview](#)

[Dynamics 365 Commerce authentication flows](#)

[Dynamics 365 Commerce component versioning requirements](#)

[Commerce Scale Unit architecture](#)

[Select an in-store topology](#)

[Commerce Data Exchange](#)

Commerce Data Exchange best practices

Commerce Data Exchange and commerce channel communications

Commerce Data Exchange troubleshooting

Modern POS architecture

Online store publishing architecture

Retail Channel Performance Power BI content

Payment Application Data Security Standards (PA-DSS) certification

Commerce capabilities available in on-premises deployments

Deployment and hotfixes

Synchronize self-service installers in Dynamics 365 Commerce

Commerce Scale Unit (self-hosted)

Configure and install Commerce Scale Unit (self-hosted)

Modern POS configuration and installation

Manage accounts and devices from headquarters

POS device activation

Help secure Cloud POS in shared environments

Retail Hardware Station configuration and installation

Mass deployment of self-service components

Commerce component events for diagnostics and troubleshooting

Apply a deployable package

Initialize seed data in new Commerce environments

Set up your channels

Channels overview

Channel prerequisites

Channel setup prerequisites

General prerequisites

Organizations and organizational hierarchies

Warehouse setup

Sales tax setup

Set up an email notification profile

Number sequences

Set up a default customer and address book

## Retail channel prerequisites

- Info codes and info code groups

- Create a retail functionality profile

- Create a new address book

- Screen layout overview

## Online channel prerequisites

- Create an online functionality profile

## Set up a retail channel

- Set up a retail channel

- Configure retail workers

## Modern Point of Sale (MPOS) and Cloud POS

- Choose between MPOS and Cloud POS

- Online and offline POS operations

- Device activation of a customized Modern POS

- Enable Azure Active Directory authentication for POS sign-in

- Product and customer search in the POS

- Search for products and product variants during order entry

- Cash management improvements

- Inventory lookup in the POS

- Suspend and resume transactions in the POS

- Offline POS functionality

- Shift and cash drawer management

- Cash out gift card balance

- Ship orders from another store

- Hide non-carrier delivery modes from POS shipment dialog

- Task recorder and Help for MPOS and Cloud POS

- Hardware peripherals

- Screen layouts

- Sales associates

- Device activation

- Create financial dimensions for POS registers and configure dimension values on registers

- Create POS permissions group

- Create POS visual profiles

- Configure and install Retail hardware station

- Receipts

- Set up and design receipt formats

- Send email receipts from Modern POS (MPOS)

- Reset receipt numbers

- Embed processor credit card receipts in customer receipts

- Create functionality profiles for sales representatives

- Create financial dimensions for retail channels and configure values on stores

- Set up an online channel

- Set up an online channel

- Set up a call center channel

- Call center sales functionality

- Set up a call center channel

- Call center catalogs

- Set up call center order processing options

- Configure call center delivery modes and charges

- Set up and work with fraud alerts

- Additional channel configuration

- Add a channel to an organization hierarchy

- Advanced auto charges

- Configure and use advanced auto charges

- Enable and configure auto charges by channel

- Prorate header charges to matching sales lines

- Manage continuity programs

- Set up continuity programs for call centers

- Define continuity schedules

- Retail store management

- Create and change store hours

- Commerce Scale Unit

- Call center management

- Configure and work with call center order holds

## Payments

Payment methods

Payment methods in call centers

Refund payment processing in call centers

Credit card setup, authorization, and capture

Configure cash denominations for the POS

Configure credit card processing

Skip "change due" dialog box in POS

Adyen payment connector

Saving online payment instruments with the Adyen connector

Strong Customer Authentication (SCA) using the Adyen connector

Dedicated payment terminals and prompts for a printer and cash drawer

Seamless offline switch for gift card and credit memo operations

Set up pay invoice scenarios

Duplicate payments prevention

Omni-channel payments overview

Omni-channel Commerce order payments

Process credit cards without a hardware station

Dynamics 365 Payment Connector for PayPal

Wallet payment support

Payments FAQ

Manage Bing Maps for your organization

Create and update time slots for customer pickup

## Merchandising your products and services

Categories, products, and assortments

Product information overview

Set up retail products

Product hierarchy

Commerce hierarchies

Create a new product hierarchy

Manage product categories and products

Change the sort order for merchandising entities



## Product dimensions

- Product dimensions

- Create a variant group

## Create a new product

- Create a new product in Commerce

- Manage product categories and products

## Product attributes

- Attributes and attribute groups

- Define channel attributes

## Channel hierarchies

- Create a channel navigation hierarchy

- Configure channel to use a channel navigation hierarchy

## Assortments

- Assortment management

- Set up assortments

- Manage assortments

## Bar codes

- Set up bar codes

- Set up bar code masks

## Warranties

- Create and configure extended warranties

## Pricing and discounts

- Discounts

- Sales price management

- Price adjustments and discounts

- Apply multiple discounts to a product

- Determine the optimal combination of overlapping discounts

- Define channel-specific discounts

- Options for preventing discounts for retail products

- Tender-based discounts

- Shipping discount overview

- Set up coupons

Show discounts in POS

Price reports

Create channel-specific trade agreements

Create trade agreements using category pricing rules

## Recommendations

Product recommendations overview

Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment

Enable product recommendations

Personalized recommendations

- Enable personalized recommendations

- Opt out of personalized recommendations

Enable "shop similar looks" recommendations

Enable "shop similar description" recommendations

POS experiences

- Add product recommendations on POS

- Add recommendations to the transaction screen

Adjust AI-ML-based product recommendation results

Manually create curated product recommendations

Create recommendations using demo data

Product recommendations FAQ

## Ratings and reviews

Ratings and reviews overview

Opt in to use ratings and reviews

Manage ratings and reviews

Configure ratings and reviews

Sync product ratings

## Manage your inventory

Inventory

- Store inventory management

- Configure inventory buffers and inventory levels

- Work with serialized items in POS

- Push products from distribution center to store using buyer's push

Set up rules and parameters for cross-docking and buyer's push

Inbound inventory operation in POS

Outbound inventory operation in POS

Calculate inventory availability for retail channels

## Purchase orders

Purchase order overview

Create purchase orders

Product receipt against purchase orders

Approve and confirm purchase orders

Purchase order approval mobile workspace

Create product packages for purchase orders

## Manage your orders

### Customer orders

Customer orders in POS

Hybrid customer orders

Recall order operation in POS

Enable multiple pickup delivery modes

Process a customer order pickup in POS

### Sales orders

Create call center orders

Customize transactional emails by mode of delivery

Change mode of delivery in POS

### Distributed order management

DOM overview

Cost configuration

### Order fulfillment and processing

Store order fulfillment

Set up order fulfillment for stores

Show order notifications in POS

Sell continuity programs and process sales orders

### Returns

Linked refunds

- Sell and return products not included in a store's assortment
- Configure and process an exchange on a return order
- Restrict payment methods for returns without a receipt
- Return items across multiple customer orders and invoices
- Create and update a returns and refunds policy for a channel

## Manage your customers

### Clienteling

- Clienteling overview

### Loyalty

- Loyalty overview

- Define loyalty programs

- Define loyalty reward points

- Define loyalty schemes

- Process adjustments of loyalty reward points

## Manage your financials

### Statements overview

### Transaction consistency checker

- Disable rules in retail transaction consistency checker

### Edit and audit retail store transactions

- Edit and audit cash and carry and cash management transactions

- Edit and audit online order and asynchronous customer order transactions

- Edit financial dimensions for retail transactions

- Create an Excel workbook to edit retail transactions

- Add fields to an Excel workbook to edit retail transactions

### Trickle feed-based order creation for retail store transactions

- Create, calculate, and post statements for a retail store

### Financial reconciliation in retail stores

- Improvements to statement posting functionality

- Improved handling of batch tracked items

- Configure payment method settings that affect retail statements

- Configure and run a job to calculate statements

- Configure and run a job to post statements

[Configure store settings that affect retail statements](#)

[Posting of online sales and payments](#)

[Manage worker tasks](#)

[Task management overview](#)

[Configure task management](#)

[Create task lists and add tasks](#)

[Assign task lists to stores or employees](#)

[Task management in POS](#)

[Manage your e-commerce site](#)

[E-commerce site overview](#)

[Create an e-commerce site](#)

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user sign-ins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Domains in Dynamics 365 Commerce](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

[E-commerce digital gift cards](#)

[Browser requirements for Dynamics 365 Commerce sites](#)

[Sample site overview](#)

[Home page overview](#)

[Product details pages overview](#)

[Cart and checkout pages overview](#)

[Account management pages overview](#)

[Manage your site settings](#)

[Authoring page overview](#)

Manage users and roles

Apply inventory settings

Search engine optimization (SEO) considerations for your site

Manage Content Security Policy (CSP)

Opt out of web activity event collection

Manage your site content

Ways to add content

Page model glossary

Document states and lifecycle

Work with publish groups

Enable and use cross-channel sharing

Brand your site and add store details

Add a logo

Select a site theme

Work with CSS override files

Work with style presets

Add a favicon

Add a welcome message

Add a copyright notice

Add languages

Add script code to site pages to support telemetry

Modules

Work with modules

Fragments

Work with fragments

Templates

Templates and layouts overview

Work with templates

Work with preset layouts

Pages

Modify an existing site page

Add a new site page

Select page layouts

Manage SEO metadata

Save, preview, and publish a page

Enrich a product page

Enrich a category landing page

Verify page content accessibility

Create dynamic e-commerce pages based on URL parameters

Site navigation

Customize site navigation

Create a page URL

Manage digital assets

Digital asset management overview

Upload images

Upload video

Upload files other than images and videos

Crop images

Customize image focal points

Upload and serve static files

Make your site compliant

Compliance overview

Accessibility features and capabilities

Cookie compliance

Add a privacy policy page

Replace user IDs associated with tracked content changes

Module library

Module library overview

Container modules

Container module

Accordion module

Tab module

Iframe module

Marketing modules

Promo banner module

Carousel module

Text block module

Content block module

Video player module

Header and footer modules

Header module

Footer module

Product detail page modules

Buy box module

Media gallery module

Ratings and reviews modules

Social share module

Purchase modules

Cart module

Cart icon module

Checkout module

Payment module

Shipping address module

Delivery options module

Pickup information module

Order details module

Gift card module

Store finder modules

Map module

Store selector module

Account management modules

Account management pages and modules

Product collection modules

Product collection modules

Quick view module

Navigation modules



Breadcrumb module

Navigation menu module

Site selector module

Search modules

Search results module

Compliance modules

Cookie consent module

Search

Default category landing page and search results page overview

Cloud-powered search overview

Reporting

Generate online channel reports

Experimentation

Experimentation in Dynamics 365 Commerce

Identify a hypothesis and determine metrics for an experiment

Set up an experiment

Connect and edit an experiment

Preview and publish an experiment

Run and monitor an experiment

Promote a variation and complete an experiment

Review the status of an experiment

Error handling

Build custom response pages for 4xx/5xx status code errors

Fraud protection

Dynamics 365 Fraud Protection integration with Dynamics 365 Commerce

B2B e-commerce

Set up a B2B e-commerce site

Create org modeling hierarchies for B2B organizations

Manage business partner users on B2B e-commerce sites

Configure the customer account payment method for B2B e-commerce sites

Set product quantity limits for B2B e-commerce sites

Monitoring and analysis

Monitor sales and margin performance

Analyze sales trends and patterns

Assess sales performance by product

Assess customer and product profitability

Analyze store performance

Set up Recency, Frequency, and Monetary (RFM) analysis

## Globalization

Fiscal integration overview for Commerce channel

Set up fiscal integration for Commerce channel

User-defined certificate profiles for retail stores

## Austria

Fiscal registration service integration sample for Austria

## Brazil

Commerce localization for Brazil

## Eastern Europe

### Common

Advance invoices for Commerce for Eastern Europe

Petty cash management for Commerce for Eastern Europe

Customer invoices and return sales orders in Eastern European countries

Deployment guidelines for cash registers for Czech Republic, Hungary, and Poland

### Czech Republic

Fiscal registration service integration sample for Czech Republic

### Poland

Fiscal printer integration sample for Poland

Customer information management for Poland

## France

Cash register functionality

Deployment guidelines for cash registers

## Germany

Fiscal registration service integration sample for Germany

## India

Goods and Services Tax (GST) integration for cash registers

Deployment guidelines for cash registers

Support for upgrade and N-1 for India

## Italy

Fiscal printer integration sample for Italy

Customer information management for Italy

## Mexico

Global CFDI e-invoices for Mexico

## Norway

Cash register functionality

Deployment guidelines for cash registers

## Sweden

Cash register functionality

Control unit integration sample for Sweden

Sample for Retail POS integration with control units (legacy)

## Reporting and Power BI integrations

Reporting and analytics with Power BI

Information access and reporting

Personalization options with Power BI

Features available through the Power BI integration

Pin Power BI content

Select analytical workspaces from Power BI

## Power BI content

Power BI content home page

Learning

Organizational training

Retail channel performance

## Analytical workspaces

Analytical Workspaces (using Power BI Embedded)

Power BI Embedded integration

Customize embedded reports in Analytical workspaces

## Document reporting and printing

Document Reporting Services overview

Supported fonts

Configure SSRS for on-premises deployments

Printing

Document printing overview

Install the Document Routing Agent to enable network printing

Upgrade the Document Routing Agent

Install network printer devices in on-premises environments

Document generation, publishing, and printing in on-premises deployments

Run the Document Routing Agent as a Windows service

Available SQL Server Reporting Services reports

Install report design templates

Power user management tools

Power BI integration with Entity store

Resolve issues after Entity store maintenance

Configure the Power BI integration

Bring your own database

Preview PDF documents using a PDF viewer

Developer tools for creating and extending solutions

Analytics, aggregate measurements, and KPI modeling

Model aggregate data

Add financial dimensions to aggregate measurements

Create analytical reports using Power BI Desktop

Add analytics to workspaces by using Power BI Embedded

Help secure analytical workspaces and reports using Power BI Embedded

Create reporting solutions

Customize App Suite reports by using extensions

Expand Application Suite report data sets

Extend report menu items to redirect user navigation

Create custom designs for business documents

Help prevent long-running reports from timing out

Electronic reporting

Electronic reporting overview

## Configurations

- Create configurations

- Prepare components of the Electronic reporting solution

  - Design domain specific data model

  - Prepare application specific metadata

  - Test the prepared ER solution in the target application

  - Upgrade your format by adopting a new base version of that format

## Formula designers in Electronic reporting

- Formula designer

- Advanced formula designer

## Formula language and functions

- Formula language

- Functions

  - Date and time

  - List

  - Logical

  - Mathematical

  - Records

  - Text

  - Data collection

  - Other

  - Type conversion

- Manage ER model mapping in separate ER configurations

- Select data model definitions when you create formats

- Enhanced filtering options for finding configurations in the Global repository

- Generate reports by adding content as raw XML

- (ER) Cross-company data sources in Electronic reporting

- Design ER formats to generate outgoing electronic documents in various formats

  - (ER) Design a configuration for generating reports in OPENXML format

  - (ER) Design configurations to generate reports in Word format

  - Design configurations to fill out PDF templates

- Embed images and shapes

(ER) Embed images and shapes in documents that you generate

Example of embedding images and shapes

Design configurations to generate reports in Office formats that have embedded images

Review configurations to generate reports in Office formats that have embedded images

Generate reports in Office formats that have embedded images

Modify Electronic reporting formats by reapplying Excel templates

Define the dependency of ER configurations on other components

Design an ER solution to use financial dimensions

Design data models to use financial dimensions as data sources

Map models to use financial dimensions as data sources

Design reports to use financial dimensions as data sources

Run reports that use financial dimensions as data sources

Design ER formats to perform counting and summing by using information from generated output

Create formats to perform counting and summing

Configure format to perform counting and summing

Use computations to generate the output for counting and summing

Run formats to perform counting and summing

Dynamically add columns to Excel reports as horizontally expandable ranges

Design formats to dynamically add columns to Excel reports as horizontally expandable ranges

Run formats to dynamically add columns to Excel reports as horizontally expandable ranges

Use Document management files in format outputs

Prepare data models to use Document management files in ER outputs

Extend data models to use Document management files in ER outputs

Create formats to use Document management files in ER outputs

Run formats to use Document management files in ER outputs

Modify and run formats to use Document management files in ER outputs

Update application data using Electronic reporting

Generate electronic documents and update application data

Import configurations to generate documents with application data

- Design configurations to generate documents with application data
- Modify models and mappings to generate documents with application data
- Modify formats to generate documents with application data
- Generate documents that have application data

Import data from incoming files by using ER solutions

- Parse incoming documents in different formats
- Configure data import from manually selected files
  - Create configurations to import data from an external XML file
- Configure data import in batch mode from SharePoint located files
- Import files in XML format
  - Import files in XML format with optional attributes
  - Configure ER formats to import XML format files with optional attributes

Split generated XML files based on file size and content quantity

Support parameterized calls of ER data sources of Calculated field type

Configuration lifecycle

- Manage configurations lifecycle
- Upload configurations into lifecycle Services
- Import configurations from Lifecycle Services
- Download configurations from Lifecycle Services
- Import configurations
  - Download ER configurations from the RCS repository
    - Download configurations from the ER repository of the RCS type
  - Import configurations from RCS

Tutorials

- Use JOIN data sources to get data from multiple application tables
- Configure country context dependent ER model mappings
- Configure ER formats to use parameters that are specified per legal entity
- Set up the parameters of an ER format per legal entity
- Defer execution of ER format elements
  - Defer the execution of sequence elements in ER formats
  - Defer the execution of XML elements in ER formats

Use Electronic reporting solutions in Finance and Operations

Configure the framework

- Configure the Electronic reporting framework

- Back up and recover ER templates

- Create configuration provides and mark them as active

Destinations

- Destinations overview

- Destination types

- Configure destinations

Set up a reference to an ER format

Automatically validate generated reports

- Trace generated report results and compare them with baseline values

- Improvements in tracing results of generated ER reports and comparisons with baseline values

- Automate testing with Electronic reporting

Automate business processes by using Electronic reporting configurations

- Configure Electronic reporting to pull data into Power BI

- Generate printable free text invoice (FTI) forms

Trace execution of ER formats to troubleshoot performance issues

Business document management

- Business document management overview

- New document user interface in Business document management

- Add new fields to a business document template in Microsoft Excel

ER framework API

- Extend the list of Electronic reporting functions

- Electronic reporting framework API changes for Application update 7.3

- Specify a custom storage location for generated documents

Finance and Operations (Dynamics 365) mobile app

Mobile app home page

Mobile workspaces

- Available mobile workspaces

- Asset management

- Company directory

- Cost controlling



[Expense management](#)

[Inventory on-hand](#)

[Invoice approvals](#)

[My team](#)

[Project mobile applications overview](#)

[Project time entry](#)

[Project timesheet](#)

[Purchase order approval](#)

[Sales orders](#)

[Vendor collaboration](#)

[Publish mobile workspaces](#)

## [Office integration](#)

[Office integration overview](#)

[Office integration tutorial](#)

[Open entity data in Excel and update it using the Excel add-in](#)

[Create Open in Excel experiences](#)

[Add templates to the Open lines in Excel menu](#)

[Customize the Open in Microsoft Office menu](#)

[Configure and send email](#)

[Create email templates for transactional events](#)

[Develop email experiences](#)

[Troubleshoot the Office integration](#)

## [Organization administration](#)

[Organization administration home page](#)

[Number sequences](#)

[Number sequences overview](#)

[Set up number sequences on an individual basis](#)

[Set up number sequences using a wizard](#)

## [Organizations and organizational hierarchies](#)

[Organizations and organizational hierarchies overview](#)

[Plan your organizational hierarchy](#)

[Create an organization hierarchy](#)

Create legal entities

Create operating units

Global address book

Global address book overview

Plan for the global address book and other address books

Configure the global address book

Address books FAQ

Configure address books

Workflow system

Workflow system overview

Workflow system architecture

Workflow elements

Workflow actions

Create workflows

Create workflows overview

Configure workflow properties

Configure manual tasks

Configure automated tasks

Configure approval processes

Configure approval steps

Configure manual decisions

Configure conditional decisions

Configure parallel activities

Configure parallel branches

Configure line-item workflows

Configure the Workflow Message Processing batch job as critical

Delegate work items in a workflow

View workflow history

Configure the system to send workflow-related email to users

Workflow types

Creating a workflow type

Workflow type checklist

[Create a workflow category](#)

[Create a query for a workflow type](#)

[Create a new workflow type](#)

[Create a workflow document class](#)

[Create a SubmitToWorkflow class](#)

[Associate a workflow document class with a workflow type](#)

[Workflow types report](#)

[Workflow FAQ](#)

## [Alerts](#)

[Alerts overview](#)

[Create alert rules](#)

[Batch processing of alerts](#)

[Client alert notifications by email](#)

## [Electronic signatures](#)

[Electronic signatures overview](#)

[Set up electronic signatures](#)

## [Case management](#)

[Case management overview](#)

[Plan case security, processes, and categories](#)

## [Record templates](#)

[Record templates overview](#)

[Create a record template](#)

[Use a record template to create a new record](#)

## [Configure document management](#)

## [Configure and send email](#)

## [Date/time data and time zones](#)

## [Extensibility](#)

### [Commerce development and extensibility](#)

### [Online channel extensibility](#)

[Online channel extensibility overview](#)

[Get started](#)

[Get started with e-commerce online extensibility development](#)

System requirements

Set up a development environment

Configure a development environment (.env) file

Configure an e-commerce development environment against a Commerce cloud environment

Set up Azure DevOps code sharing and create a build pipeline

Platform overview

E-commerce architectural overview

E-commerce components

CLI command reference

Anatomy of a module

Modules overview

Module definition file

Module React component file

Module view file

Module data file

Module mock file

Module test file

Module props.autogenerated.ts file

Develop modules

Create a new module

Clone a module library module

Add module configuration fields

Preview and debug modules

Test modules with module mocks

Test modules by using page mocks

Container modules

Create a layout container module

Create a page container module

Localize a module

Advanced module topics

Request properties object

App settings

[Platform settings file](#)

[Extend a module definition](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Mock the signed-in state during local development](#)

[Configure module properties to be shown based on context](#)

[Globalize modules by using the CultureInfoFormatter class](#)

[Set up Azure Key Vault for secure key management](#)

[Develop data actions](#)

[Data actions overview](#)

[Data action cache options](#)

[Test data actions with mocks](#)

[Page load data actions](#)

[Event-based data actions](#)

[Core data actions](#)

[Call Retail Server APIs](#)

[Call Retail Server extension APIs](#)

[Advanced data action topics](#)

[Chain data actions](#)

[Batch data actions](#)

[Observable data actions](#)

[Share state across modules](#)

[Data action cache settings](#)

[Data action overrides](#)

[Data action hooks](#)

[Develop themes](#)

[Theming overview](#)

[Create a theme](#)

[Configure theme settings](#)

[Configure theme style presets](#)

[Extend a theme to add module extensions](#)

[Override a module library component in a theme](#)

[Extend a theme from a base theme](#)

[Add custom resources to your customization code](#)

[Inject script](#)

[Script injectors](#)

[Connectors](#)

[Configure and enable connectors](#)

[Telemetry and analytics](#)

[Telemetry logger](#)

[Deployment and maintenance](#)

[Package configurations and deploy them to an online environment](#)

[SDK and module library updates](#)

[Best practices and FAQ](#)

[Best practices for Dynamics 365 Commerce development](#)

[Dynamics 365 Commerce online SDK FAQ](#)

[Retail SDK](#)

[Retail SDK architecture](#)

[Upcoming changes in the Retail SDK](#)

[Development and ALM changes from version 10.0.10 to 10.0.13](#)

[Merge the build systems for Commerce and Finance and Operations](#)

[Migrate the Retail SDK from Visual Studio 2015 to Visual Studio 2017](#)

[Retail SDK extensibility samples](#)

[Create deployable packages](#)

[Development in cloud-hosted environments without admin access](#)

[Set up Commerce SDK build pipeline](#)

[Download samples and packages from GitHub and NuGet](#)

[Retail SDK FAQ](#)

[Point of sale extensibility](#)

[POS screen layouts](#)

[Open URL in POS](#)

[Install the POS Layout designer](#)

[Extend the POS Dual display view](#)

[Add custom controls to POS views](#)

Add POS operations to POS layouts using Button grid designer

POS APIs

POS payment extension

Support for tipping in the POS payments SDK

Support for external gift cards

Create and apply branding to the Retail Experience app

Set up POS Hybrid app on Android and iOS

Override POS request handler

View POS extension package information

Test recorder and Regression suite automation tool for Cloud POS

Add custom fields to the POS Totals panel

Show custom notifications in POS

Sign MPOS with code signing certificate

Extend POS views to add custom columns and app bar buttons

Add custom controls to Modern POS transaction pages

Add custom columns to a POS transaction grid

Call POS APIs or operations from POS extensions

Modern POS (MPOS) triggers and printing

Add custom controls to non-screen designer-based POS views

Run the POS samples

Device management implementation guidance

POS Cart view events and handlers

Add custom buttons to the POS header bar

Hardware Station extensibility

Hardware Station and extensibility

Integrate POS with a new hardware device

Retail Server extensibility

Commerce Scale Unit customer and consumer APIs

Create Retail Server extension API (Retail SDK version 10.0.11 and later)

Create new Retail Server extension API (Retail SDK version 10.0.11 and earlier)

Retail Typescript and C# proxies

Consume Retail Server APIs in external applications

## Commerce Runtime (CRT) extensibility

CRT architecture and configuration

CRT services

CRT extension pattern

Manage secrets for Retail channels

CRT extensibility and triggers

Block transactions using triggers

Loyalty extension

Extension points for packing slips

Channel database extensions

Enable custom CDX synchronization via extension

Define and set order attributes

Customer attributes

CRT SDK samples

Log extension events to Application Insights

Create async Commerce (CRT) APIs in your business logic

Pre-extended columns in the channel database

Extend Commerce Data Exchange - Real-time Service

## Payment connector extensibility

Create a payment integration for a payment terminal

Implementing a payment connector and payment device (white paper)

Deploy payment connectors

Incremental capture for back-office invoicing

Create custom localized error messages for payment terminal extensions

Create Windows installers for payment connectors

Payment data use

Enable duplicate payment protection for payment connector

Create Commerce payment packaging for Finance and Operations deployment

## Third-party e-Commerce extensibility

Configure online stores

Configure authentication providers

e-Commerce platform SDK



Localize Commerce extension resources and label files

Finance and Operations apps development and administration

Dynamics 365 Finance

Dynamics 365 Supply Chain Management

# Commerce home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

Dynamics 365 Commerce—built on the proven Dynamics 365 Retail capabilities—delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.

This release enables the creation of digital experiences using built-in web authoring and development tools to produce engaging and intelligent digital storefronts. A connected marketing and headless commerce platform further enable the seamless management of content, assets, promotions, inventory, and pricing across all channels.

- **Everything to build and run digital commerce:** Streamline your business and end-to-end commerce solution that scales to your needs across traditional and emerging channels. Built-in web authoring and development tools enable you to create engaging intelligent digital storefronts, while a connected marketing and headless commerce platform enables seamless management of content, assets, promotions, inventory, and pricing across channels.
- **Build loyalty and exceed customer expectations:** Use clienteling tools to gain a comprehensive view of your customer and respond to their needs at every level of engagement, based on customer profile, history, and preferences that flow across physical and digital channels. Empower your employees to foster lasting relationships through AI-driven recommendations, customer insights, and loyalty programs that elevate brand appeal.
- **Flexible and intelligent omnichannel experience:** Unify physical and digital commerce by providing consistent experiences to customers across cloud search and discovery, product reviews, wish lists, inventory, gift cards, and loyalty. Allow customers to purchase when, how, and where they want, on any device—while providing choice around modern payment methods and product collection or delivery.
- **Streamline operations using AI in the cloud:** Drive omnichannel commerce experiences and integrated, optimized back-office operations through ingrained, pervasive, and context-aware cloud intelligence. Use advanced merchandising, inventory management, distributed order management, and pricing and promotion to innovate and stay ahead of competition. Derive insights by visualizing and analyzing comprehensive and consistent data across all aspects of your business. Use AI-driven technologies to provide accessible websites, protect your business against payment fraud, and efficiently moderate user-generated content like ratings and reviews.

## Core concepts and tasks

Select a feature area to learn more about it.

- [Configure a Commerce preview environment](#)
- [Commerce architecture](#)
- [Set up your channels](#)
- [Merchandising your products and services](#)
- [Manage your orders](#)
- [Manage your customers](#)
- [Manage your financials](#)

- [Manage your e-Commerce site](#)
- [Fraud protection](#)
- [Commerce development and extensibility](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Before you buy

2/18/2021 • 3 minutes to read • [Edit Online](#)

New to Dynamics 365 Finance, Dynamics 365 Supply Chain Management, or Dynamics 365 Commerce? We've put together step-by-step guidance whether you're still evaluating or ready to make a purchase.

## Step one: Try out Finance and Operations free for 30 days

You can try Dynamics 365 for Finance, Dynamics 365 Supply Chain Management, or Dynamics 365 Commerce for 30 days through a simple email signup. The trial version of Finance and Operations applications includes Getting started task guides that provide step-by-step instructions that allow you to view specific scenarios in action. The product is available to explore and exercise scenarios, but cannot be customized. Demo data is included to ease the use of the product and to make the experience more meaningful. A reminder email will be sent 3 days prior to the trial expiration. Get details at [Sign up for preview subscriptions](#).

## Step two: Choose a deployment option

You can now deploy Finance and Operations applications in the cloud or on-premises. Cloud deployments offer an ERP service that is fully managed by Microsoft, while on-premises deployments are deployed locally within a customer's data center.

The following considerations must be taken when you choose on-premises as a deployment option:

- Regulatory and compliance needs that are not available in the cloud certifications.
- Disconnected business process with intermittent internet connectivity required to access Microsoft Dynamics Lifecycle Services (LCS) for application lifecycle management.
- [Comparison of cloud and on-premises features](#).
- [System requirements for on-premises deployments](#).

### IMPORTANT

On-premises deployments are not supported on any public cloud infrastructure, including Microsoft Azure.

For more information, see [Deployment options](#).

## Step three: Buy and manage a subscription

To explore subscription options, go to the [Dynamics 365 pricing](#) page. This page includes several different plans to fit your organization's needs.

There are many ways to buy a subscription:

- Buy through a Cloud solution provider (cloud only).
- Buy through a partner, and use volume licensing (cloud or on-premises).
- Buy through a partner from the Dynamics price list (on-premises only).

### **Buy through a Cloud solution provider (cloud-only)**

A Microsoft Cloud Solution Provider can work closely with you to understand the needs of your business or organization. Use the [Microsoft Partner Center portal](#) to find a partner to fit your needs.

### **Buy through a Dynamics partner (on-premises)**

You must work with a partner to purchase Finance and Operations on-premises. For more information, see [Buy Finance + Operations \(on-premises\)](#).

### **Buy through volume licensing (cloud or on-premises)**

If your organization has 250 or more Dynamics 365 users, you may be interested in a [Volume licensing agreement](#).

In volume licensing, Finance and Operations applications are available through:

- Enterprise Agreement
- Enterprise Agreement Subscription
- Enrollment for Education Solutions (under the Campus and School Agreement)
- Microsoft Products and Services Agreement (MPSA)

### **Choose your support option**

Microsoft provides flexible, industry-leading support, services, and resources that enable users to quickly address technical issues and maximize return on your Dynamics 365 investment. Choose a plan that best meets your business requirements.

For additional resources, see:

- [Dynamics 365 Support](#)
- [Quick Start Guide for Microsoft Dynamics Cloud Service Support Benefits](#) (PDF)

## **Step four: Learn about FastTrack and plan your deployment**

Microsoft FastTrack for Dynamics 365 is our customer success service designed to help you move to Dynamics 365 smoothly and confidently, so you can realize business value faster. When you participate in the FastTrack program, you will receive guidance on best practices and how to plan for successful rollouts. You will also learn ways to enable new users and expand capabilities – all at your own pace. Additionally, you will have access to Microsoft engineering resources committed to make your experience with Dynamics 365 a success. For more information, see [Microsoft FastTrack](#).

## **If you are upgrading from Dynamics AX 2012 or migrating from AX 2009**

If you are a customer who is upgrading from Microsoft Dynamics AX 2012 or migrating from Microsoft Dynamics AX 2009, you may be eligible for a longer trial. Contact [daxcf@microsoft.com](mailto:daxcf@microsoft.com) for more information.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# One Version service updates overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following set of topics provide information that is related to service updates for Microsoft Dynamics 365 for Finance and Operations version 8.1 (October 2018) and later. This is applicable for cloud releases only.

- [Service update availability](#) – This topic provides information about the release cadence and release process.
- [Software lifecycle policy and cloud releases](#) – This topic provides information about the service updates, availability, and end of service.
- [One Version service updates FAQ](#) – This topic answers questions about the update process, tools, planning, and Retail service updates.

The experience for service updates consists of four distinct steps:

1. Configure
2. Notice
3. Update
4. Validate

The rest of this topic describes each step and provides links to related topics.

## Configure

Customers can select a maintenance window, based on their business constraints. In Microsoft Dynamics Lifecycle Services (LCS), use the fields in the **Production environment update cadence** section on the **Update settings** tab of the **Project settings** page, as shown in the following image. A calendar of upcoming updates is available to help you plan ahead.

The screenshot shows the 'Project settings' page in Microsoft Dynamics Lifecycle Services. The 'Update settings' tab is selected. The page is divided into two main sections: 'SANDBOX UPDATE ENVIRONMENT' and 'PRODUCTION ENVIRONMENT UPDATE CADENCE'. In the 'SANDBOX UPDATE ENVIRONMENT' section, a dropdown menu shows 'app81servicingsb2 (Selected)'. In the 'PRODUCTION ENVIRONMENT UPDATE CADENCE' section, there are four dropdown menus: 'Cadence' (Second week (Selected)), 'Time Zone' (Eastern Time (UTC - 5) (Se...)), 'Day of the week' (Saturday (Selected)), and 'Time Slot' (4 AM to 7 AM (Selected)). Below these settings, there is a 'No' checkbox and a 'Save' button. A link to 'View the update calendar' is also present.

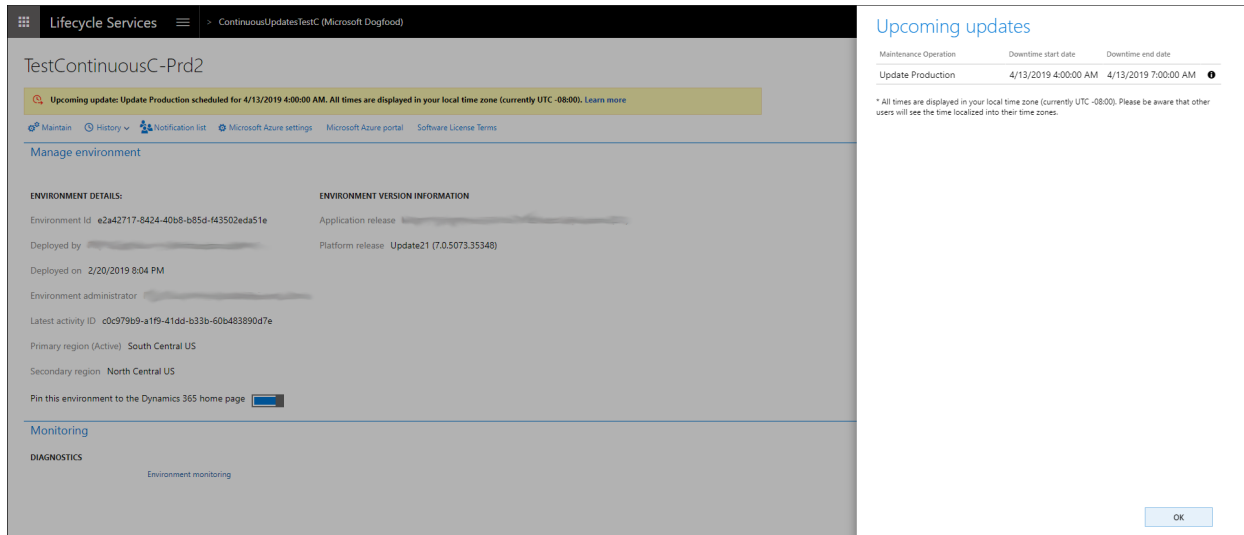
Users must opt in to new features and turn them on. All updates are applied first to the user acceptance testing (UAT) environment and then to the production environment. Therefore, customers have time to do any

validation that is required. Customers can select the environment that is updated. They can also pause an update for up to three months.

## Notice

[Release plans](#) will be available to help you plan ahead and understand what is changing. You can learn about upcoming features up to three months in advance. The [What's new](#) topics provide details about the updates for specific months.

Additionally, a notification email will be sent five days in advance, and a notification will appear in LCS just before an update, as shown in the following illustration.

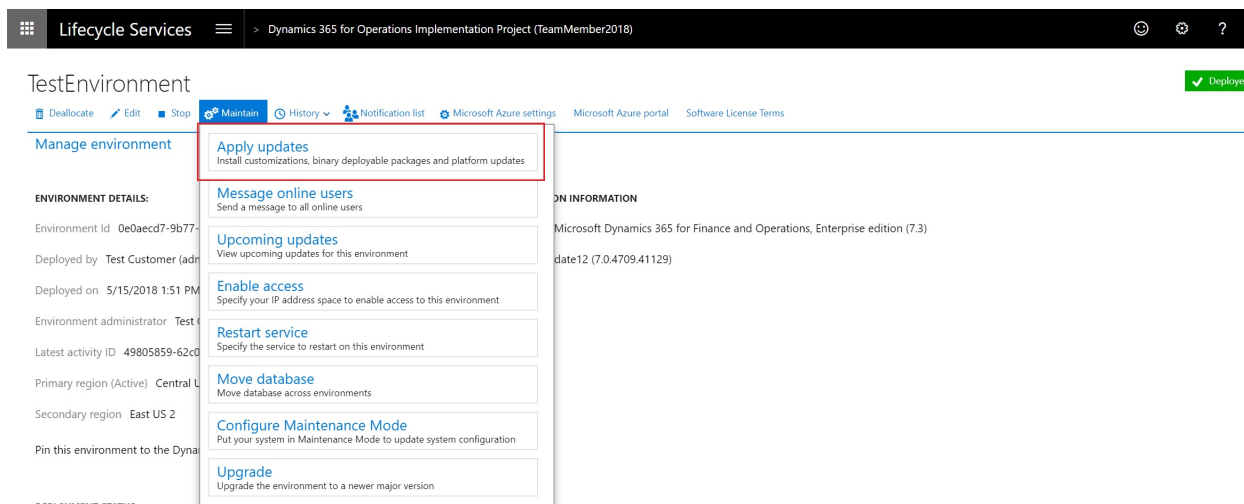


## Update

After notifications have been sent, Microsoft will apply the update (**auto update**) during the designated maintenance window. After this operation is completed, a notification email will be sent to indicate the status of the update. Customers will also be able to **self-update** by using the standard update experience in LCS. For more information, see [Apply updates to cloud environments](#).

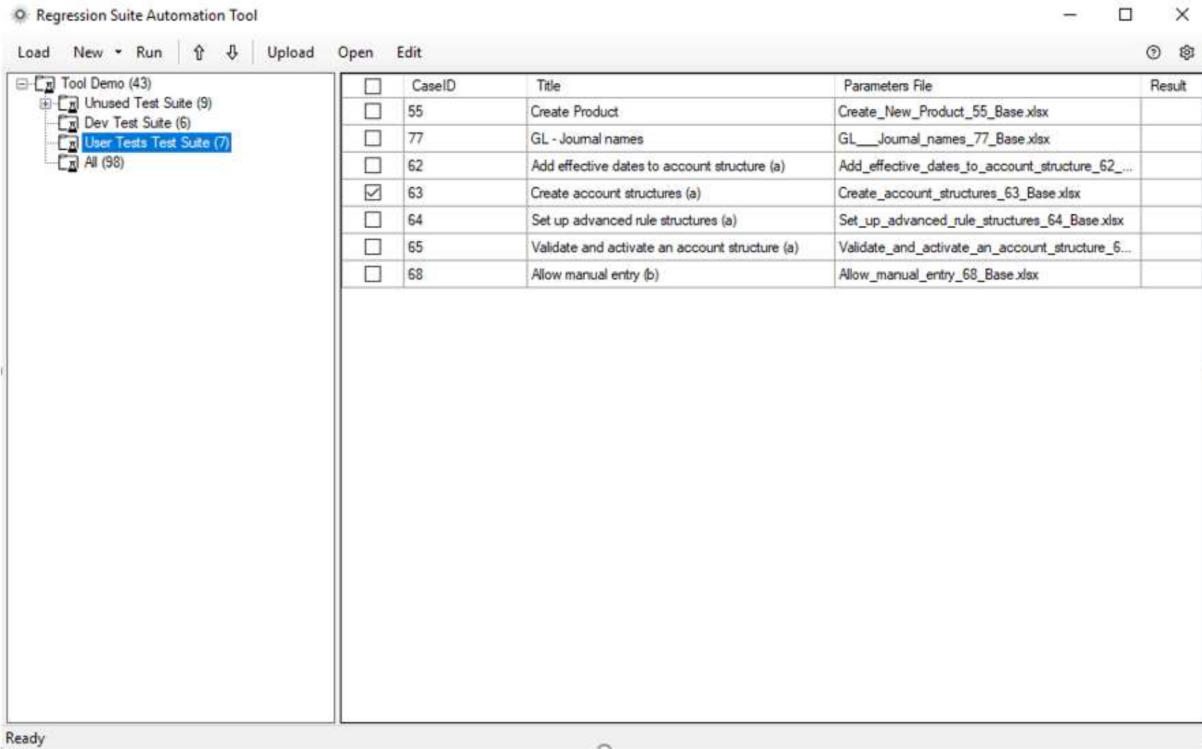
Customers who participate in the First release program will have an opportunity to update their sandbox environment and other environments early. To sign up for the First release program, go to <https://experience.dynamics.com>.

Customers will also be able to **self-update** by using the standard update experience in LCS, as shown in the following illustration.



# Validate

After an update is completed in the UAT environment, a basic business process test can be executed to validate the environment. To support this effort, a no-code automation test tool for business process testing is available, as shown in the following illustration. For more information, see [Create and automate user acceptance tests](#).



Some customers have both external data integrations and internal data integrations. We recommend that these customers use the [Data task automation tool](#) for testing.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# One Version service updates FAQ

2/18/2021 • 18 minutes to read • [Edit Online](#)

In July 2018 we announced a [change to the way we deliver Dynamics 365 updates](#) that will help you stay current in a consistent, predictable, and seamless manner. In June 2019, based on customer feedback we announced [New flexible service updates being made available](#). This FAQ is intended to provide clarity on the service updates, processes, and tools you can use to prepare for it. We will continue to add additional information to this topic as needed.

## Can the update be delayed, what is the policy?

Yes, the customer can pause, delay, or opt-out of an update via Update Settings in the Lifecycle Services projects. A customer can choose to pause up to 3 consecutive updates. The following is an example of a delayed update:

- The customer is currently on version 10.0.2.
- The customer can pause updates 10.0.3, 10.0.4, and 10.0.5.
- The customer must take the 10.0.6 update when it is available.

## With a release date in early April, when will the general availability package be made available?

Production updates for a monthly release will be scheduled for the first, second, and third weeks in of the month. Depending on the configuration that you set up in Lifecycle Services (LCS), you will receive updates during that specific week.

For the April 10.0 release, Microsoft will perform updates during the weekends of April 6, April 13, or April 20 based on the configuration that you set up in LCS. Sandbox updates will always be scheduled a week before the update. The configuration setup is available in LCS.

Customers can always choose to apply the update at an earlier time, or if there is a more convenient time than the suggested times in Lifecycle Services. If the customer is on the latest version the auto update will be canceled.

## Service updates

### What product versions are impacted by service updates?

VERSION	DESCRIPTION
8.1 and later	All customers on 8.1 and later will be scheduled for automatic monthly updates with a combined application and platform update starting November 2018. You will be required to be on an update that's no older than 4 months or 3 service updates. To pause an update, refer to <a href="#">Pause service updates</a> .
8.0	Customers on 8.0 can manually apply the monthly platform and financial reporting updates. You will be required to have an update that's no older than 4 months or 3 service updates. The 8.0 application lifecycle ends in April 2019. Customers on 8.0 must update by April 30, 2019 to stay supported. In order to be on a supported application, customers should follow the process to update to the latest version. For more information, see <a href="#">Update environments from version 8.0 to 10.0.X</a> .

VERSION	DESCRIPTION
7.x	Customers on 7.x can manually apply the monthly platform and financial reporting updates. You will be required to have an update that's no older than 4 months and 3 service updates. Customers on 7.x must update by April 30, 2019 to stay supported. If customer stays on version 7.3 past April 30 they will still receive automated platform updates every month. You are required to upgrade to 8.1 by April 2019 (unless extensions are not available). The only overlaid version in market will be version 7.3.

### What does the service update contain?

For release 8.1 and later, service updates will contain application (including financial, reporting, and Commerce) and platform changes that are critical improvements to the service including regulatory updates. New experiences will be configurable. The service updates are backward compatible. There will be a single version representing this update.

### What is a regulatory update?

A regulatory update is a new feature or an existing feature change required by law (usually for a specific country/region). The regulatory update is always required by a specific law enforcement date (LED) and should be enabled by this date or earlier.

### What's the upcoming schedule of updates?

Service updates are available since November 2018. You have the option to apply the update when it is convenient for you, or let Microsoft auto-apply the service updates based on the selected maintenance window. You are required to have an update no older than 4 months.

To see a targeted release schedule, see [Service update availability](#).

### Are there any major updates post 8.1?

There will be 2 major updates in April and October where new experiences can be enabled. Major updates will not require code or data upgrade. Breaking changes will be communicated 12 months in advance such that customers can plan accordingly. Such a change will only be introduced during a major update. The 10.0 release, which will be available in April 2019, will also be an update and not an upgrade.

### What does it mean when an update is backward compatible?

Backward compatibility covers binary and functional compatibility. Binary compatibility means that you can apply an update on any runtime environment without needing to recompile, reconfigure, or redeploy customizations. This also means that on a development environment at design time, X++ public and protected APIs and metadata are not modified or deleted. If Microsoft needs to break compatibility by removing obsolete APIs, it will be communicated 12 months in advance and follow a deprecation schedule. Functional compatibility is about user experience, all new experiences will be opt-in for a 12-month period.

Backward compatibility does not include non-X++/metadata APIs. Microsoft reserves the right to update versions of any dependencies the product uses, as well as remove dependencies without early warning. Microsoft does not commit to maintain backwards compatibility of dependent software libraries unless expressly stated.

For more information on deprecation guidelines and deprecated methods and metadata elements, see [Deprecation of methods and metadata elements](#).

### Can I apply a Platform service update to my existing 8.1 or later environments?

Customers on version 8.1 or later will only be able to apply the 8.1.x or v10.x Service updates. Platform only service updates cannot be applied to version 8.1.x or later. Platform service updates can only be applied to versions 7.x or 8.0.

### **Will Platform updates be able to be scheduled and delay/pause by customers?**

Yes, customers who are on version 7.3 are able to schedule platform updates directly in Lifecycle Services. A delay/pause experience is also available.

### **Service updates for on-premises deployments**

The policy and schedule for service updates are now the same for both cloud and on-premises deployments. This includes, for example, the option to delay applying up to 3 consecutive updates. How to apply each of these updates remains slightly different. For more information, see [Apply updates to on-premises deployments](#).

## Process

### **How will Microsoft ensure quality of releases?**

Ensuring quality of the release is a fundamental principle that's enabled through a series of progressive, rigorous, automated validations as described in [Service update availability](#).

### **Can I select the day and time to update?**

Customers can configure the day and maintenance time windows in LCS. The service update, which is based on your update settings, will start within 15 minutes. Email will be sent to customers who opt in to receive LCS notifications with instructions included on how to update. Customers will be able to select the designated tier 2/UAT sandbox for the update. Customers will have 7 calendar days for testing and validation.

Customers can optionally choose to apply the update earlier to all environments through LCS. The production-ready, deployable package will be made available to all customers via the Action Center in Lifecycle Services. Customers are responsible for deploying the update to any additional sandbox or developer/build (tier 1) environments. For more information, see [Configure service update](#).

### **A service update was applied to the environment, when looking at the tile in Lifecycle Services for this environment what does the number on the tile represent?**

The same service update will be auto applied to all customers by Microsoft. Microsoft will continue to service the latest update. The tile in LCS for that environment represents the cumulative number of hotfixes that are available to be applied to your environment. Because Microsoft will only auto apply the same version to all customers, you will be responsible for apply the cumulative hotfix package if it is required.

### **How do I update to the latest version?**

Users can update to the latest version using the tiles on the Environment details page in LCS. After the update is released by Microsoft, the tile will show the update available. Customers can choose to apply the update on their own by going through the update experience on their sandbox and production environments.

### **How do I update the production environment to the same version after Microsoft updates the sandbox environment?**

When Microsoft updates a sandbox environment, the package that is used for the update is saved in the project's asset library. The name of the package is prefixed by the words "Service Update." Because the package was already applied to the sandbox environment, you can mark this package as a Release Candidate. You can then go to the production environment and schedule to apply the package, just as you might schedule to apply any other update.

### **What is the expected downtime?**

The expected downtime for a successful update is 30 minutes to 1 hour. However, we ask for three hours of downtime in case issues occur while the update is applied. We are actively working to reduce the downtime that is required, and you should expect improvements in the next few months.

### **What's the process for deprecation?**

Before any feature is removed from the product, the deprecation notice will be announced in the product documentation 12 months prior to the removal.

For breaking changes that only affect compilation time, but are binary compatible with sandbox and production environments, the deprecation time will be less than 12 months. Typically, these are functional updates that need to be made to the compiler.

### **Can I delay an update?**

You can pause an update up to 4 months or 3 consecutive service updates by way of LCS configuration. After this period, an update will be scheduled and auto-applied by Microsoft. The update experience for a delayed update will incur additional downtime.

### **Can I delay an update for longer than 3 consecutive service updates due to seasonal activity or other business reason?**

No, service updates will be automatically applied to the sandbox, then 7 days later the update will be applied to the production environment if the environments are more than 3 service updates old. A customer can only pause up to 3 consecutive updates in a row. For example, if a customer on version 10.0 chooses to pause updates 10.0.1, 10.0.2, and 10.0.3 then service update 10.0.4 will be auto applied to the sandbox.

### **What if I find an issue during the sandbox update?**

If you find an issue when doing validations in a sandbox environment, you can request to skip the update through LCS directly by providing a valid support ticket number and a business justification.

### **What if I find a critical issue during sandbox testing and I am not able to pause the Production auto update?**

Critical issues should always be submitted to the support team via Lifecycle Services as soon as they are identified. The support staff will work with you on the resolution to the critical issue.

### **How much time do I get for validation?**

You will get 7 calendar days for validation after the update is applied to your sandbox environment. If you need more time, you can access the deployable package via the action center in Lifecycle Service and apply to your environments. This will provide you with additional time to test the update prior to a production roll-out.

### **What happens when the service update is complete?**

Once the service update is applied by Microsoft you will receive a notification if the update was successful or if it was not able to be applied. There can be several reasons an update was unable to be applied:

- Pending Package Sign-off - If a package is pending signoff, Microsoft will not apply the service update to production.
- Deployment Failure - If there was a deployment failure, the environment will be rolled-back to the original state.

### **If there is a failure, can I reschedule the update to be auto applied?**

You will not be able to reschedule the update per se, but you may apply the package when it is convenient for you, just as you might schedule to apply any other update.

### **Will critical hotfixes be automatically applied to my sandbox/ production environment during auto-update?**

The service update that will be made generally available, and auto applied to all customers will contain hotfixes and potentially new functionality. If a critical issue is reported after the service update has been applied, customers can pull that cumulative hotfix update from the tile in Lifecycle Services.

### **How will my ISVs stay current?**

Service updates to customer environments will be backward compatible and no action is required by the Independent software vendors (ISVs). ISVs develop on the minimum required platform release that their code depends on. Breaking changes will have a 12-month lead time to enable ISVs to include and validate. We recommend that the ISVs be part of our [Partner early access program](#), so that they can get early access to the platform bits and validate their solutions against the update before it's made generally available.

### **What about new features?**

All new features will be opt-in for a 12-month period and will not require any change management until you choose to enable the feature.

### **Are batch jobs suspended during a service update?**

Batch jobs are suspended during the maintenance windows and resume when the maintenance is completed.

## Tools

### **How can I get early access to non-released platform updates?**

You can join the [First release program](#), where Microsoft will keep your system always current with the latest updates. If you are not already a member of the Dynamics 365 Insider Program, you will need to:

1. Sign up for the Insider Program using this URL: <https://experience.dynamics.com>
2. Accept the terms and conditions to become a Dynamics 365 Insider.
3. After your application has been approved (approximately 24 hours) you can then sign back into the Insider Portal to find the different preview programs available for you to join.
4. Preview Early Access Program (PEAP) and First Release: The program requires that you accept additional terms and conditions to join. Please look for these programs within the Dynamics 365 Insider Program after your nomination has been accepted.

### **Is there tooling available to support testing the latest release?**

The [Regression Suite Automation Tool](#) is [available now](#). This tool significantly reduces the time and cost of user acceptance testing. User acceptance testing is typically required before taking a Microsoft application update or applying custom code and configurations to your production environment. It enables functional power users to record business tasks using the Task recorder and convert them into a suite of automated tests without the need to write source code. Test libraries are stored and distributed in Lifecycle Services using the Business Process Modeler (BPM) libraries and fully integrated with Azure DevOps for test execution, reporting, and investigation. Test data parameters are decoupled from test steps and stored in Excel data files.

### **How can I test and validate that the integrations continue to work?**

Data task automation lets you easily repeat many types of data tasks and validate the outcome of each task. You can also use automated testing of data entities by using task outcome validation. For more information, see the [Data task automation](#) topic.

### **How can I determine what's changed in a service update?**

The What's new or Changed documentation is the primary source for the details contained in each service update. The [Release plans](#) are the primary source of information for all new features and changes for a future release. Features will also include help topics in docs.microsoft.com as needed. An impact analysis tool will be available in LCS to help you better understand the impact on the features that you use.

### **How will I know if there is a deprecated feature that will impact me if I'm not doing active development/recompile my code?**

Deprecated features will be documented with each release. For more information, see [Removed or Deprecated features](#).

## Preparing for One Version

### **How can I log an extensibility request?**

Extensibility requests can be logged in LCS. Details are available in the [Extensibility requests](#) topic. Please note the following timelines to log and use the available extensions.

DATE	EXTENSIBILITY REQUESTS
January 2019	All extensibility requests must be logged by January 1, 2019. ISVs and customers are requested to analyze the code and make these requests by this time. We will not provide exceptions to stay on 7.3 after April 2019, if the request has not been filed by January 1, 2019.
December 2019	Extensions will be available on/ before December 31, 2019 for the requests logged by January 1, 2019. Customers using these extensions are required to move to current version by April 2020.

### **What does end of service mean?**

Microsoft will not provide any fixes to issues on versions that have reached end of service. Microsoft will also not investigate or troubleshoot any issue that you may encounter on a version that's older than 3 service updates. If you encounter an issue on a version that has reached end of service, you will be required to update to the latest update and report the issue if it persists.

All environments will continue to be operated by Microsoft. All automatic processes around your environments, such as monitoring or self-healing, will also continue as long as the environment is on a supported version.

### **Will individual hotfixes be supported?**

Individual hotfixes will not be supported after 8.1. Customers must update to the latest cumulative update available to apply the fix (such as 8.1.1). Critical fixes will also be cumulative and available through the LCS servicing experience.

### **Will you notify me about critical hotfixes released for the monthly update that I'm on?**

Customer reported issues are searchable via Lifecycle Services Issue Search. You can sign up to be notified when an open issue is resolved.

### **How can I upgrade to 8.x?**

Refer to the [Process for moving to the latest update](#) topic to learn how to upgrade to the latest application. Updating from [8.0 to 8.1](#) will not require any data upgrade and will be a self-serve update with much reduced downtime.

## Commerce service updates

### **What options are available to minimize impact to my Commerce cloud components?**

Commerce cloud components will require the same down time as your Dynamics 365 headquarters. In an upcoming release, the Retail Cloud Scale Unit (RCSU) will be available to reduce and further schedule updates to your deployment. Please refer to our published release information on our [documentation](#) and [release notes](#) sites for additional details on RCSU.

### **Will there be options to take individual hotfixes for my commerce solution components?**

All fixes and updates for commerce components will be cumulative.

### **What are the maintenance downtime requirements that may impact channel operations?**

For retailers with a business need for redundancy, Modern POS offline capability allows core POS operations to be available for use while disconnected from the internet or while the cloud environment is being updated. Stores operating with Commerce Scale Unit will also continue to operate with support for core POS operations during cloud maintenance windows. For more information, see [Online and offline point of sale \(POS\) operations](#).

### **When will I need to update my in-store components?**

All in-store components must be running released software that is less than one year old in order to maintain support. Customers are responsible for updating self-hosted components (such as components installed in stores or in privately managed datacenters) and ensuring that the installed versions of these components are actively supported.

### **Will there continue to be backward compatibility for the in-store components?**

Updates to components hosted in the cloud will continue to preserve backward compatibility with component versions self-hosted by the retailer (such as components installed in stores or in privately managed datacenters - Modern Point of Sale, Commerce Scale Unit, Hardware Station) for 12 months after the release date for that version. Self-hosted components do not need to be updated at the same time as cloud-hosted components and can be updated on a separate cadence allowing time to roll out updates to stores.

### **What options are available for updating in-store components across my organization?**

Customers can choose to update self-hosted components manually at each store or use mass update tools such as Microsoft System Center Configuration Manager, Microsoft Intune, etc.

### **What options do I have to slowly enable new functionality across my channels?**

Microsoft provides several mechanisms to progressively roll-out and enable functional enhancements across stores, devices, and users.

- **Screen layout designer** – Most visual elements in POS are configured and centrally managed by an administrative user in the customer organization. This means that new POS operations will not automatically be displayed on POS unless explicitly configured for inclusion in corresponding screen layouts. Screen layouts are configured using Screen layout designer and can be specific to a store or POS device. For more information, see [Screen layouts for the point of sale \(POS\)](#).
- **Functionality profiles, POS permissions, Commerce parameters** – Significant elements of functionality in POS are typically configurable by the user. This can be configured through functionality profiles, POS permissions, commerce parameters, or other controls which allow for device, register, store, or user-level functionality control in applicable scenarios.
- **Modern Point of Sale and Commerce Scale Unit** – Because Modern Point of Sale and Commerce Scale Unit are self-hosted by the retailer, topologies which include either of these components enable roll out of updates at a separate (and slower) cadence, and in a more granular fashion than with cloud-only topologies.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage Finance and Operations updates and your custom code lifecycle

2/18/2021 • 19 minutes to read • [Edit Online](#)

This topic describes application lifecycle use cases for Finance and Operations implementations. It's focused on the following scenarios:

- Managing your source code development branches
- Applying the next version of a Microsoft service update
- Applying a new version of your custom code

This topic applies to Microsoft Dynamics 365 Finance, Dynamics 365 Supply Chain Management, Dynamics 365 Commerce, and Dynamics 365 Project Operations.

The main goal is to show how to complete the following tasks:

- Stay up to date and manage Microsoft service updates (or quality updates) for Finance and Operations apps (including Dynamics 365 Commerce) in incremental phases, independently of the lifecycle of your own customization. This approach simplifies the update process, and reduces the cost and risk of regressions that are associated with all-in-one upgrade projects.
- Take advantage of source code branches for version control of your custom code. By using version control, you can isolate the rollout of critical changes and hotfixes from the development of new features and capabilities.

This topic doesn't explain how to use the different tools in Azure DevOps and Microsoft Dynamics Lifecycle Services (LCS). Instead, it's focused on processes and best practices. The [Apply the next version of a Microsoft service update](#) and [Apply a new version of your custom code](#) sections contain both an overview of the phases and the steps of the process.

This topic includes the following sections:

- [Environments](#)
  - [Environments that run your current release](#)
  - [Environments that run the next version of your custom code](#)
- [Manage source code branches](#)
- [Apply the next version of a Microsoft service update](#)
  - [Backward compatibility of Microsoft updates](#)
    - [Runtime compatibility](#)
    - [Design-time compatibility](#)
  - [Phase 1: Update the Finance and Operations implementation](#)
    - [Track 1: Update your runtime environments](#)
      - [Update Test 1](#)
      - [Update UAT](#)
      - [Update Prod](#)
    - [Track 2: Update your development environments](#)



- Error situations
  - Case 1
  - Case 2
  - Case 3
- Phase 2: Update CSU to version 10.0.11
  - Phase 2 prerequisites
  - Track 1: Update your CSU runtime environments
    - Update Test 1
    - Update UAT
    - Update Prod
  - Track 2: Update your development environments
- Phase 3: Update POS to version 10.0.11
  - Phase 3 prerequisites
  - Update your Commerce development environment
  - Option 1: Store component updates that include only runtime changes
  - Option 2: Store component updates that include runtime and custom changes
  - Update Process
    - Update Test 1
    - Update UAT
    - Update Prod
- Apply a new version of your custom code
  - Create a hotfix of your code
  - Update your custom code from release N to release N+1
- Upload, update, and deploy store components

## Environments

This section describes the collection of Finance and Operations environments that the application lifecycle management (ALM) scenarios in this topic rely on. This configuration is typical for organizations that have implementations that rely on custom code (extensions). This custom code includes customizations that are provided by independent software vendors (ISVs).

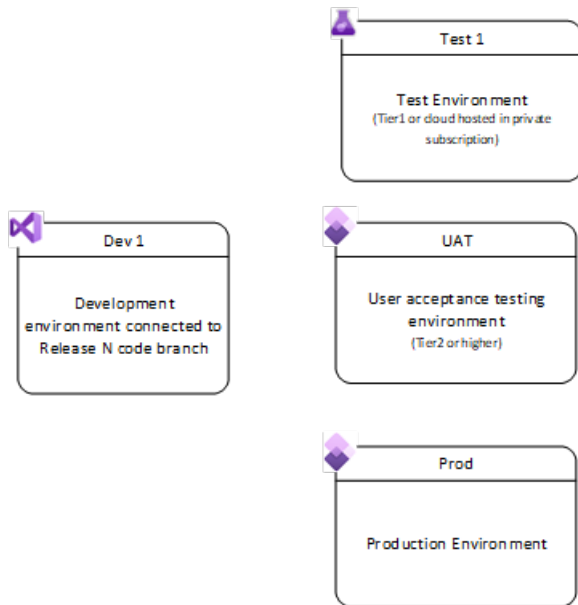
### Environments that run your current release

The following environments are the environments in your current release:

- **Dev 1** – A development environment that runs the same version of Finance and Operations apps as the production environment. The Dev 1 environment uses Azure DevOps for version control of custom code. It's connected to the current release branch of your custom code. For more information, see the [Manage source code branches](#) section.
  - There are many options for development environments, both cloud and on-premises. For more information, see [Deploy and access development environments](#).
  - For build automation, use the new Azure DevOps Hosted Agents functionality. For more information, see [Build automation that uses Microsoft-hosted agents and Azure Pipelines](#).
- **Test 1** – A Tier-1 test environment that is used for functional and configuration testing. The Test 1

environment runs the same version of Finance and Operations apps as the production environment. It also runs the latest release version of your custom code extensions.

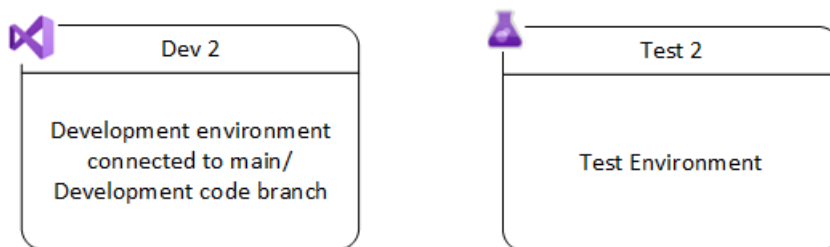
- **UAT** – A pre-production environment that is used for user acceptance testing. The UAT environment is a Tier-2 (Standard Acceptance Test) or higher environment. It runs the same version of Finance and Operations apps as the production environment. It also runs the latest release version of your custom code extensions. This environment is typically connected to a copy of the production database.
- **Prod** – Your live production environment that runs on your production database.



### Environments that run the next version of your custom code

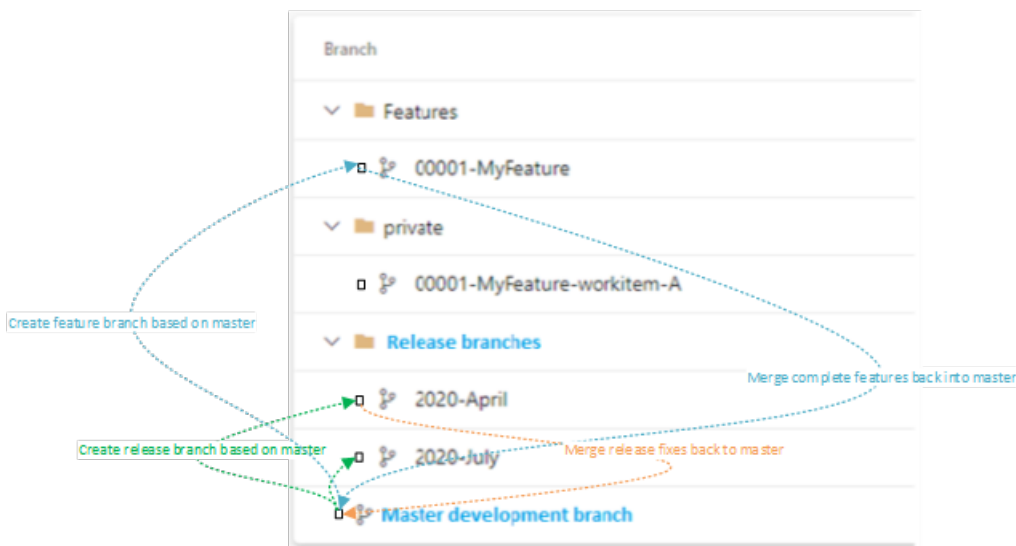
The following environments run the next version of your custom code:

- **Dev 2** – A development environment that is used for development of the next version of your custom code extensions. It uses Azure DevOps for version control of custom code. It's connected to the development branch (**main** branch) of your custom code. For more information, see the [Manage source code branches](#) section.
- **Test 2** – A functional test environment that is used for testing of the next version of your custom code extensions.



## Manage source code branches

It's important that you follow best practices as you manage branches of custom code. In this way, you help minimize cost, and ensure the quality of your releases and updates.



The **main** branch (development branch) contains the latest functioning version of the next release of your code.

When you work on new features, create a new feature branch out of the **main** branch. Then, when the feature work is completed, integrate the feature branch back into the **main** branch.

The release branches contain the code base of your official releases. In the preceding illustration, the assumption is that you have only one release branch, **release/2020-April**. **2020-April** isn't a build. It's a source code branch. Because you will probably create hotfixes for your release, you will make changes and produce builds out of this release branch.

- Don't use release branches to develop new features. Use them only for critical fixes or changes that are required in your live environment.
- After you've made a change in a release branch, integrate the branch back into the **main** branch. In this way, you ensure that your next release also contains the fix.
- Among the example environments that were described earlier, the Dev 1 environment will be connected to the **release/2020-April** branch.

When you're ready to release a new version of your custom code, create a new release branch that is based on the **main** branch. For this example, you will create a new release branch that is based on **main** and named **2020-July**.

You might have private branches that individual developers work in while they work on a specific work item that is based on a specific branch of your code. Private branches are merged back into their parent branch when the work is completed. For more information, see [Learn about branching strategies for Team Foundation Version Control \(TFVC\) and how to select an effective strategy](#).

## Apply the next version of a Microsoft service update

By using a phased approach, you help maximize the efficiency when service updates are taken. Each phase updates one component of your implementation.

### 1. **Phase 1** – Update your Finance and Operations environments.

Your current version of Commerce Scale Unit (CSU) and Point of Sale (POS) will work correctly with the new Finance and Operations update. For example, version 10.0.7 of CSU is compatible with version 10.0.11 of Finance and Operations apps.

### 2. **Phase 2** – Update CSU.

### 3. **Phase 3** – Update POS.

When you take a Microsoft update, you don't have to update your custom code to the next version. By taking

Microsoft updates without bundling them with custom code updates, you help simplify the update process. You also help reduce the cost and risk of regressions that are associated with all-in-one upgrade projects.

### **Backward compatibility of Microsoft updates**

It's important that you understand what Microsoft means by *backward compatibility of service updates*, so that you have context for the next sections of this topic. Service and quality updates are *runtime* backward-compatible. However, they aren't always *design-time (compile-time)* backward-compatible.

#### **Runtime compatibility**

All Microsoft updates are intended to be runtime backward-compatible. This compatibility covers both binary compatibility and functional compatibility. Runtime compatibility means that customizations that exist in production and sandbox environments will continue to work after Microsoft service updates are deployed to those environments. Those updates include service updates and quality updates. Runtime compatibility also means that Microsoft updates are backward-compatible with customizations that were compiled on an earlier platform.

Binary compatibility is backward only. You can compile a customization on an older application version and platform version, and deploy it to an environment that is running a later version. However, you can't deploy code to an environment that is running a version that is earlier than the version that the code was compiled on.

#### **Design-time compatibility**

Design-time (compile-time) backward compatibility means that developers can apply updates to their development environments and successfully compile their code without having to make any changes.

Microsoft aims for design-time compatibility. However, some updates might include changes that aren't compatible at design time, but that are binary-compatible. Therefore, after an update is applied, new errors or warnings might occur when your code is compiled. Here are some examples of these changes:

- Microsoft makes an enumeration extensible.
- Microsoft marks an API as obsolete or internal.
- Microsoft introduces a new compiler error to help prevent unsafe coding practices.

All these changes might require work on your solution. Design-time breaking changes that are binary-compatible don't require a 12-month deprecation notice. These breaking changes are documented for each service update. For more information, see [What's new and changed in Platform updates](#).

### **Phase 1: Update the Finance and Operations implementation**

This section summarizes the process that you use to update your Finance and Operations implementation to the latest service update. An update from version 10.0.7 to version 10.0.11 is used as an example.

This phase is divided into two tracks. These tracks can occur in parallel.

- **Track 1** – Update your runtime environments.
- **Track 2** – Update your development environments.

After you complete track 1, you will be live on version 10.0.11, unless you encounter one of the error situations that are described in the [Error situations](#) section.

#### **Track 1: Update your runtime environments**

By completing track 1, you essentially complete your Finance and Operations update to version 10.0.11, because your production environment will be live on version 10.0.11. You don't have to recompile your custom code as part of this track.

##### **Update Test 1**

The Test 1 environment is running version 10.0.7 together with the latest released version of your custom extension. Although an update of Test 1 isn't a prerequisite in this flow, it's recommended, because it adds another level of functional verification and predictability. This update should be completed before the UAT environment is updated. The sooner that you update Test 1 and validate on it, the more predictable your "real"

update (that is, the update of the UAT and Prod environments) will be.

1. Apply version 10.0.11 of Finance and Operations apps to Test 1.
2. Sign off on functional scenarios. You can use the [Regression Suite Automation Tool](#) to automate user acceptance testing on test and UAT environments.
3. If regressions are encountered, see the [Error situations](#) section.

#### Update UAT

The UAT environment is running version 10.0.7 together with a released version of your custom extension. (The UAT environment is the same as the Prod environment.)

1. Apply version 10.0.11 of Finance and Operations apps to UAT.

Your UAT environment might be configured so that it's automatically updated by Microsoft. However, you can always pull the update as soon as it's available.

2. Complete user acceptance testing, and sign off.
3. If regressions are encountered, see the [Error situations](#) section.

#### Update Prod

1. Apply version 10.0.11 of Finance and Operations apps to Prod.

Your Prod environment might be configured so that it's automatically updated by Microsoft. However, you can always pull the update as soon as it's available.

2. Sign off.

### Track 2: Update your development environments

The purpose of track 2 is to update the Dev 1 environment to version 10.0.11. Dev 1 is your **main** development environment that is connected to the current release branch of your custom code. It's running version 10.0.7 of Finance and Operations apps. By completing track 2, you ensure that Dev 1 runs version 10.0.11 together with your latest release, and that it's ready for any future hotfixes that are required for your code.

1. Apply version 10.0.11 of Finance and Operations apps to Dev 1.
2. Compile your custom code, and do testing.
3. Make any required changes to your custom code.
4. Check code changes in to the release branch.
5. If you have more than one development environment, follow these steps for each environment:
  - a. Apply version 10.0.11 of Finance and Operations apps to the development environment.
  - b. Sync and compile your latest custom code from the target code branch.

### Error situations

#### Case 1

During track 1, a bug is found that requires a Microsoft quality update. If Microsoft releases the update in a timely manner, use the new Microsoft update instead of the original service update to complete track 1.

#### Case 2

During track 1, a bug is found that requires a Microsoft service update. However, the update can't be released in a timely manner.

As a workaround for this issue, Microsoft proposes a change to your custom code.

1. Modify your code in Dev 1 (in the release branch), and test your updates.
2. Check code changes in to the release branch.
3. Create a new deployable package out of the release branch.

#### 4. Apply the new deployable package to Test 1 and/or UAT.

##### NOTE

Custom code that is compiled on version 10.0.7 can be deployed to any runtime environment that is running version 10.0.7 or later. Therefore, you don't yet have to update Dev 1 to version 10.0.11. However, you might already have done that update as part of track 2.

##### Case 3

During track 1, a bug is found that requires a change to your custom code. This bug might be in either your code or the ISV code.

1. Modify your code in Dev 1 (in the release branch), and test your updates.
2. Check code changes in to the release branch.
3. Create a new deployable package out of the release branch.
4. Apply the new deployable package to Test 1 and/or UAT.

### Phase 2: Update CSU to version 10.0.11

This section summarizes the process that you use to update your Commerce Scale Units to the latest service update. An update from release 10.0.7 (Commerce version 9.17) to release 10.0.11 (Commerce version 9.21) is used as an example.

#### Phase 2 prerequisites

Before you update your CSUs, you must update the Commerce headquarters environments (in the Finance and Operations app) to the same release or a later release. For this example, that release is 10.0.11.

This phase is divided into two tracks. These tracks can occur in parallel.

- **Track 1** – Update your CSU runtime environments.
- **Track 2** – Update your development environments.

After you complete track 1, you will be live on release 10.0.11 (Commerce version 9.21), unless you encounter one of the [error situations](#) that were described for phase 1.

#### Track 1: Update your CSU runtime environments

By completing track 1, you essentially complete your CSU update to version 10.0.11, because your production environment will be live on version 10.0.11. You don't have to recompile your custom code as part of this track.

##### Update Test 1

The software as a service (SaaS) components of Commerce aren't currently supported in Tier-1 environments (development/test environments). A copy of Retail Server runs locally in each Tier-1 environment, and the deployment of both Microsoft code for Retail Server and retail customizations will be done through the previous system, where application binary packages and retail deployable packages are applied against the infrastructure as a service (IaaS) instance.

##### Update UAT

The UAT environment is running CSU that corresponds to release 10.0.7, together with the same version of your retail extension that the production environment is running.

1. Update the Commerce Scale Unit. Select **9.21 (10.0.11)** as the target version. For more information, see [Phase 2: Update CSU to version 10.0.11](#)
2. Complete user acceptance testing, and sign off.
3. If regressions are encountered, see the [Error situations](#) section.

##### Update Prod

1. Update the Commerce Scale Unit. Select **9.21 (10.0.11)** as the target version. For more information, see [Phase 2: Update CSU to version 10.0.11](#)
2. Sign off.

## Track 2: Update your development environments

1. Get the latest version of the Retail software development kit (SDK).
  - a. Apply the Finance and Operations service update for release 10.0.11 to Dev 1.
  - b. Get the updated version of the Retail SDK from %ServiceDrive%\RetailSDK\Update\**<newest directory>**
2. In your **main** (development) branch, update the Retail artifacts from the new version of the Retail SDK.
3. Compile. The new version should be backward-compatible and should not require any changes to your code.
4. Commit the change that includes the Retail SDK update.
5. Make any required changes to your custom code.
6. Commit code changes to the target branch.
7. Optional: If you have more than one development environment, merge the latest changes from step 4, and compile the latest custom code.

## Phase 3: Update POS to version 10.0.11

This section summarizes the process that you use to update your store components, such as Modern POS and Hardware Station, to the latest service update. An update from release 10.0.7 (Commerce version 9.17) to release 10.0.11 (Commerce version 9.21) is used as an example.

Unlike updates for Commerce headquarters (in the Finance and Operations app) and CSU, updates for store components are delivered in the same packages. After you update Commerce headquarters and CSU, you have the following options:

- **Option 0 (no operation is required)** – Leave the store components in their previous release if the version is supported and in-policy.
- **Option 1** – Update the store components runtime (Microsoft code) so that it matches the same release as CSU.
- **Option 2** – Update the store components runtime (Microsoft code) and customization together.

In the rest of this section, the assumption is that you want or have to update the store components (option 1 or option 2).

After you complete this phase, you will be live on release 10.0.11 (Commerce version 9.21) for store components, unless you encounter one of the [error situations](#) that are described for phase 1.

### Phase 3 prerequisites

Phase 3 has the following prerequisites:

- The Commerce headquarters components (in the Finance and Operations app) were updated to the same release or a later release before the CSUs were updated. In this example, the version is 10.0.11.
- The CSUs were updated to the same release as, or a later release than, the store components. In this example, the release is 10.0.11.

### Update your Commerce development environment

Follow the steps in the [Track 2: Update your development environments](#) section.

#### Option 1: Store component updates that include only runtime changes

This option generates a new retail deployable package that contains your store components. It includes changes only from the Microsoft code.

1. Update your release branch, which has the code that is currently being used in the production environment, to the Retail SDK that matches your target release. In this example, the version is 10.0.11 (9.21). Follow the

steps in the [Track 2: Update your development environments](#) section.

2. Generate a new build for the retail deployable package. This build contains the same set of customizations that is currently in the production environment, plus version 10.0.11 (9.21) of the Microsoft code.

#### **Option 2: Store component updates that include runtime and custom changes**

This option generates a new retail deployable package that contains your store components. It includes changes from both the Microsoft code and customizations.

1. Update your release branch, which has the code that is currently being used in the production environment, to the Retail SDK that matches your target release. In this example, the version is 10.0.11 (9.21). Follow the steps in the [Track 2: Update your development environments](#) section.
2. Commit changes.
3. Update custom code for store components, or update references to ISV-updated components.
4. Generate a new build for the retail deployable package. This build contains updated custom code plus version 10.0.11 (9.21) of the Microsoft code.

#### **Update process**

##### **Update Test 1**

The SaaS components of Commerce aren't currently supported in development or test one-box environments. A copy of Retail Server runs locally in each development or test one-box environment, and the deployment of both Microsoft code for Retail Server and retail customizations will be done through the previous system, where application binary packages and retail deployable packages are applied against the IaaS instance.

The Test 1 environment is running version 10.0.11 of Commerce headquarters and a local version of CSU from previous phases. An update of Test 1 isn't a prerequisite in this flow. Because there is no CSU in this environment, this step is mostly for verification.

1. Upload, update, and deploy the store components. For more information, see the [Upload, update, and deploy store components](#) section.
2. Sign off on functional scenarios.
3. If regressions are encountered, see the [Error situations](#) section.

##### **Update UAT**

Clients that point to the UAT environment are running applications (for example, Modern POS) for release 10.0.7 (the same version that is currently running in the production environment).

1. Upload, update, and deploy the store components. For more information, see the [Upload, update, and deploy store components](#) section.
2. Sign off on functional scenarios.
3. If regressions are encountered, see the [Error situations](#) section.

##### **Update Prod**

1. Upload, update, and deploy the store components.
2. Sign off.

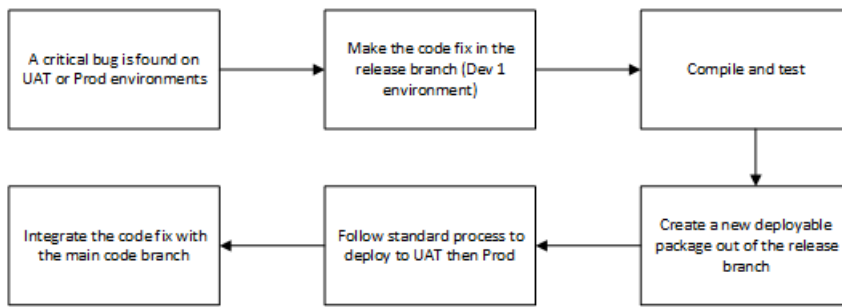
## Apply a new version of your custom code

This section describes the recommended flow for two use cases that require that you update your UAT and/or Prod environment with a new build of your custom extension.

#### **Create a hotfix of your code**

When a critical bug is found in the UAT or Prod environment, fix the bug in the release branch (not in the **main** or development branch), and use the standard process to apply a deployable package to UAT and Prod.

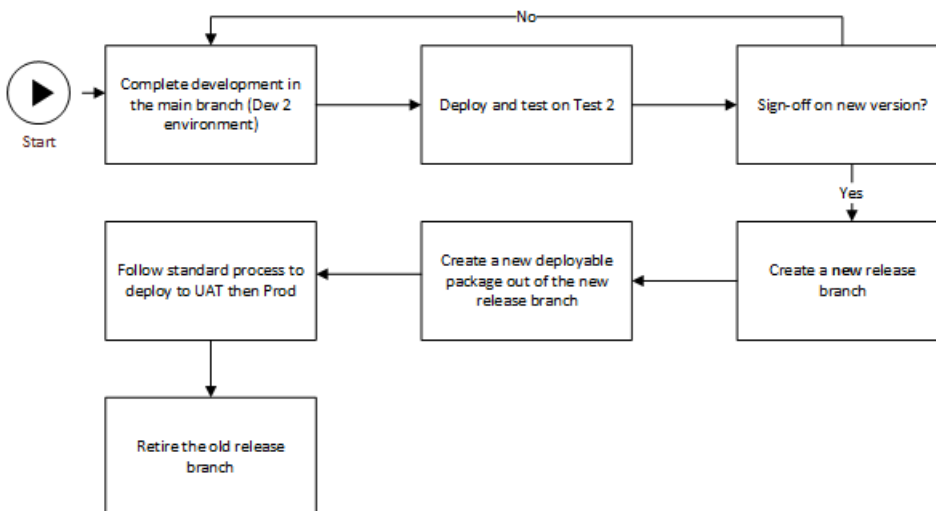




1. In Dev 1, make the code fix in the release branch. If the required fix is in ISV code, ask the ISV to send you a new build of the current release, not the next version of the solution.
2. Compile and test.
3. Create a new deployable package out of the release branch.
4. Deploy on Test 1, and sign off if sign-off is required.
5. Use the standard process to deploy custom code first to UAT and then to Prod.
6. Integrate the code fix with your **main** code branch.

### Update your custom code from release N to release N+1

When you're ready to release the next version of your custom code, use the following process to create and deploy the new release.



1. In the Dev 2 environment, or another environment that is connected to the **main** branch, complete development in the **main** branch.
2. Deploy and test on Test 2.
3. Repeat steps 1 through 2 until the new version is signed-off on.
4. Create a **new** release branch.
5. Create a new deployable package out of the **new** release branch.
6. Use the standard process to deploy custom code first to UAT and then to Prod.
7. Retire the old release branch.

## Upload, update, and deploy store components

1. Upload the retail deployable package to LCS.
2. Update Application Object Server (AOS) with the latest client.
3. Upload the retail deployable package to your Asset library:
  - Older deployments (earlier than version 10.0.11): Software deployable package
  - New deployments (version 10.0.11 and later): Retail self-service installers

4. Update Commerce headquarters with the new self-service installers for store components:
  - **Older deployments (earlier than version 10.0.11):** Deploy the retail deployable package to Commerce headquarters via LCS.
  - **New deployments (version 10.0.11 and later):** In Commerce headquarters, on the **Commerce parameters** form, on the **Channel deployment** tab, select **Check for package updates**. This update will bring in the packages that are available on the **Retail Self-service package** tab in the LCS Asset library.
5. Assign the client versions to your devices.
6. Download the installers for the desired client type and device.
7. Install in target device.
8. Test and validate.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Software lifecycle policy and cloud releases

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic outlines the lifecycle and support policies for the Finance and Operations online service.

## Modern Lifecycle Policy

The Finance and Operations online service is covered by the Modern Lifecycle Policy. The Modern Lifecycle Policy covers products and services that are serviced and supported continuously. For more information about this policy, see [Modern Lifecycle Policy](#). Licensed customers must stay current with updates to the Finance and Operations online service in accordance with the following servicing and system requirements:

- Customers purchasing subscriptions of Finance and Operations and operating on the following application versions will experience continuous updates of the Platform and Financial Reporting. Microsoft will continually update these components with the option to postpone up to 3 consecutive service updates.
  - Dynamics 365 for Operations version 1611 (November 2016)
  - Dynamics 365 for Finance and Operations, Enterprise edition (July 2017)
  - Dynamics 365 for Finance and Operations, Enterprise edition 7.3
  - Dynamics 365 for Finance and Operations, version 8.0 (April 2018)
- Platform versions maintain backward compatibility with the application versions that are supported at the time of the platform release within the application support lifecycle. For more information about platform versions, see [Cloud platform monthly updates FAQ](#).
- Critical fixes and non-critical updates are handled in the following way:
  - **Critical fixes** – Critical fixes include security fixes and any fixes that are required to adhere to the availability service level agreement (SLA) that the service supports. Critical fixes will be made available in the latest platform update version and in the latest service update for customers operating on version 8.1. In addition, to help protect the customer and the online service, Microsoft might apply critical fixes directly to a customer's environment. If a critical fix must be applied, Microsoft will notify the customer about the required downtime window (if there will be any downtime) and apply the fix to the applicable environment. The critical fix will update the system to the latest update version.
  - **Non-critical updates** – Customers operating on the following application releases must update to the most current Finance and Operations platform and financial reporter version to deploy non-critical updates.
    - Dynamics 365 for Operations version 1611 (November 2016)
    - Dynamics 365 for Finance and Operations, Enterprise edition (July 2017)
    - Dynamics 365 for Finance and Operations, Enterprise edition 7.3
    - Dynamics 365 for Finance and Operations, version 8.0 (April 2018)

Customers operating on release 8.1 must update to the most current service update to deploy non-critical updates.

## NOTE

Application and platform releases expire at the end of the month of their software lifecycle.

Microsoft will not provide any fixes to issues on versions that have reached end of service. Microsoft will also not investigate or troubleshoot any issue that you may encounter on an older version. If you encounter an issue on a version that has reached end of service, you will be required to update to the latest update and report the issue if it persists.

All environments will continue to be operated by Microsoft. All automatic processes around your environments, such as monitoring or self-healing, will also continue as is for supported versions.

## Dates and versions for application and platform releases

**Table 1: Continuous update releases**

For information about the new features included in each release, click the links in the **Version** column.

RELEASE	MAJOR RELEASE OR SERVICE UPDATE	VERSION	BUILD NUMBER	AVAILABILITY	END OF SERVICE
Dynamics 365 for Finance and Operations	Major release	<a href="#">10.0</a>	10.0.8	April 2019	Not applicable (continuously updated)*
Dynamics 365 for Finance and Operations	Major release	<a href="#">8.1</a>	8.1.136	October 2018	Not applicable (continuously updated)*

\* Indicates a major release is required to be updated through service updates. Service updates are cumulative in nature and may include updates for some or all of the following components: Platform, Application, Financial Reporting, Retail, and operating system updates. You will be required to have an update that's no older than 3 service updates. The 8.1.x version series will be replaced by version 10.0, which is targeted for release in April 2019. For more information, see [One Version service updates FAQ](#).

**Table 2: Application releases**

For information about the new features included in each release, select the links in the **Version** column.

RELEASE	MAJOR OR MINOR RELEASE	VERSION	BUILD NUMBER	AVAILABILITY	END OF SERVICE
Dynamics 365 for Finance and Operations	Major release	<a href="#">8.0</a>	8.0.30	April 2018	April 30 2019
Dynamics 365 for Finance and Operations, Enterprise edition	Major release	<a href="#">7.3</a>	7.3.11971.56116	December 2017	April 30 2019*
Dynamics 365 for Finance and Operations, Enterprise edition	Major release	<a href="#">July 2017</a>	7.2.11792.56024	June 2017	April 30 2019

RELEASE	MAJOR OR MINOR RELEASE	VERSION	BUILD NUMBER	AVAILABILITY	END OF SERVICE
Dynamics 365 for Operations	Major release	<a href="#">1611</a>	7.1.1541.3036	November 2016	April 30 2019
Dynamics AX	Minor release	<a href="#">7.0.1</a>	7.0.1265.23014	May 2016	June 2017
Dynamics AX	Major release	<a href="#">7.0</a>	7.0.1265.3015	February 2016	June 2017

\* All customers must be on the latest version of Finance and Operations by April 2019. However, we are making an exception for customers who have unfulfilled [extension requests](#) that have been submitted to Microsoft. Those customers who submitted extensibility requests by January 1, 2019, will be supported on version 7.3 until their extensibility requests are fulfilled. Customers are expected to upgrade to the latest version within 90 days of the extensibility request being fulfilled. For more information, see [One Version service updates FAQ](#).

### Table 3: Platform releases

For information about the new features included in each release, select the links in the **Release** column.

RELEASE	BUILD NUMBER	AVAILABILITY	EXPIRATION DATE
<a href="#">Platform update 31</a>	7.0.5457	January 2020	N/A (Continuously updated)
<a href="#">Platform update 30</a>	7.0.5407	November 2019	N/A (Continuously updated)
<a href="#">Platform update 29</a>	7.0.5372	October 2019	N/A (Continuously updated)
<a href="#">Platform update 28</a>	7.0.5314	July 2019	N/A (Continuously updated)
<a href="#">Platform update 27</a>	7.0.5286	June 2019	N/A (Continuously updated)
<a href="#">Platform update 26</a>	7.0.5257	May 2019	N/A (Continuously updated)
<a href="#">Platform update 25</a>	7.0.5222	April 2019	N/A (Continuously updated)
<a href="#">Platform update 24</a>	7.0.5179	March 2019	N/A (Continuously updated)
<a href="#">Platform update 23</a>	7.0.5126	January 2019	N/A (Continuously updated)
<a href="#">Platform update 22</a>	7.0.5095	December 2018	N/A (Continuously updated / Retired)
<a href="#">Platform update 21</a>	7.0.5073	October 2018	N/A (Continuously updated / Retired)
<a href="#">Platform update 20**</a>	7.0.5030	October 2018	N/A (Continuously updated / Retired)
<a href="#">Platform update 15*</a>	7.0.4841	March 2018	N/A (Continuously updated / Retired)
<a href="#">Platform update 12</a>	7.0.4709	November 2017	November 2018
<a href="#">Platform update 11</a>	7.0.4679.35176	October 2017	October 2018

RELEASE	BUILD NUMBER	AVAILABILITY	EXPIRATION DATE
<a href="#">Platform update 10</a>	7.0.4641.16233	August 2017	August 2018
<a href="#">Platform update 9</a>	7.0.4612.35162	July 2017	July 2018
<a href="#">Platform update 8</a>	7.0.4565.16212	June 2017	June 2018
<a href="#">Platform update 7</a>	7.0.4542.16189	May 2017	May 2018
<a href="#">Platform update 6</a>	7.0.4509.16180	April 2017	April 2018
<a href="#">Platform update 5</a>	7.0.4475.16165	March 2017	March 2018
<a href="#">Platform update 4</a>	7.0.4425.16161	February 2017	February 2018
<a href="#">Platform update 3</a>	7.0.4307.16141	November 2016	November 2017
<a href="#">Platform update 2</a>	7.0.4230.16130	August 2016	August 2017
<a href="#">Platform update 1</a>	7.0.4127.16103	May 2016	May 2017
<a href="#">Platform 7.0</a>	7.0.4030.16079	February 2016	January 2017

\*\* Platform updates 16, 17, 18, and 19 have not been made generally available.

\* Platform updates 13 and 14 have not been made generally available.

#### Table 4: Application updates

The application updates listed below consist of a small subset of application enhancements released on top of Finance and Operations versions 8.0, 7.3, and 7.2 (July 2017). These updates do not affect the support lifecycle of the release--support is in-line with the policies for each release.

Note that application updates are not cumulative. The individual packages only contain the enhancements that were included in that specific release. However, if there is a dependency between two packages, then both packages will be included.

For information about the new features included in each update, click the links in the **Version** column.

RELEASE	VERSION	BUILD NUMBER	AVAILABILITY
Dynamics 365 for Finance and Operations	<a href="#">8.1.3: KB 4470000</a> <a href="#">Microsoft Dynamics 365 for Finance and Operations version 8.1.3 with Platform update 23*</a>	8.1.227	January 2019
Dynamics 365 for Finance and Operations	<a href="#">8.1.2: KB 4470000</a> <a href="#">Microsoft Dynamics 365 for Finance and Operations version 8.1.2 with Platform update 22*</a>	8.1.195	December 2018

RELEASE	VERSION	BUILD NUMBER	AVAILABILITY
Dynamics 365 for Finance and Operations	8.1.1: <a href="#">KB 4470000</a> Microsoft Dynamics 365 for Finance and Operations version 8.1.1 with Platform update 21*	8.1.170	October 2018
Dynamics 365 for Finance and Operations	8.0.4: <a href="#">KB 4458992</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.4 (Binary part)*, <a href="#">KB 4458993</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.4 (X++ part)*	8.0.35.15532	August 2018
Dynamics 365 for Finance and Operations	8.0.3: <a href="#">KB 4346176</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.3 (Binary part)*, <a href="#">KB 4346172</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.3 (X++ part)*	8.0.35.15342	July 2018
Dynamics 365 for Finance and Operations	8.0.2: <a href="#">KB 4340414</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.2 (Binary part)*, <a href="#">KB 4340413</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.2 (X++ part)*	8.0.35.15211	July 2018
Dynamics 365 for Finance and Operations	8.0.1: <a href="#">KB 4295107</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.1 (Binary part)*, <a href="#">KB 4294515</a> Microsoft Dynamics 365 for Finance and Operations - Version 8.0.1 (X++ part)*	8.0.30.15107	June 2018
Dynamics 365 for Finance and Operations, Enterprise edition	7.3.2: <a href="#">KB 4093261</a> Microsoft Dynamics 365 for Finance and Operations - Version 7.3.2 (Binary part)*, <a href="#">KB 4093262</a> Microsoft Dynamics 365 for Finance and Operations - Version 7.3.2 (X++ part)*	7.3.11971.62687	March 2018
Dynamics 365 for Finance and Operations, Enterprise edition	7.3.1: <a href="#">KB 4093139</a> Microsoft Dynamics 365 for Finance and Operations - Version 7.3.1 (Binary part)*, <a href="#">KB 4091727</a> Microsoft Dynamics 365 for Finance and Operations - Version 7.3.1 (X++ part)*	7.3.11971.62430	March 2018

RELEASE	VERSION	BUILD NUMBER	AVAILABILITY
Dynamics 365 for Finance and Operations, Enterprise edition	Application update 5: <a href="#">KB 4053277 Application Update 5 for Microsoft Dynamics 365 for Finance and Operations (Binary part)*</a> , <a href="#">KB 4053278 Application Update 5 for Microsoft Dynamics 365 for Finance and Operations (X++ part)*</a>	7.2.11792.62725	November 2017
Dynamics 365 for Finance and Operations, Enterprise edition	Application update 4: <a href="#">KB 4047325 Application Update 4 for Dynamics 365 for Finance and Operations (Binary part)*</a> , <a href="#">KB 4047321 Application Update 4 for Dynamics 365 for Finance and Operations (X++ part)*</a>	7.2.11792.62509	October 2017
Dynamics 365 for Finance and Operations, Enterprise edition	Application update 3: <a href="#">KB 4043284 Application Update 3 for Dynamics 365 for Finance and Operations (Binary part)*</a> , <a href="#">KB 4043285 Application Update 3 for Dynamics 365 for Finance and Operations (X++ part)*</a>	7.2.11792.62370	September 2017
Dynamics 365 for Finance and Operations, Enterprise edition	Application update 2: <a href="#">KB 4039142 Application Update 2 for Dynamics 365 for Finance and Operations (Binary part)*</a> , <a href="#">KB 4039487 Application Update 2 for Dynamics 365 for Finance and Operations (X++ part)*</a>	7.2.11792.62192	September 2017
Dynamics 365 for Finance and Operations, Enterprise edition	Application update 1: <a href="#">KB 4035749 Application Update 1 for Dynamics 365 for Finance and Operations (Binary part)*</a> , <a href="#">KB 4035751 Application Update 1 for Dynamics 365 for Finance and Operations (X++ part)*</a>	7.2.11792.62089	July 2017

\* The link points to a Knowledge Base (KB) article. You must sign in to Lifecycle Services (LCS) to view the KB article.

## Support matrix

Platform updates are compatible with all application versions that are supported at the time of release.

### Table 5: Downloadable virtual hard drive (VHD) releases

Use of the VHDs is subject to the [Software license terms](#).



RELEASE	VHD NAME	VHD EXPIRATION DATE
Platform update 12 / Application release 7.2	FinandOps7.2PlatUpdate12.vhd	May 24, 2018
Platform update 12 / Application release 7.3	FinandOps7.3PlatUpdate12.vhd	June 05, 2018
Platform update 15 / Application release 7.3	FinandOps7.3withPlatUpdate15	December 08, 2018

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Software lifecycle policy and on-premises releases

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic outlines the lifecycle and support policies for Microsoft Dynamics 365 Finance + Operations (on-premises) releases.

## Modern Lifecycle Policy

Finance + Operations (on-premises) software is covered by the Modern Lifecycle Policy. The Modern Lifecycle Policy covers products and services that are serviced and supported continuously. For more information about this policy, see [Modern Lifecycle Policy](#).

Licensed customers must stay current with updates to the Finance + Operations (on-premises) software in accordance with the following servicing and system requirements. This policy requires that the customer maintain Software Assurance (SA) or the Enhancement Plan, and deploy updates as noted later in this topic. Customers who want to use the Fixed Support Lifecycle Policy (5+5) must downgrade to Microsoft Dynamics AX 2012 R3. If a customer lapses on SA or the Enhancement Plan, the customer will be eligible only for the perpetual license rights to Microsoft Dynamics AX 2012 R3 and must uninstall Microsoft Dynamics 365 for Finance and Operations (on-premises) software.

## On-premises software update policies

The customer is in full control of its on-premises deployments and must follow this policy. The customer is in control of installing updates in its on-premises environments. Microsoft will support the Finance + Operations (on-premises) software through December 31, 2027, at a minimum, but only if the customer keeps the deployed software current according to this policy.

Critical fixes and non-critical updates are handled in the following way:

- **Critical fixes** – Critical fixes include security fixes and any fixes that are required to support reliability and availability. Critical fixes will be made available in the latest platform update version.
- **Non-critical updates** – Customers must update to the most current Finance and Operations platform and financial reporter version to deploy non-critical updates.

## Finance + Operations (on-premises) release dates

### Continuous service updates

As of November 2018, on-premises service updates are released continuously. For more information about version numbers and availability dates, see [Software lifecycle policy and cloud releases](#). For more information about One Version service updates, see [One Version service updates FAQ](#).

### Application releases

Application releases expire at the end of the month of their software lifecycle.

RELEASE	VERSION	BUILD NUMBER	AVAILABILITY	EXPIRATION DATE	PRODUCT LIFE
<a href="#">Continuous service updates</a>	-	-	-	-	-

RELEASE	VERSION	BUILD NUMBER	AVAILABILITY	EXPIRATION DATE	PRODUCT LIFE
Dynamics 365 for Finance and Operations (on-premises)	10.0	10.0.8	March 2019	June 2019	December 2027
Dynamics 365 for Finance and Operations (on-premises)	8.1	8.1.136	November 2018	April 2019	December 2027
Dynamics 365 for Finance and Operations, Enterprise edition (on-premises)	7.3	7.3.11971	March 2018	April 2020*	December 2027
Dynamics 365 for Finance and Operations, Enterprise edition (on-premises)	July 2017	7.2.11792	June 2017	April 2019	December 2027

\* All customers must be on the latest version of Finance and Operations by April 30, 2019. However, we are making an exception for customers who have unfulfilled [extension requests](#) that have been submitted to Microsoft. Those customers can be on version 7.3 until April 2020. For more information, see [One Version service updates FAQ](#).

### Platform releases

Platform releases expire at the end of the month of their software lifecycle.

RELEASE	BUILD NUMBER	AVAILABILITY	EXPIRATION	END OF LIFE
<a href="#">Continuous service updates</a>	-	-	-	-
Platform update 20	7.0.5030	November 2018	February 2019	December 2027
Platform update 15	7.0.4841	June 2018	September 2018	December 2027
Platform update 12	7.0.4709.41182	March 2018	June 2018	December 2027
Platform update 11	7.0.4679.35176	October 2017	April 2018	December 2027
Platform update 10	7.0.4641.16233	August 2017	April 2018	December 2027
Platform update 9	7.0.4612.35162	August 2017	April 2018	December 2027
Platform update 8	7.0.4565.1612	July 2017	April 2018	December 2027

**NOTE**

Platform releases are cumulative in nature. Any fixes, critical or non-critical, will require customers to take the latest available version of the platform.

**Downloadable virtual hard drive (VHD) releases**

Use of the VHDs is subject to the [Software license terms](#).

RELEASE	VHD NAME	VHD EXPIRATION DATE
Platform update 12 / Application release 7.2	FinandOps7.2PlatUpdate12.vhd	May 24, 2018
Platform update 12 / Application release 7.3	FinandOps7.3PlatUpdate12.vhd	June 05, 2018
Platform update 15 / Application release 7.3	FinandOps7.3withPlatUpdate15	December 08, 2018

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Service update availability

2/18/2021 • 5 minutes to read • [Edit Online](#)

Microsoft is committed to delivering predictable service updates. These service updates will be made generally available for self-deployment approximately 2 weeks prior to Microsoft automatically applying the update.

Customers will be able to take up to 8 service updates per year and are required to take a minimum of 2 service updates per year. Customers can choose to pause up to 3 consecutive updates at a time. Pausing a service update can apply to the designated user acceptance testing (UAT) sandbox, production, or both environments. After the pause window has ended and if the customer has not self-updated to a supported service update, Microsoft will auto-apply the latest update based on the configuration selection made available in Lifecycle Services (LCS). To learn more about how to pause service updates, see [Pause service updates through Lifecycle Services](#).

## NOTE

Service updates will not be provided during the months of March, June, September, and December.

## Targeted release schedule (dates subject to change)

## NOTE

Sandbox auto-update takes place 7 days prior to the production update. End of service indicates the date when no new cumulative service updates will be provided.

VERSION	PREVIEW AVAILABILITY (PEAP)	GENERALLY AVAILABLE (SELF-UPDATE)	AUTO-UPDATE SCHEDULE (VIA LCS UPDATE SETTINGS) PRODUCTION START DATE	END OF SERVICE
10.0.20	May 28, 2021	July 16, 2021	July 30, 2021	October 22, 2021
10.0.19	April 23, 2021	June 18, 2021	July 2, 2021	September 17, 2021
10.0.18	March 5, 2021	April 16, 2021	April 30, 2021	July 16, 2021
10.0.17	February 1, 2021	March 19, 2021	April 2, 2021	June 11, 2021
10.0.16	November 20, 2020	January 22, 2021	February 1, 2021	April 30, 2021
10.0.15	October 9, 2020	December 4, 2020	January 1, 2021	March 19, 2021
10.0.14	September 4, 2020	October 23, 2020	November 1, 2020	January 22, 2021
10.0.13	August 3, 2020	September 18, 2020	October 1, 2020	December 4, 2020
10.0.12	May 29, 2020	July 22, 2020	August 1, 2020	October 23, 2020
10.0.11	April 17, 2020	May 29, 2020	July 1, 2020	September 11, 2020

VERSION	PREVIEW AVAILABILITY (PEAP)	GENERALLY AVAILABLE (SELF-UPDATE)	AUTO-UPDATE SCHEDULE (VIA LCS UPDATE SETTINGS) PRODUCTION START DATE	END OF SERVICE
10.0.10	March 6, 2020	April 10, 2020	May 1, 2020	July 3, 2020
10.0.9/Platform update 33	February 1, 2020	March 13, 2020	April 1, 2020	June 5, 2020
10.0.8/Platform update 32	November 29, 2019	January 17, 2020	February 1, 2020	May 1, 2020
10.0.7/Platform update 31	October 25, 2019	November 29, 2019	January 1, 2020	March 9, 2020
10.0.6/Platform update 30	September 6, 2019	October 11, 2019	November 1, 2019	January 13, 2020
10.0.5/Platform update 29	August 2, 2019	September 13, 2019	October 1, 2019	December 2, 2019
10.0.4/Platform update 28	June 7, 2019	July 12, 2019	August 1, 2019	October 14, 2019
10.0.3/Platform update 27	May 10, 2019	June 14, 2019	July 1, 2019	September 9, 2019
10.0.2/Platform update 26	April 12, 2019	May 17, 2019	June 1, 2019	August 12, 2019
10.0.1/Platform update 25			May 1, 2019	June 10, 2019

#### NOTE

The [Software lifecycle policy](#) applies to customers enrolled in First Release and when the service update is made generally available.

Sign up for the PEAP program by joining the Insider Program available at <https://experience.dynamics.com>. After your nomination has been accepted, join the program.

Public previews are made available as a deployable package via the Shared Asset Library in Lifecycle Services. For more details, see [One Version service updates FAQ](#).

## Service update overview

Service updates are continuous, touchless updates that provide new features and functionality. They eliminate the need to do expensive upgrades every few years. The service updates maintain backward compatibility, which means there is no need to 'merge your code'. We recommend leveraging tools such as the Regression suite automation tool (RSAT) for regression testing.

You are in control and manage how your organization receives these updates. For example, you can sign up for the First Release program so that your organization receives updates first. You can apply the updates to any of

your environments manually (self-update) or remain on the default release schedule and receive the auto-updates when you schedule them using LCS.

Service updates contain both application and platform changes that are critical improvements to the service, including regulatory updates.

## Release processes

Each new release is designed and developed by the Dynamics 365 team. Any new release is first validated by the feature team, then by the Finance and Operations teams. During this time, extensive testing is done on various test topologies. A compatibility checker also runs tests to ensure backward compatibility. In addition, a [Release Validation Program](#) is available for customers to join. This program allows customers to share artifacts, such as databases and code, that is used for benchmarking and tested with automation to provide an additional layer of quality assurance.

Preview Early Access Program (PEAP) is available to partners, customers, and ISV's who opt in through the [PEAP Survey](#). As a participant in the PEAP program you will have first access and visibility into the preview for the upcoming service update. The preview service update is used to validate customizations, learn about new features, and provide feedback to Microsoft. During this phase, the service update must be deployed on a Dev/Test environment. This release cannot be used in production. To join the PEAP program, sign up via the [PEAP Survey](#).

The First Release program is open to all customers. Customers who join the First Release program will be the first, select group of customers to take the service update all the way to production. Microsoft will manage the deployment of this service update to a UAT sandbox and then 7 days later will auto-deploy the update to production. Customers participating in this program have the additional benefit of having dedicated Microsoft engineers closely monitoring the environments for any issues after updates have been applied. To join First Release, sign up via the [First Release Survey](#).

The service update will be made generally available using the action center in LCS. When the service update is available, it can be manually applied to all environments including production. If the service update has not been applied to the designated sandbox or production environment, Microsoft will auto-apply the update based on the Update settings for the LCS project. To learn more, see [Configure service updates through Lifecycle Services](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Apply updates to cloud environments

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic describes how you can use Microsoft Dynamics Lifecycle Services (LCS) to automatically apply updates to cloud environments.

## IMPORTANT

Updates are applied using deployable packages. Applying updates causes system downtime. All relevant services will be stopped, and you won't be able to use your environments while the package is being applied. You should plan accordingly.

## Supported environments

All environments deployed through Lifecycle Services are supported.

## NOTE

If you have a build environment, you can only use LCS to apply Binary updates and Data upgrade packages. You can't use LCS to apply an Application Deployable package.

For other environments (listed below), you must use Remote Desktop Protocol (RDP) to connect to the environment and install from the command line. For information about manual package deployment, see [Install deployable packages from the command line](#).

- Local development environments (Downloadable virtual hard disk [VHD])
- Multi-box dev/test environments in Microsoft Azure (Partner and trial projects)

## Key concepts

Before you begin, you should understand *deployable packages*, *runbooks*, and the *AXInstaller*. A deployable package is a unit of deployment that can be applied in any environment. A deployable package can be a binary update to the platform or other runtime components, an updated application (AOT) package, or a new application (AOT) package. The AXInstaller creates a runbook that enables installing a package. For more details, see [Packages, runbooks, and the AXUpdateInstaller in depth](#) at the end of this topic.

## Supported package types

- **AOT deployable package** – A deployable package that is generated from application metadata and source code. This deployable package is created in a development or build environment.
- **Application and Platform Binary update package** – A deployable package that contains dynamic-link libraries (DLLs) and other binaries and metadata that the platform and application depend on. This is a package released by Microsoft. This is available from the **All binary updates** tile from LCS.
- **Platform update package** – A deployable package that contains dynamic-link libraries (DLLs) and other binaries and metadata that the platform depend on. This is a package released by Microsoft. This is available from the **Platform binary updates** tile from LCS.
- **Commerce deployable package** – A combination of various packages that are generated after the Commerce code is combined.
- **Merged package** – A package that is created by combining one package of each type. For example, you can



merge one binary update package and one AOT package, or one AOT package and one Commerce deployable package. The packages are merged in the Asset library for the project in LCS.

#### NOTE

A binary package and a Commerce deployable package can't be included in the same merged package.

For information about how to download an update from LCS and what you see in the tiles based on your environment version, see [Download updates from Lifecycle Services \(LCS\)](#).

If your environment is on an application version 8.1 and later, then the **Platform Update package** does not apply to your environment. Starting with 8.1 and later releases, **Application and Platform Binary update package** is the one that applies since application and platform will be combined into a single cumulative package and will be released by Microsoft. Also note that you will no longer be applying granular X++ hotfixes and will get all application and platform updates together. This means that on the environment details page, clicking on **View detailed version information** will not have details on the granular hotfixes or KBs applied as there is no way to apply them.

## Prerequisite steps

- **Make sure that the package that should be applied is valid.** When a package is uploaded to the Asset library, it isn't analyzed. If you select the package, the package status appears in the right pane as **Not Validated**. A package must pass validation before it can be applied in an environment by using the following procedures. The status of the package will be updated in the Asset library to indicate whether the package is valid. We require validation to help ensure that production environments aren't affected by packages that don't meet the guidelines.

There are three types of validations:

- Basic package format validations
  - Platform version checks
  - Types of packages
- **Make sure that the package is applied in a sandbox environment before it's applied in the production environment.** To help ensure that the production environment is always in a good state, we want to make sure that the package is tested in a sandbox environment before it's applied in the production environment. Therefore, before you request that the package be applied in your production environment, make sure that it has been applied in your sandbox environment by using the automated flows.
  - **If you want to apply multiple packages, create a merged package that can be applied first in a sandbox environment and then in the production environment.** Application of a single package in an average environment requires about 5 hours of downtime. To avoid additional hours of downtime when you must apply multiple packages, you can create a single combined package that contains one package of each type. If you select a binary package and an application deployable package in the Asset library, a **Merge** button becomes available on the toolbar. By clicking this button, you can merge the two packages into a single package and therefore reduce the total downtime by half.
  - **Make sure that the Application binary update package is applied to your dev/build environment before it is applied to your sandbox and production environment** - If the application binary package is applied directly to your Tier 2+ sandbox environment but is not applied on your dev/build environment, the next time you move an AOT package from dev/build box (which does not have the same application binaries as your sandbox environment) to sandbox, some of the application binaries will be overwritten with what is in your dev/build environment. This could result in a regression of the version of your sandbox environment.

# Apply a package to a non-production environment by using LCS

Before you begin, verify that the deployable package has been uploaded to the Asset library in LCS.

1. For a binary update, upload the package directly to the Asset library. For information about how to download an update from LCS, see [Download updates from Lifecycle Services \(LCS\)](#). For an application (AOT) deployable package that results from an X++ hotfix, or from application customizations and extensions, create the deployable package in your development or build environment, and then upload it to the Asset library.
2. Open the **Environment details** view for the environment where you want to apply the update.
3. Click **Maintain > Apply updates** to apply an update.
4. Select the package to apply. Use the filter at the top to find your package.
5. Click **Apply**. Notice that the status in the upper-right corner of the **Environment details** view changes from **Queued** to **In Progress**, and an **Environment updates** section now shows the progress of the package. You can refresh the page to check the status.
6. Continue to refresh the page to see the status updates for the package application request. When the package has been applied, the environment status changes to **Deployed**, and the servicing status changes to **Completed**.

# Apply a package to a production environment by using LCS

In a production environment, customers can schedule a downtime for when they want the update to be applied.

## IMPORTANT

An important prerequisite for applying a package to a production environment is that the package must be successfully applied to at least one sandbox environment in the same project.

1. After the update is successfully applied in a sandbox environment, go to the project's asset library. On the **Asset library** page, select the **Software deployable package** tab, select the package that you want to move to production, and click **Release candidate**. This indicates that this package is ready for production deployment.
2. Open the **Environment details** view for the production environment where you want to apply the package.
3. Select **Maintain > Apply updates** to apply the package.
4. Select the package to apply in your production environment, and then click **Schedule** to submit a request to apply it.

## NOTE

The list of packages includes only the packages that have been successfully signed off in the sandbox environment, and that have been marked as release candidates.

5. Specify the date and time to schedule the package application. Click **Submit**, and then click **OK** to confirm. Note that your environments will be unavailable to perform business while the package is being applied.
6. At the scheduled downtime, package deployment will start.
7. After the environment is serviced, you can monitor the status. The **Servicing status** field indicates the status of package application. Additionally, a progress indicator shows the number of steps that have

been run, out of the total number of steps that are available.

8. After the deployment is successfully completed, the **Servicing status** field is set to **Completed**.
9. If package application isn't successfully completed, Microsoft will investigate the issue. The **Servicing status** field will indicate that package application has failed. The environment will be rolled back to a good state.

## Troubleshoot package deployment failures

If package deployment fails, see the [Troubleshoot package application issues](#) topic.

## Applying updates and extensions

If you are updating a Tier-2 Sandbox or Production environment on application version 8.1.2.x or newer and have initialized Cloud Scale Unit, you will also need to update Commerce channel components. For more information, see [Update Retail Cloud Scale Unit](#).

If you're using components (such as Modern POS), after you've applied updates and extensions in your environment, you must also update your in-store components. For more information, see [Configure, install, and activate Modern POS \(MPOS\)](#).

## Packages, runbooks, and the AXUpdateInstaller in depth

Deployable packages, runbooks, and the AXUpdateInstaller are the tools you use to apply updates.

**Deployable package** – A deployable package is a unit of deployment that can be applied in an environment. A deployable package can be a binary update to the platform or other runtime components, an updated application (AOT) package, or a new application (AOT) package. Deployable packages downloaded from LCS or created in a development environment cannot be applied across product types. For example, a Finance deployable package cannot be applied in a Commerce app environment, and vice versa. If you have an existing customization for a Finance and Operations app that is compatible with the Commerce app, and you would like to apply it to a Commerce environment, you will need to re-package your source code in a Commerce development environment, and conversely if moving in the other direction.

AXDeployablePackage\_20160212\_22\_57\_44.zip - ZIP archive, unpacked size 1,221,43; ← Zip format

Name	Size	Pack...	Type
ALMService			File folder
AOSService			File folder
BIService			File folder
DevToolsService			File folder
DIXFService			File folder
MRApplicationService			File folder
MROneBox			File folder
MRProcessService			File folder
PerfSDK			File folder
ReportingService			File folder
RetailCloudPos			File folder
RetailSDK			File folder
RetailSelfService			File folder
RetailServer			File folder
RetailStorefront			File folder
SCMSelfService			File folder
TestAssets			File folder
UserSID			File folder
AutoTriggerETWManifestRefresh.ps1	10,1...	6,065	Window...
AXUpdateInstaller.exe	18,6...	9,365	Applicati...
DefaultServiceModelData.xml	13,4...	724	XML Do...
DefaultTopologyData.xml	1,199	430	XML Do...
HotfixInstallationInfo.xml	2,999	443	XML Do...
Microsoft.Dynamics.AX.AXInstallationInfo.dll	26,3...	12,7...	Applicati...
Microsoft.Dynamics.AX.AXUpdateInstallerBase.dll	42,7...	18,7...	Applicati...
Switch.dll	40,6...	18,9...	Applicati...
System.Management.Automation.dll	2,68...	896...	Applicati...

← Changed files/ folder

← Update Installer

← Modules information

← Topology information

**Runbook** – The deployment runbook is a series of steps that are generated in order to apply the deployable package to the target environment. Some steps are automated, and some steps are manual. AXUpdateInstaller lets you run these steps one at a time and in the correct order.

- **Generated based on topology of deployments with multiples VMs**
- **Contains step by step information for applying deployable package**
- **Provides sequence of steps across VMs in multi-box/ HA environment**
- **Integration for apply automation scripts at each step**
  - **Stop/ start AOS service, batch service**
  - **Report deployment, DB sync, ...**

```
<?xml version="1.0" encoding="UTF-8"?>
<RunbookData xmlns:xsi="http://www.w3.
  <RunbookID>AXDeployablePackage_20
  + <RunbookTopologyData>
  + <RunbookServiceModelData>
  + <RunbookStepList>
  + <RunbookLogs>
</RunbookData>
```

```
<Name>AX topology</Name>
<MachineList>
  - <Machine>
    - <Name>AOS-77edc66f7a1</Name>
    - <ServiceModelList>
      <string>AOSService</string>
      <string>DIXFService</string>
      <string>RetailCloudPos</string>
      <string>RetailSelfService</string>
      <string>RetailServer</string>
      <string>SCMSelfService</string>
    </ServiceModelList>
  </Machine>
  + <Machine>
  - <Machine>
    - <Name>BI-4bb1b0a48fa5</Name>
    - <ServiceModelList>
      <string>ReportingService</string>
    </ServiceModelList>
  </Machine>
  - <Machine>
    - <Name>BI-3c0207c4482e</Name>
```

```
- <Step>
  <ID>1</ID>
  <Description>Stop script for service model: AOSService on machine: AOS-77edc66f7a1</Description>
  <MachineName>AOS-77edc66f7a1</MachineName>
  <ServiceModelName>AOSService</ServiceModelName>
  - <ScriptToExecute>
    <FileName>AutoStopAOS.ps1</FileName>
    <Automated>true</Automated>
    <Description>Stop AOS service and Batch service</Description>
    <RetryCount>0</RetryCount>
  </ScriptToExecute>
  <StartTime>2016-02-17T01:14:45.2397318+00:00</StartTime>
  <EndTime>2016-02-17T01:14:48.6772116+00:00</EndTime>
  <StepState>Completed</StepState>
</Step>
+ <Step>
```

**AXUpdateInstaller** – When you create a customization package from Microsoft Visual Studio or a Microsoft binary update, the installer executable is bundled together with the deployable package. The installer generates the runbook for the specified topology. The installer can also run steps in order, according to the runbook for a specific topology.

## Additional resources

## Install deployable packages from the command line

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Apply updates to on-premises deployments

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains how to apply supported updates to Dynamics 365 Finance + Operations (on-premises). All updates to on-premises environments are done through Microsoft Dynamics Lifecycle Services (LCS).

## Search for and download updates

For more information about how to find the updates that you can apply to your on-premises environment, see [Issue search in Lifecycle Services \(LCS\)](#). For information about how to download updates from the tiles in the **Updates** section of the **Environment details** page in LCS, see [Download updates from Lifecycle Services \(LCS\)](#).

### NOTE

When you are updating an on-premises environment, always select updates from the update tiles on the **Environment details** page. If you select updates from another location, the updates might not work.

## Update an on-premises deployment

You can apply updates to an on-premises environment either during deployment or after the deployment is completed.

While an on-premises environment is being deployed, you can select to deploy a custom package in the **Advanced** settings. For more information about how to apply customizations or application X++ updates, see [Develop and deploy custom models to on-premises environments](#).

To apply updates to an on-premises environment after it has been deployed, in LCS, on the **Environment details** page for the environment, under **Maintain**, select **Apply updates**.

### NOTE

You can apply updates after deployment only on environments that have Platform update 12 for Finance and Operations or later. The environment must also have the latest version of the local agent available in LCS. For more information, see [Update the local agent](#). If you're on a platform version that is older than Platform update 12, you can reconfigure an environment that is already deployed to update the customizations or update to the latest platform release. For more information about how to redeploy an environment, see [Redeploy on-premises environments](#).

## Apply application or binary updates through LCS

The following steps can be used to apply X++, All Binary, or Platform binary updates.

### IMPORTANT

The application of updates requires downtime for your environment. Therefore, no business transactions can be performed in the environment during the update. When you complete the following steps, verify that the system isn't being used, and that an official downtime notice has been communicated to all system users.

## IMPORTANT

To move to the latest platform, always select the platform update from the **Platform Binary Updates** tile on the **Environment details** page. If you select updates from another location, the updates might not work.

### Prerequisites

- Before you begin, complete a full backup of the Management Reporter (MR), Microsoft Dynamics AX, and Microsoft SQL Server Reporting Services (SSRS databases). Although the code is restored through LCS, the database must be manually restored to help guarantee that there is no data loss.
- Update your environment to the latest build of Platform update 12.
- Update the local agent to the latest version. For more information, see [Update the local agent](#).
- Depending on the type of update, complete the following steps to generate a deployable package:
  - **Platform binary updates** – Download or save the update directly to the Asset library in LCS by following the steps in [Download updates from Lifecycle Services \(LCS\)](#).
  - **Application binary updates** – Download or save the update directly to the Asset library in LCS by following the steps in [Download updates from Lifecycle Services \(LCS\)](#).
  - **Application X+ + updates** – Download the required hotfix to your development environment, and then follow the steps in [Create deployable packages of models](#).
  - **Customizations** – Follow the steps in [Develop and deploy custom models to on-premises environments](#).

### Update a sandbox environment

1. In the LCS Asset library, upload the deployable package that was generated in the "Prerequisites" section of this topic to the **Software deployable packages** tab.
2. In LCS, open the on-premises implementation project, and then open the **Environment details** page of the environment to update.
3. Under **Maintain**, select **Apply updates**. A slider shows the updates that were uploaded to the Asset library. Note that only packages that are marked as **Valid** in the Asset library appear.

If you are on local agent version 2.1.0 and higher, complete the following steps.

1. Select the update, and then click **Prepare**. Clicking on **Prepare** will prepare your on-premises environment for servicing.

#### NOTE

During preparation, the environment state will be **Deployed** but the Deployment status field will show the progress of Preparation. Steps such as formatting the package and downloading the package are executed during preparation. The environment is not directly touched during preparation and hence there is no downtime during the preparation phase. Users can continue to use the system during preparation.

2. After the preparation is complete, you will see **Abort** and **Update Environment** buttons. To start applying the update, click **Update Environment**. If preparation fails, see the "Resolve a failed update application" section later in this topic.
3. In the confirmation message, select **Yes**. The servicing operation has started on this environment. This is the start of the downtime on your environment.
4. The environment state is changed from **Deployed** to **Deploying**.
5. After the update is completed, the environment state is changed back to **Deployed**. If application of the

update fails, the environment state is changed to **Failed**. For information about what to do if package application fails, see the "Resolve a failed update application" section later in this topic.

6. Open the **History** and **Environment details** pages to view the operations that were performed on the environment. You can also view a record of major actions that were performed on the environment, such as deployments, servicing, and rollbacks.

**If you are on local agent version lower than 2.1.0, complete the following steps.**

1. Select the update, and then click **Apply**.
2. In the confirmation message, select **Yes**. The servicing operation has started on this environment. This is the start of the downtime on your environment.
3. Environment state changes from **Deployed** to **Preparing**.

#### **NOTE**

During preparation, steps such as formatting the package and downloading the package are executed during preparation. The environment is not directly touched during preparation and hence there is no downtime during the preparation phase. Users can continue to use the system during preparation. However, we recommend that the downtime starts when the environment enters the **Preparing** state.

4. After preparation is complete, the environment state is changed from **Preparing** to **Deploying**.
5. After the update is completed, the environment state is changed back to **Deployed**. If application of the update fails, the environment state is changed to **Failed**. For information about what to do if package application fails, see the "Resolve a failed update application" section later in this topic.
6. Open the **History** and **Environment details** pages to view the operations that were performed on the environment. You can also view a record of major actions that were performed on the environment, such as deployments, servicing, and rollbacks.

### **Update a production environment**

Before you update a production environment, you must successfully complete the package application update on a sandbox environment.

1. In the project for the sandbox environment that you applied the package to, open the Asset library, and then, on the **Software deployable packages** tab, select the package, and mark it as a **Release candidate**.
2. On the **Environment details** page, under **Maintain**, select **Apply updates**. In the dialog box, only packages that are marked as a **Release candidate** are shown.
3. Select the Release candidate package to be applied to the Production environment.
4. The rest of the Update flow is the same as that of a sandbox environment. Your update experience will differ based on the version of the local agent running on your environment. We recommend that you always run with the latest version.

## Resolve a failed update application

When preparation fails, the environment state is **Deployed**. When the application of an update fails, the environment state is **Failed**. The first step is to determine why there is a failure. The location of the logs varies, depending on the stage where the failure occurred:

- **Preparation stage:** If the operation fails during the **Preparation** stage, the logs are uploaded to LCS. In the log files, select **Download logs** to download the log files. If the package has any merge issues, the error is included in the log file.
- **Deploying stage:** If the operation fails during the **Deploying** stage, the logs are located in the on-premises



environment. You must sign in to the environment, and then access the logs and event viewer.

For more information about how to use the troubleshooting logs, see [Troubleshoot on-premises deployments](#).

After you review the logs and determine the cause of the failure, complete one of the following operations to restore the environment to a healthy state. No actions can be performed on an environment that is in a **Failed** state. The environment must first be restored to a healthy state.

- **Retry failed operation** – If update application fails, select **Retry** to recover from the failed operation.
- **Abort failed operation** – Because there is no change made to the on-premises environment, if the preparation fails, you have the option to cancel the operation. Select **Abort** to cancel the preparation.
- **Roll back the update** – To roll back the update that failed, select **Rollback**. Before you start the rollback, you must restore the database to the last known good state. When you select **Rollback**, the environment is restored to the last known good state. The environment state is then changed to **Preparation**, then to **Deploying**, and then to either **Deployed** or **Failed**.

#### NOTE

The **Rollback** button doesn't roll back the database. You're responsible for restoring the database to the last known backup that was made before update application. This step is critical to help guarantee that there is no data loss.

- **Refresh the state** – If update application fails during the **Preparation** stage, the failure is on the LCS side, and update application hasn't yet started. Therefore, the on-premises environment is in a good state. To restore the LCS environment state to **Deployed**, on the project dashboard page, select **Refresh**.
- **Delete and redeploy an environment** – If the retry and rollback options don't work, you must delete and redeploy the environment. To delete the environment, on the project dashboard page, select **Delete**. You then see the option to configure the environment.

#### IMPORTANT

This option should **not** be used on a production environment. However, it can be used on a sandbox deployment to restore the environment to a healthy state.

Because this option requires that you do a fresh deployment of the environment, you lose any updates that were previously applied. Any customizations and binary updates must be reapplied to the environment.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure service updates through Lifecycle Services (LCS)

2/18/2021 • 4 minutes to read • [Edit Online](#)

In Microsoft Dynamics Lifecycle Services (LCS), you can specify how and when you receive service updates from Microsoft for your environments.

## IMPORTANT

This feature is available only to customers who are using **version 8.1 and later** or are using **version 7.3**, and who are **not** part of the [First release](#) program. Microsoft is working to make the feature available to First release customers. For customers who are on version 7.1, 7.2, or 8.0, you can take the update manually using the regular servicing flows.

Only users (customers or partners) who are assigned to the **Project owner** role in LCS can configure updates. Additionally, updates can be configured only for **implementation projects**.

Follow these steps to change your update settings.

## NOTE

These settings apply only to service updates. They have no effect on the operating system-level security updates that are applied to your environments every month.

1. In LCS, in your implementation project, open the **Project settings** page.

This page has a new tab that is named **Update settings**.

2. On the **Update settings** tab, set the following configuration options as you require:

- **Update environment** – Select an alternate sandbox environment that should be updated before the production update.

By default, Microsoft first updates the Tier 2 Standard Acceptance Test (sandbox) environment that is included in the base offer. It then applies the update to the production environment. If you've purchased Tier 2 and higher sandbox add-on environments and want a different sandbox to be updated, you can use this field to change the default environment to an alternate environment.

## IMPORTANT

The environment that you select here will be updated seven calendar days before the update cadence that is selected for the production environment.

- **Production environment update cadence** – Select a recurring cadence for updates to your production environment. The sandbox environment that is selected in the **Update environment** field will be updated seven calendar days before the selected cadence. The following options are available:
  - **Select the cadence** – Select whether to receive updates in the first, second, or third week of the month.
  - **Select one of the three time-zones** – Select the time zone that the production environment

should be updated in: Eastern Time (UTC – 5), Hong Kong Time (UTC + 8), or Greenwich Mean Time (UTC + 0).

- **Select a day of the week:** Select the day in the week when you want to receive updates.
- **Select a time slot:** Select the time slot when you want to receive updates.

#### NOTE

Currently, only a few options are available for the day of the week and time slot options. Microsoft will add more options soon, such as weekdays for customers.

#### IMPORTANT

If the above time slots do not meet your needs, you always have the option to do a self-update at a time that is convenient to you by taking the update and applying it to your environments using the regular servicing flows.

3. When you've finished setting the configuration options, select **Save**.

After you set the update environment and update cadence, Microsoft generates an update calendar for the next six months. This calendar shows exactly when the configured sandbox and production environments will be updated. Therefore, you will know when to expect each update. To view the calendar, select the **View update calendar** link under the **Production environment update cadence** options.

#### IMPORTANT

After the settings are saved, you can change them at any time. However, if there is an ongoing rollout, the new settings won't be used to update the existing rollout timings. Instead, they will start to be used in the next rollout. An ongoing rollout is defined by the 14-day period between the date when the email notification about the update of the sandbox environment is sent and the date when the production environment is updated.

For more information about how to pause updates to configured sandbox and production environments, see [Pause service updates through Lifecycle Services \(LCS\)](#).

For more information about One Version and Microsoft-managed service updates, see [One Version service updates FAQ](#).

## Canceled updates

A scheduled update could be canceled for various reasons. Here are some of the common reasons that could cause a scheduled update to be canceled by Microsoft.

- An error was found during update preparation. The update preparation starts approximately 4 hours before the update to ensure that the environment is in a healthy state. If the environment was in a failed state or maintenance mode, the scheduled update will be canceled before it starts.
- An error was found while updating the environment. If there were issues during the update, the scheduled update will be canceled and the environment is rolled back to the previous state.
- The environment is already running on the latest version. There's no need to apply the update again, the scheduled update will be canceled before it starts.
- The target environment is not found. If the designated sandbox was deleted or the production environment has not been deployed, the scheduled update will be canceled before it starts.
- You're enrolled in the [First Release program](#). The First Release program has different release cadence so the previously scheduled updates will be canceled.

You can find the canceled updates via the **View recent canceled updates** in the update settings. It will show all canceled updates, if any, within the last 2 scheduled updates.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Pause service updates through Lifecycle Services (LCS)

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how to pause updates to your sandbox and production cloud environments by using Microsoft Dynamics Lifecycle Services (LCS). This topic does not apply to on-premises environments.

## IMPORTANT

This feature is available only to customers who are using **version 8.1 and later** or are using **version 7.3**, and who are **not** part of the [First release](#) program. Microsoft is working to make the feature available to First release customers. For customers who are on version 7.1, 7.2, or 8.0, you can take the update manually using the regular servicing flows.

Microsoft updates your configured sandbox and production environments to the latest service update that Microsoft has released. Microsoft notifies you about upcoming updates to your environments via email and through notifications in LCS. At that point, if you can't proceed with the update for some reason, you can pause it through LCS.

For more information about how to change the configured sandbox environment and set the production update cadence, see [Configure service updates through Lifecycle Services \(LCS\)](#).

## Who can pause service updates?

Only users (customers or partners) who are assigned to the **project owner** role in LCS can pause updates. Additionally, updates can be paused only for **implementation projects**.

Staying current with service updates helps guarantee that customers always run on the latest set of fixes that Microsoft has released, so that they have the best service experience. Therefore, Microsoft doesn't allow updates to be paused indefinitely.

You can't use LCS to pause updates if you're three or more updates behind the latest update that Microsoft has released. For example, if the latest update that Microsoft has released is version 10.0.0, customers who are on version 8.1.3, version 8.1.2, and version 8.1.1 **can** pause updates. However, customers who are on version 8.1.0 **can't** pause updates, because they are more than three updates behind. Customers who are on version 7.3 can get only platform updates. For example, if the last platform update that Microsoft has released is Platform update 25, customers who are on Platform update 24, Platform update 23, and Platform update 22 **can** pause updates. However, customers who are on Platform update 21 **can't** pause updates.

## What can I pause?

If you decide to pause updates, you have two options:

- Pause updates only to your production environment.
- Pause updates to both your sandbox environment and your production environment.

You can pause a maximum of three continuous updates at a time. For example, if you're using version 8.1.3, you can pause update version 10.0.0, 10.0.1 and 10.0.2. However, you can't pause update version 10.0.3. In addition, in the month of June, you can pause the next three updates. However, you will not be able to pause updates scheduled for October, November, December and later. Similarly, for customers on version 7.3 for platform only updates, if you're using Platform update 23 then you can pause update 24, update 25, and update 26, but you

cannot pause update 27. We will be releasing 8 updates in a year. We require you to take at least two updates in a year.

#### **IMPORTANT**

There is no way to pause more than three updates, regardless of your industry or business schedule. If you are more than three updates behind and you find a critical issue during validations in your sandbox environment after the update, you can contact Microsoft Support to pause the update to your production environment. This is only required if you are more than three updates behind and you are unable to use the pause updates functionality available in LCS to pause the update to production.

If you pause updates to your sandbox environment, updates are automatically also paused for your production environment, because Microsoft always updates configured sandbox environments before production environments.

## How do I pause updates?

To pause updates, follow these steps.

1. In LCS, in your implementation project, open the **Project settings** page.  
  
This page has a new tab that is named **Update settings**.
2. On the **Update settings** tab, set the **Pause updates** option to **ON**.
3. Select **Edit settings**.
4. In the dialog box that appears, select whether you want to pause updates to your production environment only, or to both your sandbox environment and your production environment.
5. Select **Next**.
6. Select your reason for pausing updates. If you select **Issue found during validations**, you must enter a valid support ticket number. You can add any additional details that will help Microsoft understand why you want to pause updates.
7. When you've finished, select **Confirm**.

You can also edit an existing pause. You can either extend the duration of the pause, so that updates are paused for a longer time, or cancel it, so that updates are resumed. To edit a pause, select **Edit settings**. The limitations about the number of updates that you can pause still apply.

To cancel a pause and resume updates to your environments, set the **Pause updates** option to **OFF**.

Any time that you pause updates or edit an existing pause, a notification appears at the top of the **Update settings** tab. This notification shows what has been paused. An email is also sent to all stakeholders (the project owner and environment manager), to notify them that service updates for the selected environments have been paused. If someone cancels an existing pause and resumes updates, the notification disappears, and an email is sent to inform the stakeholders that updates have resumed.

#### **IMPORTANT**

You can pause updates through LCS until four hours before the start of the downtime window.

You can cancel a pause and choose to resume updates only 7 days prior to the start of the downtime date. If you are past that date then you will not be able to cancel a pause.

## What happens after the pause duration expires?

Cumulative service updates help guarantee that customers always run on the latest set of fixes that Microsoft has released, so that they have the best service experience. Therefore, Microsoft doesn't allow updates to be paused indefinitely.

There are two ways to cancel pauses, so that updates are resumed:

- Someone manually cancels an ongoing pause, as explained in the previous section.
- The duration that was set for the pause expires, and updates to the configured environments are automatically resumed.

In both cases, an email is sent to inform the stakeholders.

For more information about service updates, see [One Version service updates FAQ](#).

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Get notified about service updates through Lifecycle Services (LCS)

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how you can stay up to date about service updates from Microsoft.

Microsoft uses service updates to update your configured sandbox and production environments to the latest released version. Microsoft notifies you about upcoming updates to your environments via email and through notifications in Microsoft Dynamics Lifecycle Services (LCS).

Here are the different types of notifications that you will receive:

- **Notification when an update is made available:** When a new release is made generally available, Microsoft surfaces a notification in your implementation projects' action center. You can then save that update in your projects' asset library, if you want to apply the update to your environments before Microsoft does an automatic update. When Microsoft does an automatic update, it also saves a copy of the update in your projects' asset library.
- **Notification that is sent five days before the update:** Microsoft notifies you five days before it updates your environment. After you've configured your update cadence, you will receive notifications about upcoming updates five days before they occur. These notifications take three forms:
  - **Email notification:** Project owners, environment managers, and users who are listed as additional stakeholders for an environment are notified by email about the upcoming update.
  - **Notification bar on the environment details page:** A notification that appears on the environment details page in LCS informs the customer about the upcoming update.
  - **Upcoming update reflects the update:** On the environment details page in LCS, select **Maintain > Upcoming Update** to open a dialog box that contains details about the upcoming update.
- **Notification that is sent one hour before the update:** One hour before the start of the downtime window, users in the application receive a notification. This notification asks users to save their work, because the environment will be taken down for an update.
- **Notification that is sent when the update is completed:** After Microsoft has finished updating your configured environment, it notifies you by email about the outcome of the update. This email is always sent, regardless of whether the update was successfully applied. It's sent to project owners, environment managers, and users who are listed as the additional stakeholders for the environment. If Microsoft can't start the update for some reason, the email includes a reason to explain why the update wasn't started.

After you receive a notification, if you can't proceed with the update for some reason, you can pause it. For more information about how to pause updates to configured sandbox and production environments, see [Pause service updates through Lifecycle Services \(LCS\)](#).

For more information about One Version and Microsoft-managed service updates, see [One Version service updates FAQ](#).

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Data task automation

2/18/2021 • 17 minutes to read • [Edit Online](#)

Data task automation lets you easily repeat many types of data tasks and validate the outcome of each task. Data task automation is very useful for projects that are in the implementation phase. For example, you can automate the creation and configuration of data projects. You can also configure and trigger the execution of import/export operations, such as the setup of demo data and golden configuration data, and other tasks that are related to data migration. You can also create automated testing of data entities by using task outcome validation.

## IMPORTANT

Data task automation isn't currently supported for on-premises environments. The user who executes data task automation must be in the same tenant as the application environment and LCS project.

We recommend the following approach for data task automation.

### 1. Identify the data-related tasks that will benefit from automation.

We recommend that implementation teams review their configuration management plan and data migration plan to identify potential data tasks for automation, and also to identify data entity test cases.

### 2. Define tasks.

Tasks are defined in an XML manifest. You can keep your manifest under source control as part of configuration management in your application lifecycle management (ALM) strategy.

### 3. Put the data packages that are related to data task automation in the Shared asset library of Microsoft Dynamics Lifecycle Services (LCS). You can also use a specific LCS project as you require.

Data task automation manager can consume packages from any sandbox and/or production environment that is related to the LCS project.

## IMPORTANT

- The user account that runs Data task automation manager must have access to LCS and to the LCS project that is referenced in the manifest for data packages.
- Although data task automation can be run in any environment in the cloud, we strongly recommend that you not run any import/export tasks that use integration application programming interfaces (APIs) in a production environment. Data task automation that involves integration APIs should be used only for automated testing.

### 4. Run the data tasks, and then review the outcomes.

Data task automation manager provides the success or failure outcome for each task. It also provides insights into the reason why a task failed.

## IMPORTANT

Although data task automation can be run in any environments in the cloud, we recommend that you not run any import/export tasks that use integration APIs in a production environment. Data task automation that involves integration APIs should be used only for automated testing.

The following video is a 55-minute TechTalk that walks you through an early release of Data task automation manager: [Task automation framework](#).

## Task manifest

A task must be defined in an XML manifest. This section describes the manifest. For guidance about how to name and design the manifest, see the "Best practices for manifest design" section later in this topic.

### Manifest root

The `<TestManifest>` element is the root of the manifest. All other elements are children of this element.

```
<?xml version='1.0' encoding='utf-8'?>
<TestManifest name='Data management demo data set up'>
  <SharedSetup />
  <JobDefinition ID='ImportJobDefinition_1' />
  <EntitySetup ID='Generic' />
</SharedSetup>
<TestGroup />
</TestManifest>
```

ELEMENT	ELEMENT CARDINALITY	ATTRIBUTES	ATTRIBUTE DESCRIPTION
<code>&lt;TestManifest&gt;</code>	1..1	name	The <i>name</i> helps to identify the purpose of the manifest.

### Shared setup

The **Shared setup** section defines general task parameters and behaviors for all tasks in the manifest.

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTES	ATTRIBUTE DESCRIPTION
<code>&lt;TestManifest&gt;</code>	<code>&lt;SharedSetup&gt;</code>	1..1	-	This element takes no attributes.

### Data files

`<DataFile>` elements define the data packages and data files that the tasks in the manifest will use. The data files must be either in the LCS asset library of your LCS project or in the Shared asset library.

```
<DataFile ID='SharedSetup' name='Demo data-7.3-100-System and Shared' assetType='Data package'
lcsProjectId='' />
<DataFile ID='FinancialsHQUS' name='Demo data-7.3-200-Financials-HQUS' assetType='Data package'
lcsProjectId='' />
<DataFile ID='FinancialsPICH' name='Demo data-7.3-200-Financials-PICH' assetType='Data package'
lcsProjectId='' />
<DataFile ID='FinancialsPIFB' name='Demo data-7.3-200-Financials-PIFB' assetType='Data package'
lcsProjectId='' />
```

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTES	ATTRIBUTE DESCRIPTION
<SharedSetup>	<DataFile>	1..n	-	-
	<DataFile>	-	ID	
	<DataFile>	-	name	Name of the asset that represents the data file.
	<DataFile>	-	assetType	The asset type in LCS asset library that stores the data file. This is the asset type name as shown in LCS asset library.
	<DataFile>	-	lcsProjectId	The LCS project that has the data file in its asset library. If the project ID is specified as " " then, it indicates the Shared asset library.

### Data project definition

The <JobDefinition> element defines the data project definition. There can be more than one job definition in a manifest.

```

<JobDefinition ID='ImportJobDefinition_1'>
  <Operation>Import</Operation>
  <ConfigurationOnly>No</ConfigurationOnly>
  <Truncate></Truncate>
  <Mode>Import async</Mode>
  <BatchFrequencyInMinutes>1</BatchFrequencyInMinutes>
  <NumberOfTimesToRunBatch >2</NumberOfTimesToRunBatch>
  <UploadFrequencyInSeconds>1</UploadFrequencyInSeconds>
  <TotalNumberOfTimesToUploadFile>1</TotalNumberOfTimesToUploadFile>
  <SupportedDataSourceType>Package</SupportedDataSourceType>
  <ProcessMessagesInOrder>No</ProcessMessagesInOrder>
  <PreventUploadWhenZeroRecords>No</PreventUploadWhenZeroRecords>
  <UseCompanyFromMessage>Yes</UseCompanyFromMessage>
  <LegalEntity>DAT</LegalEntity>
  <PackageAPIExecute>true</PackageAPIExecute>
  <PackageAPIOverwrite>false</PackageAPIOverwrite>
  <PackageAPIReexecute>false</PackageAPIReexecute>
  <DefinitionGroupID>TestExport</DefinitionGroupID>
  <PackageName>TestExportPackage</PackageName>
</JobDefinition>

```

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
<SharedSetup>	<JobDefinition>	1..n	ID	The job definition ID is used in the tasks to reference the definition to be used for the data project.

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
<JobDefinition>	<Operation>	1..1	-	The operation to be performed is specified by the following values: - Import - Export
	<Truncate>	1..1	-	This is a Boolean field with possible values of Yes or No. This is applicable only when operation is set to <i>Import</i> .
	<Mode>	1..1	-	The mode specifies the method using which the operation must be performed. The possible values are: - Import async - Export async - Recurring batch: This uses the enqueue API. Dequeue API is not supported yet. Package API supports both export and import.
	<ConfigurationOnly>	0..1	-	This is a Boolean field with possible values of Yes or No. This must be set to Yes if the task is only to configure the data project but not to perform the specified operation.
	<BatchFrequencyInMinutes>	1..1	-	This specifies the frequency in which the batch must be scheduled. This is applicable only when mode is set to <i>recurring batch</i> .
	<NumberOfTimesToRunBatch>	1..1	-	This is used to set a limit to how many times the scheduled batch should run. This is applicable only when mode is set to <i>recurring batch</i> .

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
	<UploadFrequencyInSeconds>	1..1	-	This is used to control the rate at which a file is uploaded to the recurring batch job for import. This must be used only for automated testing of recurring integrations in non-production environments. This is applicable only when mode is set to <i>recurring batch</i> and operation is set to <i>Import</i> .
	<TotalNumberOfTimesToUpload>	1..1		This controls the total number of times the file should be uploaded to the recurring batch. This must be used only for automated testing of recurring integrations in non-production environments. This is applicable only when mode is set to <i>recurring batch</i> and operation is set to <i>Import</i> .
	<SupportedDataSourceType>	1..1		This must be used to specify if a file is being sent to the recurring batch or a package. This is only applicable when mode is set to 'recurring batch'.
	<ProcessMessagesInOrder>	1..1		This is a Boolean field with possible values of Yes or No. This is applicable only when mode is set to <i>recurring batch</i> and operation is <i>Import</i> .
	<PreventUploadWhenZeroRecords>	1..1		This is a Boolean field with possible values of Yes or No. This is applicable only when mode is set to <i>recurring batch</i> and operation is <i>Export</i> .

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
	<UseCompanyFrom Message>	1..1		This is a Boolean field which can be set to Yes or No. This is applicable only when mode is set to <i>recurring batch</i> and operation is <i>Import</i> .
	<LegalEntity>	1..1		This is used to specify the legal entity in which the import/export job must be executed.
	<PackageAPIExecute >	1..1		Refer to package API documentation to understand this parameter. This is a Boolean field which takes 'true' or 'false'.
	<PackageAPIOverwrite>	1..1		Refer to package API documentation to understand this parameter. This is a Boolean field which takes 'true' or 'false'.
	<PackageAPIReexecute>	1..1		Refer to package API documentation to understand this parameter. This is a Boolean field which takes 'true' or 'false'.
	<DefinitionGroupID>	1..1		Refer to package API documentation to understand this parameter. This is a string field.
	<PackageName>	1..1		Refer to package API documentation to understand this parameter. This is a string field.

### Entity setup

The **Entity setup** section defines the characteristics of an entity that a task in the manifest will use. There can be more than one definition, one for each entity that is used by the tasks in the manifest.

```

<EntitySetup ID='Generic'>
  <Entity name='*'>
    <SourceDataFormatName>Package</SourceDataFormatName>
    <ChangeTracking></ChangeTracking>
    <PublishToBYOD></PublishToBYOD>
    <DefaultRefreshType>Full push only</DefaultRefreshType>
    <ExcelWorkSheetName></ExcelWorkSheetName>
    <SelectFields>All fields</SelectFields>
    <SetBasedProcessing></SetBasedProcessing>
    <FailBatchOnErrorForExecutionUnit>No</FailBatchOnErrorForExecutionUnit>
    <FailBatchOnErrorForLevel>No</FailBatchOnErrorForLevel>
    <DisableEntity>No</DisableEntity>
    <SkipStaging>Yes</SkipStaging>
    <ParallelProcessing>
      <Threshold></Threshold>
      <TaskCount></TaskCount>
    </ParallelProcessing>
    <MappingDetail StagingFieldName='RoundingRulePrices' AutoGenerate='Yes' AutoDefault='No'
    DefaultValue='' IgnoreBlankValues='No' TextQualifier='No' UseEnumLabel='No'/>
  </Entity>
</EntitySetup>

```

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
<SharedSetup>	<EntitySetup>	1..n	ID	An identification that will be used by tasks to reference an entity definition to be used.
<EntitySetup>	<Entity>	1..1	name	The entity element is identified by the entity's name. However, to facilitate easy manifest definition, this element also supports * as a wild card which will mean all entities being used in a task. This comes in very handy when using data packages with hundreds of entities in a task.
<Entity>	<SourceDataFormat Name>	1..1	-	This is the file format to be used for the entity.
	<ChangeTracking>	1..1	-	This is a Boolean field with possible values of Yes or No. It enables or disables change tracking on the entire entity.
	<PublishToBYOD>	1..1	-	This is a Boolean field with possible values of Yes or No.

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
	<DefaultRefreshType>	1..1	-	This sets the default refresh rate on the entity. The possible values are <i>Incremental push only</i> or <i>Full push</i> .
	<ExcelWorkSheetName>	1..1	-	This is used to specify the worksheet to be used for the entity.
	<SelectFields>	1..1	-	This can be used to specify the fields to be included in the template for an export operation.
	<SetBasedProcessing>	1..1	-	This is a Boolean field with possible values of Yes or No. It is used to enable or disable set based processing on an entity.
	<FailBatchOnErrorForExecutionUnit>	1..1	-	This is a Boolean field with possible values of Yes or No. It is used to enable or disable failure at execution unit level on an entity.
	<FailBatchOnErrorForLevel>	1..1	-	This is a Boolean field with possible values of Yes or No. It is used to enable or disable failure at execution level on an entity.
	<DisableEntity>	1..1	-	This is a Boolean field with possible values of Yes or No. It is used to enable or disable an entity in a data project.
	<SkipStaging>	1..1	-	This is a Boolean field with possible values of Yes or No. It is used to skip staging table for an entity during exports.



PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
	<ParallelProcessing>	1..1	-	This is used to define the parallel processing set up for an entity. The task will delete these settings if already exists at the beginning of the task and it will delete the created settings at the end of its execution.
<ParallelProcessing>	<Threshold>	1..1	-	This specifies the threshold for the parallel processing rule.
	<TaskCount>	1..1	-	This is used to specify the number of parallel tasks to be used for parallel processing.
<Entity>	<MappingDetail>	0..n	-	Allows to configure the <i>auto generate</i> , <i>auto default</i> and other settings on the mapping for an entity.
	<MappingDetail>	-	StagingFieldName	This attribute is used to identify the entity column for which the settings are to be specified.
	<MappingDetail>	-	AutoGenerate	This is a Boolean field with possible values of Yes or No for enabling/disabling auto generate option.
	<MappingDetail>	-	AutoDefault	This is a Boolean field with possible values of Yes or No for enabling/disabling auto default option.
	<MappingDetail>	-	DefaultValue	This is the default value to be used if auto defaulting is enabled.

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTE	DESCRIPTION
	<MappingDetail>	-	IgnoreBlankValues	This is a Boolean field with possible values of Yes or No for enabling/disabling this option.
	<MappingDetail>	-	TextQualifier	This is a Boolean field with possible values of Yes or No for enabling/disabling this option.
	<MappingDetail>	-	UseEnumLabel	This is a Boolean field with possible values of Yes or No for enabling/disabling this option.

## Test groups

Test groups can be used to organize related tasks in a manifest. There can be more than one test group in a manifest.

```

<TestGroup name='Set up Financials'>
  <TestCase Title='Import shared set up data package' ID='3933885' RepeatCount='1' TraceParser='off'
  Timeout='20'>
    <DataFile RefID='SharedSetup' />
    <JobDefinition RefID='ImportJobDefinition_1' />
    <EntitySetup RefID='Generic' />
  </TestCase>

  <TestCase Title='Import financials for HQUS' ID='3933886' RepeatCount='1' TraceParser='off'
  Timeout='20'>
    <DataFile RefID='FinancialsHQUS' />
    <JobDefinition RefID='ImportJobDefinition_1'>
      <LegalEntity>HQUS</LegalEntity>
    </JobDefinition>
    <EntitySetup RefID='Generic' />
  </TestCase>

  <TestCase Title='Import financials for PICH' ID='3933887' RepeatCount='1' TraceParser='off'
  Timeout='20'>
    <DataFile RefID='FinancialsPICH' />
    <JobDefinition RefID='ImportJobDefinition_1'>
      <LegalEntity>PICH</LegalEntity>
    </JobDefinition>
    <EntitySetup RefID='Generic' />
  </TestCase>

  <TestCase Title='Import financials for PIFB' ID='3933888' RepeatCount='1' TraceParser='off'
  Timeout='20'>
    <DataFile RefID='FinancialsPIFB' />
    <JobDefinition RefID='ImportJobDefinition_1'>
      <LegalEntity>PIFB</LegalEntity>
    </JobDefinition>
    <EntitySetup RefID='Generic' />
  </TestCase>
</TestGroup>

```

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTES	DESCRIPTION
<TestManifest>	<TestGroup>	1..n	-	-
	<TestGroup>	1..1	Name	This is the name for the group to identify its functional reason.
<TestGroup>	<TestCase>	1..n	-	The task is defined in this element. The task can refer to the shared set up to inherit task parameters and task behavior. The task can also override parameters and behavior at its level thus making the management of the manifest simple.
	<TestCase>	-	Title	This is the title for the task.
	<TestCase>	-	ID	This is the ID for the task. This can be alphanumeric with a max character limit of 10.
	<TestCase>	-	RepeatCount	This is a placeholder for a future functionality. However, this must be specified with a value of <i>1</i> .
	<TestCase>	-	TraceParser	This is a placeholder for a future functionality. However, this must be specified with a value <i>off</i> .
	<TestCase>	-	Timeout	This is the maximum duration a task will be monitored by the task automation manager. If the task is still active beyond the timeout specified, the manager will proceed to the next task in the manifest.

PARENT ELEMENT	ELEMENT	ELEMENT CARDINALITY	ATTRIBUTES	DESCRIPTION
<TestCase>	<DataFile>	1..n	-	This element is used to define the file or data package to be used by the task. This can reference to an already declared file or a data package in the shared section of the manifest. A task can have more than one data file specified for recurring batch import scenarios only. For other scenarios, even if more than one files are specified, the first file is what will be used by the task.
	<JobDefinition>	1..1	-	This element is used to define the data project to be used by the task. This can reference to an already declared job definition in the shared section of the manifest. The task can override elements of job definition to a new value than what is defined in the shared set up.
	<EntitySetup>	1..1	-	This element is used to define the entity set up for entities used by the task. This can reference to an already declared entity set up in the shared section of the manifest. The task can override elements of entity setup to a new value than what is defined in the shared set up.

## Best practices for manifest design

You can define a manifest in many ways. Here are a few pointers that you should consider when you design a manifest.

### Granularity

We recommend that you determine the granularity of your manifest as a functional decision. Your team must

decide whether it wants to manage many manifests or manage changes in a single manifest.

- Start with as many manifests as your team thinks you logically need. Later, when the team actually starts to run the manifests, it might find that it uses fewer manifests than it expected, and it might want to merge them. In this case, you can merge manifests.
- Consider separation of duties. For example, you might have one manifest for the setup of demo data and another manifest for the setup of the golden configuration for your environment. In this way, you can make sure that team members use only the manifests that they are supposed to use.
- Consider user access to LCS. For example, larger and globally distributed implementation teams might have multiple instances of the application or multiple LCS projects.

### Inheritance

The manifest schema supports inheritance of common elements that will apply to all tasks in the manifest. A task can override a common element to define a unique behavior. The purpose of the **Shared setup** section is to minimize repetition of configuration elements, so that elements are reused as much as possible. The goal is to keep the manifest concise and clean, to improve maintenance and readability.

### Source control

Manifests that must be used by all the members of an implementation team should be stored in source control in the Application Object Tree (AOT). This approach not only provides the benefits of source control, but also enables a process to distribute or make manifests available to all users in a consistent manner. This approach also enables configuration management for data projects that are related to data management, if manifests are used for configuration.

### Number of job definitions and entity definitions

For most of the use cases, one job definition in a manifest should be enough, because inheritance can be used to change the behavior at the task level. This principle also applies to entity definitions.

## Validations

Data task automation manager performs validations, based on the setup of a task. If a task fails, you can quickly determine the reason for the failure by viewing the validations after the task is completed. The level of information that Data task automation manager provides is optimized to facilitate initial discovery. For detailed investigation, you must look at the corresponding data project and its execution details.

The following data validations are currently supported:

- **Job status** – This validation checks whether the job was successful.
- **Batch status** – This validation checks whether the batch was successful.
- **Message status** – If the test is about integrations, the message status is validated.
- **Truncation** – If truncation is enabled, this validation checks whether truncation occurred.
- **Skip staging** – If **Skip staging** is enabled on a test, this validation checks whether staging was skipped.

## Example 1: Configuration management for data projects

The `<ConfigurationOnly>` element can be used to create configuration tasks for data projects. When `ConfigurationOnly` is set to `Yes`, the data projects are only created but not executed. This allows for managing data projects across environments in an automated manner.

```
<?xml version='1.0' encoding='utf-8'?>
<TestManifest name='Data management demo data set up'>
  <SharedSetup>
    <DataFile ID='SharedSetup' name='Demo data-7.3-100-System and Shared' assetType='Data package'
lcsProjectId='' />
    <DataFile ID='FinancialsHQUS' name='Demo data-7.3-200-Financials-HQUS' assetType='Data package'
```

```

lcsProjectId='' />
  <DataFile ID='FinancialsPICH' name='Demo data-7.3-200-Financials-PICH' assetType='Data package'
lcsProjectId='' />
  <DataFile ID='FinancialsPIFB' name='Demo data-7.3-200-Financials-PIFB' assetType='Data package'
lcsProjectId='' />

  <JobDefinition ID='ImportJobDefinition_1'>
    <ConfigurationOnly>Yes</ConfigurationOnly>
    <Operation>Import</Operation>
    <Truncate>No</Truncate>
    <Mode>Import async</Mode>
    <BatchFrequencyInMinutes>1</BatchFrequencyInMinutes>
    <NumberOfTimesToRunBatch >2</NumberOfTimesToRunBatch>
    <UploadFrequencyInSeconds>1</UploadFrequencyInSeconds>
    <TotalNumberOfTimesToUploadFile>1</TotalNumberOfTimesToUploadFile>
    <SupportedDataSourceType>Package</SupportedDataSourceType>
    <ProcessMessagesInOrder>No</ProcessMessagesInOrder>
    <PreventUploadWhenZeroRecords>No</PreventUploadWhenZeroRecords>
    <UseCompanyFromMessage>Yes</UseCompanyFromMessage>
    <LegalEntity>DAT</LegalEntity>
  </JobDefinition>

  <EntitySetup ID='Generic'>
    <Entity name='*'>
      <SourceDataFormatName>Package</SourceDataFormatName>
      <ChangeTracking>No</ChangeTracking>
      <PublishToBYOD>No</PublishToBYOD>
      <DefaultRefreshType>Full push only</DefaultRefreshType>
      <ExcelWorkSheetName></ExcelWorkSheetName>
      <SelectFields>All fields</SelectFields>
      <SetBasedProcessing>No</SetBasedProcessing>
      <FailBatchOnErrorForExecutionUnit>No</FailBatchOnErrorForExecutionUnit>
      <FailBatchOnErrorForLevel>No</FailBatchOnErrorForLevel>
      <FailBatchOnErrorForSequence>No</FailBatchOnErrorForSequence>
      <ParallelProcessing>
        <Threshold></Threshold>
        <TaskCount></TaskCount>
      </ParallelProcessing>
    </Entity>
  </EntitySetup>
</SharedSetup>

  <TestGroup name='Set up import jobs for Financials'>
    <TestCase Title='Set up import job for shared set up data package' ID='3933885' RepeatCount='1'
TraceParser='off' TimeOut='20'>
      <DataFile RefID='SharedSetup' />
      <JobDefinition RefID='ImportJobDefinition_1' />
      <EntitySetup RefID='Generic' />
    </TestCase>

    <TestCase Title='Set up import job for financials HQUS' ID='3933886' RepeatCount='1'
TraceParser='off' TimeOut='20'>
      <DataFile RefID='FinancialsHQUS' />
      <JobDefinition RefID='ImportJobDefinition_1'>
        <LegalEntity>HQUS</LegalEntity>
      </JobDefinition>
      <EntitySetup RefID='Generic' />
    </TestCase>

    <TestCase Title='Set up import job for financials PICH' ID='3933887' RepeatCount='1'
TraceParser='off' TimeOut='20'>
      <DataFile RefID='FinancialsPICH' />
      <JobDefinition RefID='ImportJobDefinition_1'>
        <LegalEntity>PICH</LegalEntity>
      </JobDefinition>
      <EntitySetup RefID='Generic' />
    </TestCase>

    <TestCase Title='Set up import job for financials PIFB' ID='3933888' RepeatCount='1'

```

```

TraceParser='off' Timeout='20'>
  <DataFile RefID='FinancialsPIFB' />
  <JobDefinition RefID='ImportJobDefinition_1'>
    <LegalEntity>PIFB</LegalEntity>
  </JobDefinition>
  <EntitySetup RefID='Generic' />
</TestCase>
</TestGroup>
</TestManifest>

```

## Example 2: Automated setup of demo data

The following manifest shows the setup of demo data for three legal entities when the demo data packages are stored in the Shared asset library. The difference in this example from the previous example is the actual execution of the data projects to set up the demo data. This is accomplished by not using the ConfigurationOnly option or setting it to No to use it for consistency of the manifest.

```

<?xml version='1.0' encoding='utf-8'?>
<TestManifest name='Data management demo data set up'>
  <SharedSetup>
    <DataFile ID='SharedSetup' name='Demo data-7.3-100-System and Shared' assetType='Data package'
lcsProjectId='' />
    <DataFile ID='FinancialsHQUS' name='Demo data-7.3-200-Financials-HQUS' assetType='Data package'
lcsProjectId='' />
    <DataFile ID='FinancialsPICH' name='Demo data-7.3-200-Financials-PICH' assetType='Data package'
lcsProjectId='' />
    <DataFile ID='FinancialsPIFB' name='Demo data-7.3-200-Financials-PIFB' assetType='Data package'
lcsProjectId='' />

    <JobDefinition ID='ImportJobDefinition_1'>
      <Operation>Import</Operation>
      <Truncate></Truncate>
      <Mode>Import async</Mode>
      <BatchFrequencyInMinutes>1</BatchFrequencyInMinutes>
      <NumberOfTimesToRunBatch >2</NumberOfTimesToRunBatch>
      <UploadFrequencyInSeconds>1</UploadFrequencyInSeconds>
      <TotalNumberOfTimesToUploadFile>1</TotalNumberOfTimesToUploadFile>
      <SupportedDataSourceType>Package</SupportedDataSourceType>
      <ProcessMessagesInOrder>No</ProcessMessagesInOrder>
      <PreventUploadWhenZeroRecords>No</PreventUploadWhenZeroRecords>
      <UseCompanyFromMessage>Yes</UseCompanyFromMessage>
      <LegalEntity>DAT</LegalEntity>
    </JobDefinition>

    <EntitySetup ID='Generic'>
      <Entity name='*'>
        <SourceDataFormatName>Package</SourceDataFormatName>
        <ChangeTracking></ChangeTracking>
        <PublishToBYOD></PublishToBYOD>
        <DefaultRefreshType>Full push only</DefaultRefreshType>
        <ExcelWorkSheetName></ExcelWorkSheetName>
        <SelectFields>All fields</SelectFields>
        <SetBasedProcessing></SetBasedProcessing>
        <FailBatchOnErrorForExecutionUnit>No</FailBatchOnErrorForExecutionUnit>
        <FailBatchOnErrorForLevel>No</FailBatchOnErrorForLevel>
        <FailBatchOnErrorForSequence>No</FailBatchOnErrorForSequence>
        <ParallelProcessing>
          <Threshold></Threshold>
          <TaskCount></TaskCount>
        </ParallelProcessing>
      </Entity>
    </EntitySetup>
  </SharedSetup>

  <TestGroup name='Set up Financials'>

```

```
<TestCase Title='Import shared set up data package' ID='3933885' RepeatCount='1' TraceParser='off'
Timeout='20'>
  <DataFile RefID='SharedSetup' />
  <JobDefinition RefID='ImportJobDefinition_1' />
  <EntitySetup RefID='Generic' />
</TestCase>

<TestCase Title='Import financials for HQUS' ID='3933886' RepeatCount='1' TraceParser='off'
Timeout='20'>
  <DataFile RefID='FinancialsHQUS' />
  <JobDefinition RefID='ImportJobDefinition_1'>
    <LegalEntity>HQUS</LegalEntity>
  </JobDefinition>
  <EntitySetup RefID='Generic' />
</TestCase>

<TestCase Title='Import financials for PICH' ID='3933887' RepeatCount='1' TraceParser='off'
Timeout='20'>
  <DataFile RefID='FinancialsPICH' />
  <JobDefinition RefID='ImportJobDefinition_1'>
    <LegalEntity>PICH</LegalEntity>
  </JobDefinition>
  <EntitySetup RefID='Generic' />
</TestCase>

<TestCase Title='Import financials for PIFB' ID='3933888' RepeatCount='1' TraceParser='off'
Timeout='20'>
  <DataFile RefID='FinancialsPIFB' />
  <JobDefinition RefID='ImportJobDefinition_1'>
    <LegalEntity>PIFB</LegalEntity>
  </JobDefinition>
  <EntitySetup RefID='Generic' />
</TestCase>
</TestGroup>
</TestManifest>
```

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Delivering ISV solutions using One Version

2/18/2021 • 17 minutes to read • [Edit Online](#)

Thanks to One Version, new updates are now automatically broadcast, downtime is minimal, and customers enjoy the benefits of staying current with recent features and fixes without having to go through expensive upgrades.

Feature management lets customers control when new features are applied. Therefore, as an independent software vendor (ISV) partner, you can innovate together with Microsoft to take advantage of new features without having to handle the waiting times that come with long release cycles. When all your customers run on current versions, you have fewer versions to maintain. You can focus instead on building quality into the solutions that you provide for your customers.

The process of servicing current versions is also more seamless and safer than it was in earlier versions. Previously, patching required that individual fixes be combined and merged into a customer environment.

Extensibility allows for deployment of side-by-side solutions that give customers more choices about how they configure their solutions.

In the One Version model, customer user acceptance testing (UAT) environments and production environments are updated every month. It's critical that updates not cause issues. However, Microsoft acknowledges that both technical issues and functional issues may arise when environments are updated.

- Technical issues include breaking changes in application programming interfaces (APIs) that customizations in your solutions use.
- Functional issues that customers experience can be caused by the untimely introduction of new features. Microsoft will put any new functionality that might affect existing processes under feature management. In this way, customers can control when new functionality is adopted. Therefore, they have time to validate, document, and train their users about the new features.
- Functional issues might also be unintended changes that cause functional regressions.

Prevention of technical and functional issues is difficult and requires close coordination between Microsoft and you as an ISV partner. The Microsoft goal is that you will adopt practices that resemble Microsoft practices. This topic explains how you and Microsoft can achieve this goal together. Over the next several months, Microsoft will release new practices and tools to help you. This topic will be updated as the tools and practices evolve.

This topic includes the following sections:

- [Servicing customers](#)
- [Compatibility](#)
  - [Runtime compatibility](#)
  - [Design-time compatibility](#)
- [Developing new releases](#)
  - [Designing for extensibility](#)
  - [Data upgrade](#)
  - [Feature exposure](#)
- [Branches and builds](#)
- [Testing](#)

- [Deploying updates](#)
- [ISV solutions as part of One Version automated deployment](#)
- [Should I release binaries or source code?](#)

## Servicing customers

Dynamics 365 Finance and Operations apps run on Microsoft Azure. Therefore, it's a solution that runs as a service. Microsoft services companies 24/7, either proactively from alerts that report unusual behavior, or from support tickets that are submitted by customers or their partners. Microsoft has a range of tools to help support the services that are running. These tools include usage data that is collected from the services. To help safeguard customer data, Microsoft is also careful about who can access customer systems.

When Microsoft analyzes an issue, it might determine that the issue is related to your ISV solution. Microsoft reports this type of issue to you, so that you can follow up offline.

Companies can opt out of updates for two consecutive service updates before the next service update is applied to their environments. Therefore, at any time, Microsoft can expect that companies will be running one of the last three monthly updates.

When Microsoft resolves an issue that requires a code fix, it generally includes that code fix in the next monthly update. However, very critical issues that are reported, such as production outage, might require that a fix be provided for the version that customers are currently running.

Similar policies apply to your ISV solution, and you might also have to provide a code update. For your solution to be binary-compatible with all your customers, it must be built on the oldest platform release that you want to support. All new updates that Microsoft releases are intended to be binary backward-compatible. This compatibility gives you the option of maintaining only one servicing version of your solution that is based on the oldest of the three most recent updates. Therefore, you must maintain just one released solution. You can then use that solution to update all your customers, regardless of which of the three most recent updates they are running. As your customers adopt new Microsoft updates, you can rebase your maintained solution to a newer release to remain current with the three most recent updates.

This recommendation applies to servicing and maintaining your released solution. You will use a different approach to develop new releases of your solution. For more information, see the [Developing new releases](#) section of this topic.

## Compatibility

Microsoft diligently tries to guarantee compatibility with existing customizations. To achieve this goal, Microsoft uses strict practices in its engineering processes, together with tool and automation support that helps identify API contracts that are unintentionally broken. Telemetry lets Microsoft engineers determine customizations that reference or extend a Microsoft API.

Updates to Finance and Operations apps that are applied to customer environments are intended to be functionally compatible and binary-compatible with existing customizations. This compatibility covers not only APIs, but also functionality and the user experience. Customers must explicitly opt in to all new experiences.

Any deprecation or breaking change in binary or functional compatibility will be announced 12 months in advance. Therefore, you will have enough time to align your customizations with an alternative design. You must pay attention to the monthly updates to Microsoft documentation, and you must review the APIs that are marked as obsolete (deprecated) or internal. In this way, you will be able to manage changes in a timely manner.

The following sections define and describe the aspects of backward compatibility: runtime compatibility and design-time compatibility.

## Runtime compatibility

All new updates are intended to be runtime backward-compatible. This compatibility covers both binary compatibility and functional compatibility. Runtime compatibility means that customizations that exist in production and sandbox environments will continue to work after new updates are deployed in those environments. These updates include both service updates and quality updates.

Runtime compatibility also means that changes made by Microsoft to the platform will be backward-compatible with customizations that were compiled on an earlier platform.

Binary compatibility is backwards only. You can compile a customization on an older platform, and deploy it to a customer environment that has been updated to a later version. You cannot deploy code compiled on a later version than the one running on the environment you deploy to.

## Design-time compatibility

Design-time backward compatibility (that is, compile-time compatibility) means that developers can apply updates to their development environments and successfully compile their code without having to make any changes.

You must be aware of how APIs in your solution are used in your customers' implementations, and how you can use these APIs without causing breaking changes. As part of this effort, you must pay careful attention to what is changed and rely on engineering best practices. For examples of changes that you should avoid, see [Breaking changes](#).

You should try to meet a bar that resembles the bar that Microsoft sets. In that way, both you and Microsoft can help avoid creating regressions.

All Microsoft updates are intended to be binary (runtime) compatible, and Microsoft also aims for design-time compatibility. However, there is one category of required changes that is **not** design time-compatible but remains binary-compatible. After an update is applied, new errors or warnings might occur when your code is compiled. Here are some examples of these changes:

- Microsoft makes an enumeration extensible.
- Microsoft marks an API as obsolete or internal.
- Microsoft introduces a new compiler error to help avoid unsafe coding practices.

All these changes might require work on your solution.

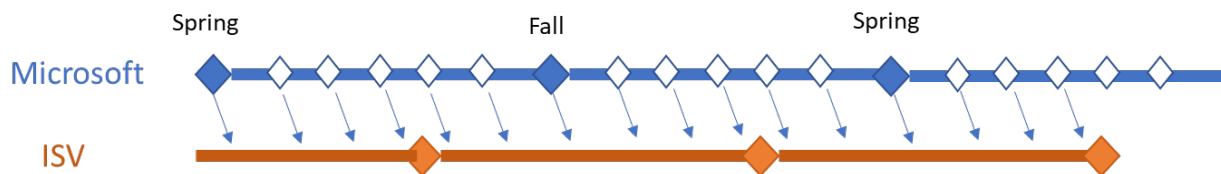
Design-time breaking changes that are binary-compatible don't require a 12-month deprecation notice.

## Developing new releases

Together, One Version and the fact that the Finance and Operations apps run as a service provides a great vehicle for collecting feedback. Feedback is useful, because it helps Microsoft decide which new features it should add to upcoming updates. Historically, the Microsoft approach has been to release major releases that include many new features. However, the new model encourages a different approach. Therefore, Microsoft has moved to a series of continuous updates that gradually build on the available capabilities of the system. In many cases, one update contains an initial small feature that Microsoft then enriches in later updates. In some cases, Microsoft must provide staging for new features, and must use feature exposure to hide the new features or control modifications to them.

We recommend that you follow a similar approach for your ISV solutions. You will benefit from quicker integration and extension of new standard features.

As the following illustration shows, the frequency of your new releases can be independent of Microsoft releases. You should consider adopting a strategy for source code branching, as described in the [Branches and builds](#) section of this topic.



What is essential is the quality of every update that is released. Although testing helps guarantee quality, quality must also be built in during the design and implementation phases. In the One Version model, there are some new dimensions to consider, as described in the following sections.

### Designing for extensibility

To design your solution for extensibility, you must consider now only how you will customize by extending the standard application, but also how you will enable customization of your ISV solutions by your customers and partners.

Make sure that customizations are additive instead of intrusive, and follow the guidance on the [Extensibility home page](#).

Don't be too creative about the way that you build your customization. Otherwise, you might extend an API that is questionable and increase the risk that later updates will break your solution. Instead, log an [extensibility request](#), and ask that Microsoft create a more explicit API that is more resilient to breakage.

Design solutions that are extensible. For inspiration, see [Write extensible code](#).

Design for backward compatibility to avoid breaking customer implementations. A good strategy is to be explicit about what you offer for hooking and wrapping extension code. The way that you decorate your methods gives you lots of control over which methods you enable extensions for. For more information, see [Attributes that make methods extensible](#).

### Data upgrade

The types of data upgrade jobs that existed in earlier versions are no longer supported. This change was made because Microsoft wants to provide minimum downtime while a production environment is updated.

Database synchronization is still run during upgrade, and it supports basic operations such as adding new tables, field, and indexes.

To prevent downtime, Microsoft is introducing new ways of driving data upgrade that are run asynchronously. For example, data upgrade will sometimes be triggered when a feature is turned on through a feature flag. This new approach for data upgrade differs significantly from earlier approaches and will become available in upcoming updates. Documentation resources will also be available.

### Feature exposure

In the One Version model, updates are managed by Microsoft and pushed to customer environments. Pushed updates should not require that customers adjust to functional changes, or that they train their users about new or changed features every month. Pushed updates also should not cause customers to delay updates to their environment.

Feature management is a new concept that puts customers in control of deciding when new or changed features are used. Customers can review, validate, and document new or changed features before they are adopted. They can also train users before new or changed processes are adopted, to reduce the impact on daily operations.

Feature management will be released in upcoming monthly updates.

You should consider using feature management with your ISV solution to let customers control when new features are adopted.

# Branches and builds

As an ISV, you should plan on a minimum of two source code branches: a servicing branch and a development branch.

## Servicing branch

The servicing branch is used to produce bug fixes for your ISV solution. As the ISV, you specify the frequency of releases from the branch and distribution of the releases. The expectation is that these releases from the servicing branch will be binary cumulative releases.

The base Microsoft version that you use to build your ISV solution should match the oldest version that customers use with the solution. In the One Version model, that version starts with version 8.1 and is a maximum of three months old.

## Development branch

The development branch is used to develop new capabilities for the ISV solution. As the ISV, you determine the frequency of releases from the development branch. You don't have to synchronize your releases with the monthly Microsoft releases. The best approach might be to decouple your release schedule from the Microsoft release schedule and deliver releases less often. A quarterly or biannual cadence is a good starting point.

The base Microsoft version in the development branch should be either the latest released version that is available or the released version that you plan to use for servicing when your new release goes out. The goals are that you innovate together with Microsoft by staying as current as is feasible, and that your development model allows for uptake of recent feature work.

# Testing

Microsoft has several checks and balances in its development process to help guarantee functional and binary compatibility. ISV solutions must be validated with each Microsoft release to help guarantee this compatibility. The expectation is that you will do this validation during the Partner Early Access Program (PEAP) phase of each release.

It's very important that you provide quick turnaround for feedback, so that you will have time to fix any issues before the monthly updates are deployed in customer environments.

Test automation is important for quick validation of new updates. Microsoft plans to release the test framework and libraries to support you as you build your test automation.

Microsoft has an extensive suite of tests that support our validation. The expectation is that you, as an ISV, will create your own suites of automated tests.

In addition to the SysTest automation framework that is aimed at developers, the [Regression Suite Automation Tool \(RSAT\)](#) enables automation of business processes without requiring that code be written. Functional users can use the RSAT to record their critical tests and automate part of their UAT. You can also use the RSAT as you start to build your test automation.

Recently, Microsoft released the [Acceptance test library resources](#) and accompanying libraries. This framework is aimed at developers and lets them build tests that are more comprehensive than unit tests. The libraries that accompany the framework help make it a seamless way to build suites of tests.

## Currently released products – Testing binary and functional compatibility

The currently released product that is maintained in the servicing branch should first be tested for binary and functional compatibility. Your suite of automated developer tests, automated functional tests, and manual tests for your ISV solution should be run on an environment that has the new version from Microsoft and your existing ISV solution. Because this test run is testing for binary and functional compatibility, the ISV solution should **not** be recompiled.

If the testing is successful, this step will validate that a customer installation of the current version of your ISV solution won't have to be updated when Microsoft broadcasts the new release to the customer.

If the testing isn't successful, you, as the ISV, must immediately notify Microsoft through the [PEAP communication process](#). This process uses Yammer and an issue notification process. The issue will require either a fix from Microsoft or a fix in your ISV solution. A fix in your solution might, in turn, require that customers be updated from the servicing branch. In both cases, Microsoft must know about the issue, so that it can become more proactive in its processes for future releases.

### **Currently released products – Testing design-time compatibility**

Next, the currently released product that is maintained in the servicing branch should be tested for design-time compatibility. To do this testing, you should compile the solution against a deployment of the new Microsoft release. Although the goal of Microsoft is to minimize design-time compatibility issues, they might occur in some situations. One example is when Microsoft makes an enumeration extensible, and the solution uses it in a manner that assumes an underlying integer representation (for example, it uses the enumeration value in a logical comparison or mathematical function). Although the code will continue to work in a customer deployment because the underlying values are maintained, a compiler error is generated and addressed in future releases. Another example of a design-time compatibility issue is when Microsoft introduces a new compiler error to protect against unsafe coding patterns. To learn about more categories of design-time compatibility issues, see [Breaking changes](#).

You should run your suite of automated developer tests, automated functional tests, and manual tests on an environment that has the new version from Microsoft and your compiled ISV solution.

If the testing is successful, this step will validate that your ISV solution won't have to be updated even if source code is supplied to the customer and the customer recompiles the ISV solution.

If the testing isn't successful, and the issue isn't one of the categories that are described in [Breaking changes](#), you, as the ISV, must immediately notify Microsoft through the [PEAP communication process](#). This process uses Yammer and an issue notification process. The issue will require either a fix from Microsoft or a fix in your ISV solution. A fix in your solution might, in turn, require that customers be updated from the servicing branch. In both cases, Microsoft must know about the issue, so that it can become more proactive in its processes for future releases.

### **Currently released products – Updating the base build**

As Microsoft updates your customers to new releases, you should periodically update the base build, so that it matches the oldest version that is used by customers who run your ISV solution.

### **Solutions that are in development**

You validate your new solution development on either the latest released version or the released version that you plan for to use for servicing when your new release goes out. However, in both cases, consider doing validation on the most current version. This validation will help with early discovery of issues or uptake work that you must do.

If an unexpected break occurs, then you, as the ISV, must immediately notify Microsoft through the [PEAP communication process](#). This process uses Yammer and an issue notification process.

## Deploying updates

For Microsoft standard platform and application updates, One Version servicing includes a process for automated updates of customer environments. However, this automation isn't currently available for ISV solutions. For more information, see [ISV's as part of One Version service updates](#).

ISV solutions are manually updated, and you control your release cadence. The binary backward compatibility allows for safe updates of the standard platform and application.

The update process includes database synchronization (for example, the addition of new fields and indexes).

## ISV solutions as part of One Version automated deployment

Although Microsoft doesn't currently plan to release ISV solutions as part of the One Version automated deployment process, this option might become available at some point. However, Microsoft must first align engineering processes to make this option feasible.

Here are some areas where alignment will be required:

- **Feature management** – The user must be able to control when a new feature is turned on.
- **Backward compatibility and compliance** – Compliance with API customization usage is required.
- **Feature deprecation** – Advanced notice about deprecation of features or APIs must be provided.
- **Test automation suite**
- **Testing during the preview phase**
- **ISV solution sign-off and upload**
- **Automated deployment scripts**
- **Zero downtime** – Deployment of updates must be instantaneous.
- **Data migration without downtime**
- **Support for on-call duty, for servicing of critical production issues**

## Should I release binaries or source code?

Binary compatibility is supported, provided that you don't recompile. We recommend that your ISV solution not be compiled in customer environments. Instead, you should deploy precompiled binaries that you've prepared and validated. Your solution binaries can then be created from your servicing branch, based on an earlier version, when this approach is practical.

If an implementation partner or customer compiles your solution in an updated environment, new warnings and errors might occur, as was mentioned in the [Design-time compatibility](#) section of this topic. Therefore, we recommend that implementation partners not compile your solution. However, this recommendation doesn't mean that you shouldn't share your source code to help support debugging, for example. You should just consider taking steps to avoid compilation of your code, so that implementation partners aren't exposed to design-time issues.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create and automate user acceptance tests

2/18/2021 • 8 minutes to read • [Edit Online](#)

You can use Task recorder and Business process modeler (BPM) to create user acceptance test libraries. Task recorder is a powerful tool to record test cases and organize them by business process using BPM. As a Microsoft partner you can use BPM to distribute test libraries to your customers via LCS and LCS solutions. If you are a customer, use BPM to author and distribute test libraries across different projects and team.

Because BPM can be synchronized with Azure DevOps (formerly known as Visual Studio Team Services), you can automatically create test cases (including test steps) in your Azure DevOps project. Azure DevOps can then serve as your test configuration and test management tool where you can create targeted test plans and test suites, manage the execution of tests and investigate results. For more information about testing with Azure DevOps, see [What are test plans, test suites, and test cases?](#)

This topic walks through the process of creating and executing acceptance test suites to be used for manual or automated testing.

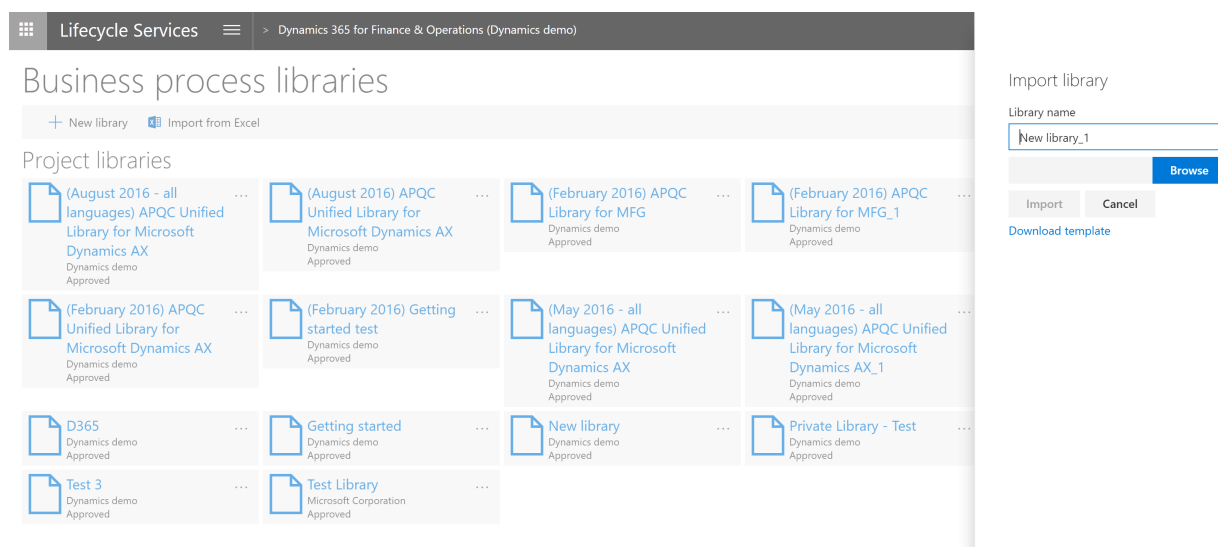
## Create a Scenario Acceptance Testing BPM library

BPM is a great LCS tool to describe a hierarchy of business processes and user tasks. LCS also allows Microsoft partners and customers to author and distribute BPM libraries across LCS projects via the Asset library. This section describes how to take advantage of BPM to define your acceptance test library.

### Create a BPM library

There are several ways to create a Business process modeler (BPM) library. For more information about how to create libraries in BPM, see [Create, edit, and browse Business process modeler \(BPM\) libraries](#).

For illustration purposes, this topic uses a library that contains common business processes, such as create an expense report and approve order requests. The library was created by using the Excel import functionality.



The screenshot displays the Lifecycle Services interface for Dynamics 365 for Finance & Operations. The main area shows a grid of 'Business process libraries' with various titles like '(August 2016 - all languages) APQC Unified Library for Microsoft Dynamics AX' and 'Getting started test'. On the right, an 'Import library' dialog is open, showing a text input field with 'New library\_1' and buttons for 'Browse', 'Import', 'Cancel', and 'Download template'.

### Record test cases and save to BPM

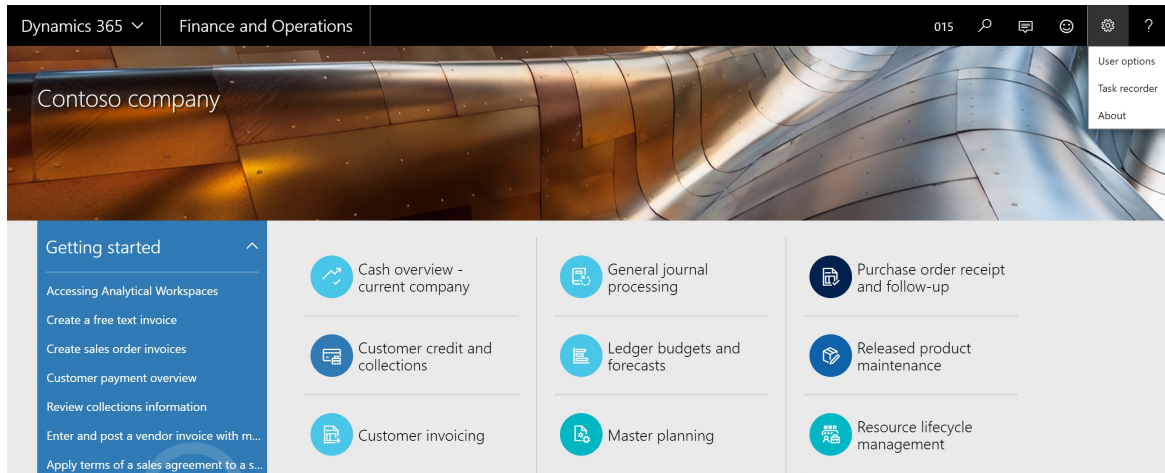
After you have created a BPM library, you'll need to use Task recorder to create your test cases and then upload the cases to BPM. There are several ways to do this.

If you're using a BPM library that already has all of the necessary task recordings (test cases) attached, you can skip this step. Otherwise, follow the instructions below to create new task recordings.

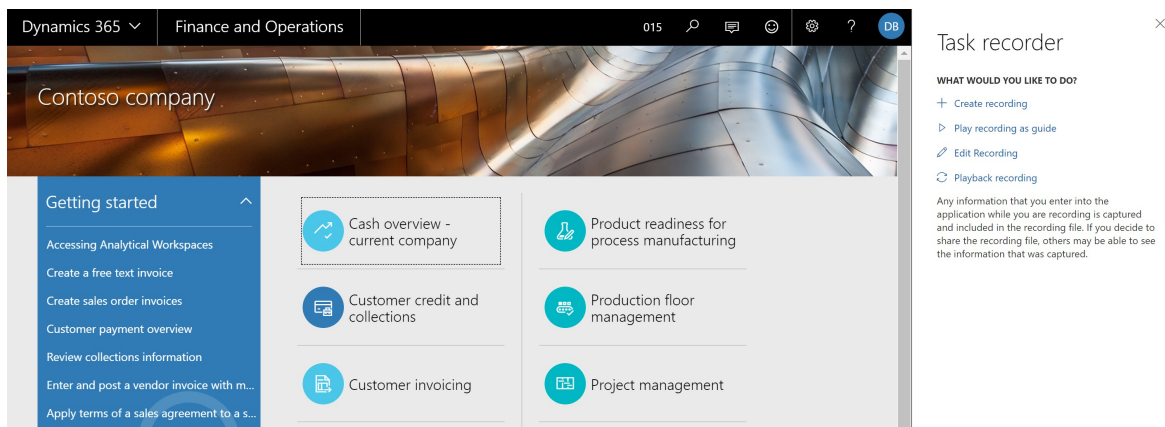


### Create and save a new task recording

1. Open the client and sign in.
2. Select the company that you want to use while recording.
3. Go to **Settings > Task recorder**.



4. Click **Create a new recording**.
5. Enter a name for the recording, and then click **Start**. Recording begins the moment that you click **Start**.
6. When the recording is complete, in the Task recorder pane, click **Stop**.
7. To save the task recording to an attached BPM, click **Save to Lifecycle Services**.



8. Select the library that you want to save the recording to, and then click **Save**. Otherwise, select **Save to Disk** and follow the steps in the next section, "Upload an AXTR file to BPM."

#### NOTE

To enable the effective execution of your tests using automation tools, make sure all of your task recordings start on the main dashboard of your application. For end-to-end processes that are performed by more than one user, we recommend that you divide your task recordings into user-specific tasks. This simplifies the maintenance of test cases and allows you to execute test cases in the context of security roles, which is a best practice.

### Upload an AXTR file to BPM

If you have saved your recordings (AXTR files) to disk, follow these steps to upload them to BPM.

1. In Lifecycle Services (LCS), in your project, on the **Business process libraries** page, select the library to upload the task recording to.
2. Click **Author and edit** and in the lines, locate and select the process to upload the task recording to.

3. In the right pane, click **Upload**.

(August 2016) APQC Unified Library for Microsoft Dynamics AX

The screenshot shows the APQC Unified Library interface. On the left, there is a search bar and a list of processes. The 'Develop Vision and Strategy' process is selected. On the right, there is a detailed view of this process, including its name, description, and metadata.

Process	Diagrams	Reviewed
Develop Vision and Strategy	18	0/4
Develop and Manage Products and Services	31	0/2
Develop and Manage Customer Experience	11	0/5
Market and Sell Products and Services	18	0/5
Market Products and Services	5	0/2
Deliver Products and Services	128	0/6
Merchandise Products and Services	9	0/2
Manage Customer Service		0/3
Deliver Products	6	0/4
Develop and Manage Human Capital	36	0/6
Manage Information Technology	13	0/7
Manage Financial Resources	267	0/10
Acquire, Construct, and Manage Assets	1	0/4
Manage Enterprise Risk, Compliance, and Resiliency		0/3
Manage External Relationships		0/5
Develop and Manage Business Capabilities	4	0/6

**Overview** Requirements

Name: Develop Vision and Strategy

Description: Develop vision and strategy establishes a direction and vision for an organ defining the business concept and long-term vision, as well as developing strategy and managing strategic initiatives. Processes in this category focus vision, a mission, and strategic objectives, which culminate in creating mea that the organization is moving in the desired direction.

Modified by:

Modified at: 08/30/2016, 2:09 PM PDT

APQC ID: 10002

APQC hierarchy ID: 1

Keywords:

4. Click **Browse** to find and select the file to upload, and then click **Upload**.

The screenshot shows the same APQC Unified Library interface as above, but with an 'Upload AXTR' dialog box open on the right side. The dialog box has a 'Browse' button and 'Upload' and 'Cancel' buttons.

**Save an existing task recording to BPM**

1. To attach an existing task recording, sign in to the client.
2. Go to **Settings > Task recorder**.
3. Select **Edit Task Recording** and attach the file by either saving directly to LCS or downloading the AXTR and then uploading to BPM.

**Guidelines for recording test cases**

Follow these guidelines when authoring and recording your test cases, especially if you are planning to automate test execution. The process and tools described in this article apply to business process acceptance tests. They are not meant to replace component and unit testing that is typically owned by developers.

- Author a limited number of test cases that, when combined, cover complete end-to-end processes.
- Focus on business processes that have been customized.
- An individual test case (recording) should cover one or two business tasks only, typically executed by one person. This simplifies task recording maintenance. Do not combine a complete end-to-end business process such as "Procure to Pay" or "Order to Cash" into one large task recording. For example, instead of having RFQ > Purchase Order > Product Receipt > Vendor Invoice > Vendor Payment as one test case, divide the process into three or four test cases. You will have the opportunity to combine these tests into an ordered

test suite later.

- A test case should have at least one validation. Try to validate critical fields that cover the impact of other fields. For example: Validation of totals on sales or purchase orders cover the unit price/quantity/discount/tax ...etc.
- Avoid printing a report in a test case. If a test case needs to print a report, it should be selected on screen.
- 80+% of test cases should be of transactions or source documents. Master data should be limited to up to 20% of test cases only.

## Synchronize and configure your test plan in Azure DevOps

An acceptance test library is your starting point. It typically contains all test cases (task recordings) of a particular application organized by business process. During a particular test pass, you usually do not need to execute all test cases. What test cases you select depends on the phase of your implementation or the nature of the update you are planning to apply to your production environment. Azure DevOps enables you to organize your test cases in test plans and test suites. A test plan contains one or more test suites (A subset of your test library); test cases can belong to more than one test suite.

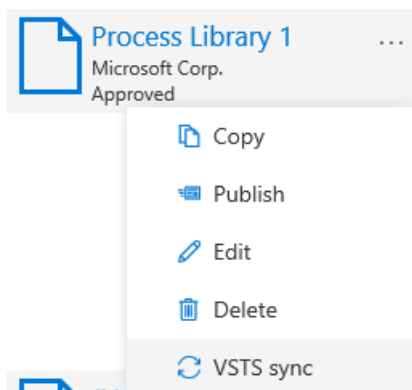
Once you have selected your acceptance testing BPM library, synchronize it with Azure DevOps and create your test plan and test suites.

### Sync with Azure DevOps

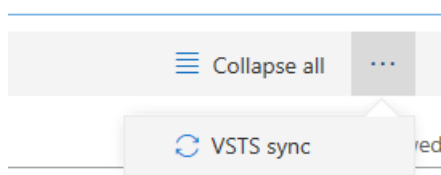
Synchronize your BPM library with your Azure DevOps project. For more information, see [Synchronize BPM libraries with Azure DevOps](#).

After configuration is complete, synchronize the BPM library with a Azure DevOps project.

1. On the **Business process libraries** page, on the tile for the library that you want to synchronize, select the ellipsis button (...), and then select **Azure DevOps sync**.



You can also start Azure DevOps synchronization from the toolbar in a BPM library. Select the ellipsis button (...), and then select **Azure DevOps sync**.



2. After Azure DevOps synchronization is complete, select the ellipsis button (...), and then select **Sync test cases**.

# DemoLibrary

Keyword or AOT object name (\$FormName)		
+ Add process   Delete process   Import   Move process   Collapse all   ...		
Process		
Create Trip Report		
^ Create Expense Report		0/8
Click New expense report		-
Select Purpose Value		-
Enter Map to travel requisition		-
Enter Transaction Date		-
Select Expense category		-
Select Merchant		-
Enter Transaction amount value		-
Click Save		-
^ Approve Order Request		0/9

✓ VSTS sync  
✓ Sync test cases

3. When this step is complete, your task recordings will become test cases in Azure DevOps and a link will appear under the **Requirements** tab.

DemoLibrary		
Keyword or AOT object name (\$FormName)		
+ Add process   Delete process   Import   Move process   Collapse all   ...		
Process	Diagrams	Reviewed
Create Trip Report		-
^ Create Expense Report		0/8
Click New expense report		-
Select Purpose Value		-
Enter Map to travel requisition		-
Enter Transaction Date		-
Select Expense category		-
Select Merchant		-
Enter Transaction amount value		-
Click Save		-
^ Approve Order Request		0/9
Open order		-
Review Order Type		-
Enter Value for Maximum Cost		-

Overview **Requirements**

+ Add requirement

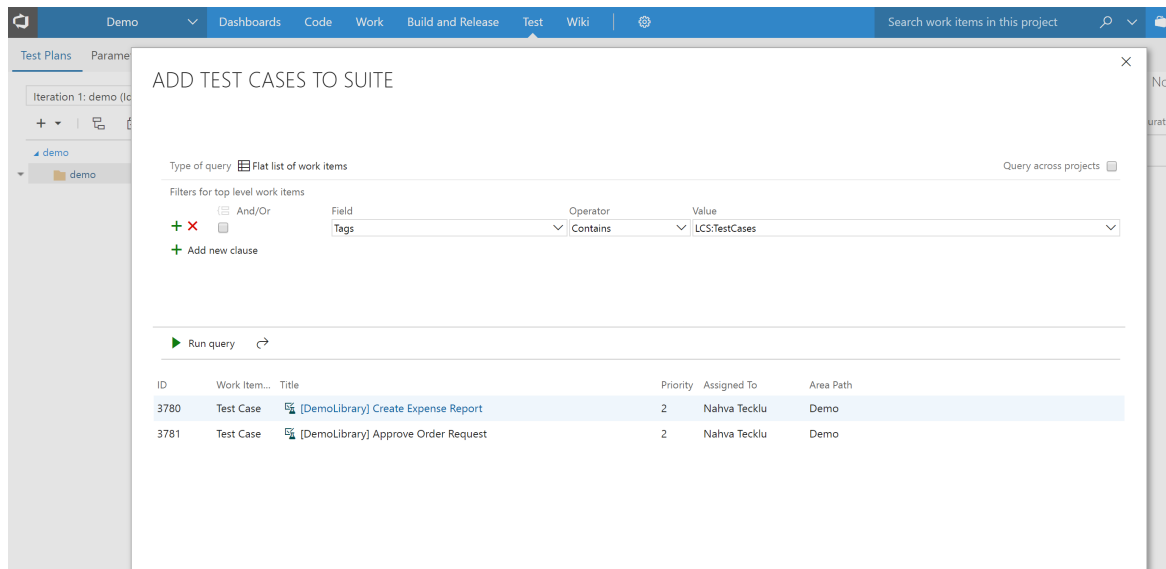
ID	Work item name
3761	[DemoLibrary] Create Expense Report
	Requirement name
	There are no requirements
	Test case name
3780	[DemoLibrary] Create Expense Report

In addition to the test steps, the task recording XML file is attached to the Azure DevOps test case. This file will be needed if you want to automate test execution.

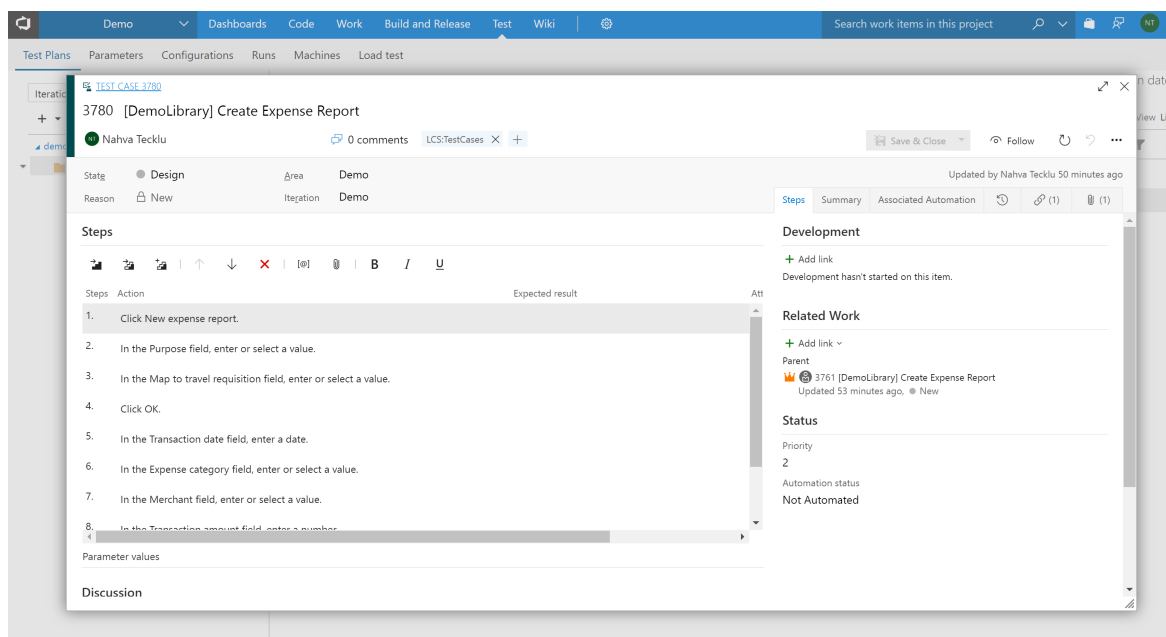
## Create a test suite in Azure DevOps

Next, you will need to create a test plan and test suite in Azure DevOps. This will allow you to execute an ordered suite of test cases and easily manage, investigate, and track the results.

1. Sign in to Azure DevOps and select the project and test plan that you want to test in.
2. On the toolbar, select **Test > Test Plans**.
3. In the left pane, select **+**, and then select **Static suite**.
4. Enter a name for the suite.
5. Click **Add existing** and query the tag **LCS:Test Cases**.
6. Click **Run > Add test cases**.



7. Select the test case to view details and the attached XML file.



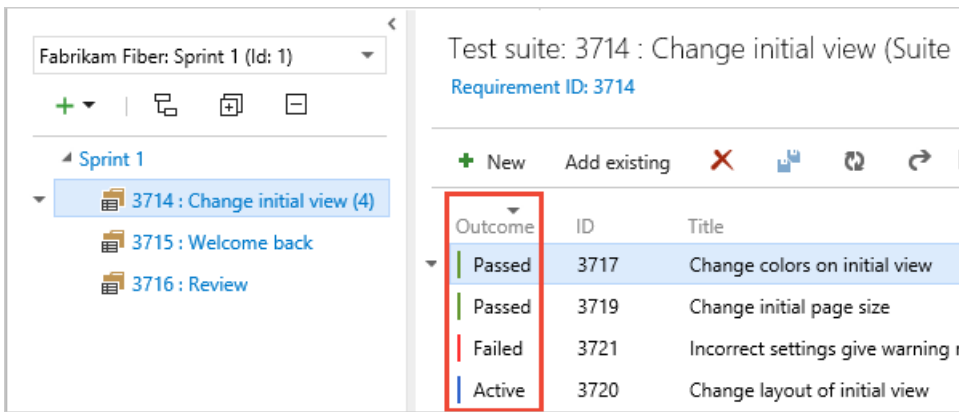
#### NOTE

This example shows how to create one comprehensive acceptance test suite with all test cases added. Instead, you should create various test suites under the same test plan and then use custom queries to add specific test cases to a test suite. A test case can belong to more than one test suite.

## Execute your tests

### Run manual test cases

After you have a test suite, you are ready to use it for regression testing after updates have been made to your application in a sandbox or test environment. You can run the test cases in your test suite manually or play the task recordings that are part of the test suite and use Azure DevOps to mark the test cases as passed or failed.



Azure DevOps also provides a tool, **Test Runner**, to manage manual test case execution. For more information about using Test Runner, see [Run manual tests](#).

We recommend that you take advantage of Azure DevOps as it provides a rich set of management features not only for testing, but result management and mitigation.

### Run automated test cases

The platform for Finance and Operations provides developers with tools to author test cases based on task recordings and use Azure DevOps to manage the automated execution of these test cases.

Developers can use the build and test automation capabilities of **build and test** environments. For details, see the [Continuous delivery home page](#).

Functional power users can automate the execution of their test cases using the **Regression suite automation tool**. For more information, [download the tool](#) and read the [Regression suite automation tool](#).

### Investigate test runs

Once an automated run is complete, on the Azure DevOps toolbar, select **Test > Runs** (or **Test Plans > Runs**) to investigate your test run. Select the desired test run to investigate test case failures and errors. You can also go to your test suite in Azure DevOps to see the latest results associated with your test cases. For more information on testing and test management in Azure DevOps, see the [Azure DevOps documentation](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Commerce evaluation environment overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic gives an overview of the Microsoft Dynamics 365 Commerce evaluation environment.

## NOTE

Commerce evaluation environments aren't generally available, and are granted to partners and customers on a per-request basis. For more information, reach out to your Microsoft partner contact.

## Overview

The Commerce evaluation environment is an optional end-to-end environment of Dynamics 365 Commerce that lets partners and potential customers try out the Commerce product.

Evaluation environments are partially preconfigured to reduce the required post-provisioning steps.

Aside from some minor limitations that don't affect features or functionality, the Commerce evaluation environment provides the complete Commerce experience, and can be used by customers and implementation partners to evaluate the product, provide feedback, and do fit/gap analysis.

## Limitations of the Commerce evaluation environment

Although the Commerce evaluation environment provides the full set of Commerce features and functionality, there are some minor limitations:

- Although the Commerce evaluation environment itself has no geographical limitations, components of the environment, such as the Retail Cloud Scale Unit (RCSU) and e-Commerce applications, should be provisioned only in the United States.
- Use of the Commerce evaluation environment is limited to 30 days from the date when e-Commerce is provisioned.
- The Commerce evaluation environment can be successfully deployed and initialized only in an environment that uses the demo topology, where all components of an environment are deployed on a single cloud-hosted virtual machine (VM). The main limitation of this topology is the number of concurrent users that the environment can support.

## Get started

To provision the Commerce evaluation environment, see [Provision a Commerce evaluation environment](#).

## Additional resources

[Provision a Dynamics 365 Commerce evaluation environment](#)

[Configure a Dynamics 365 Commerce evaluation environment](#)

[Configure BOPIS in a Dynamics 365 Commerce evaluation environment](#)

[Configure optional features for a Dynamics 365 Commerce evaluation environment](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Provision a Dynamics 365 Commerce evaluation environment

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic explains how to provision a Microsoft Dynamics 365 Commerce evaluation environment.

Before you begin, we recommend that you take a quick scan through this topic to get an idea of what the process requires.

## NOTE

Commerce evaluation environments aren't generally available, and are granted to partners and customers on a per-request basis. For more information, reach out to your Microsoft partner contact.

## Overview

To successfully provision a Commerce evaluation environment, you must create a project that has a specific product name and type. The environment and Commerce Scale Unit (CSU) also have some specific parameters that you must use when you expect to provision e-Commerce later. The instructions in this topic describe all the steps that are required to complete provisioning and the parameters that you must use.

After you successfully provision your Commerce evaluation environment, you must complete a few post-provisioning steps to prepare it for use. Some steps are optional, depending on the aspects of the system that you want to evaluate. You can always complete the optional steps later.

For information about how to configure your Commerce evaluation environment after you provision it, see [Configure a Commerce evaluation environment](#). For information about how to configure optional features for your Commerce evaluation environment, see [Configure optional features for a Commerce evaluation environment](#).

## Prerequisites

The following prerequisites must be in place before you can provision your Commerce evaluation environment:

- You have been onboarded into the evaluation program and granted capacity for an evaluation environment.
- You have access to the Microsoft Dynamics Lifecycle Services (LCS) portal.
- You are an existing Microsoft Dynamics 365 partner or customer and are able to create a Dynamics 365 Commerce project.
- You have administrator access to your Microsoft Azure subscription, or you're in contact with a subscription administrator who can assist you if required.
- You have your Azure Active Directory (Azure AD) tenant ID available.
- You've created an Azure AD security group that can be used as an e-Commerce system admin group, and you have its ID available.
- You've created an Azure AD security group that can be used as a Ratings and Reviews moderator group, and you have its ID available. (This security group can be the same as the e-Commerce system admin group.)

Note that this list isn't exhaustive. If you experience any issues, reach out to your Microsoft partner contact for assistance.

# Provision your Commerce evaluation environment

These procedures explain how to provision a Commerce evaluation environment. After you successfully complete them, the Commerce evaluation environment will be ready for configuration. All the activities that are described here occur in the LCS portal.

## Create a new project

To create a new project in LCS, follow these steps.

1. On the LCS home page, select the plus sign (+) to create a project.
2. In the right pane, follow one of these steps:
  - If you're a partner, select **Migrate, create solutions, and learn**.
  - If you're a customer, select **Prospective presales**.
3. Enter a name, description, and industry.
4. In the **Product name** field, select **Dynamics 365 Commerce**.
5. In the **Product version** field, select **Dynamics 365 Commerce**.
6. In the **Methodology** field, select **Dynamics Retail implementation methodology**.
7. Optional: You can import roles and users from an existing project.
8. Select **Create**. The project view appears.

## Add the Azure Connector

To add the Azure Connector to your LCS project, follow the steps in [Complete the Azure Resource Manager \(ARM\) onboarding process](#).

## Deploy the environment

To deploy the environment, follow these steps.

### NOTE

You might not have to complete steps 6, 7, and/or 8, because pages that have a single option are skipped. When you're in the **Environment parameters** view, confirm that the text **Dynamics 365 Commerce - Demo (10.0.x with Platform update xx)** appears directly above the **Environment name** field. For details, see the illustration that appears after step 8.

1. On the top menu, select **Cloud-hosted environments**.
2. Select **Add** to add an environment.
3. In the **Application version** field, select the most current version. If you have a specific need to select an application version other than the most current version, do not select a version prior to **10.0.14**.
4. In the **Platform version** field, use the platform version that is automatically chosen for the application version you selected.

**Select application and platform version**

Application version  
10.0.11

Platform version  
Platform Update 35

5. Select **Next**.
6. Select **Demo** as the environment topology.

**Select environment topology**

Please select a topology to deploy

**DEMO**  
Deploy an environment for demonstration purposes.

**DEVTEST**  
Deploy an environment for development, test, build.

7. On the **Deploy environment** page, enter an environment name. Leave the advanced settings as they are.

## Deploy environment

Dynamics 365 Commerce (Preview) - Demo (10.0.6 with Platform update 30) v 76

Environment name

[Advanced settings](#)

---

Virtual Machine	Instances	Size
Demo	1	D13 v2 <span>▼</span>

---

Total deployments:  
1 x D13 v2

---

[By selecting this checkbox, you agree to the pricing and licensing terms below.](#)

[Microsoft Dynamics Software License Terms](#)

[Privacy Statement](#)

[Azure price list](#)

---

[Next](#) [Cancel](#)

8. Adjust the VM size as required. (We recommend VM stock keeping unit [SKU] D13 v2.)
  9. Review the pricing and licensing terms, and then select the check box to indicate that you agree to them.
  10. Select **Next**.
  11. On the deployment confirmation page, verify that the details are correct, and then select **Deploy**. You're returned to the **Cloud-hosted environments** view, and your environment should appear in the list.
- Your requested environment will appear as queued and then deploying. The environment workflows will take some time to be completed. Therefore, check back after approximately six to nine hours.

12. Before you continue, make sure that the status of your environment is **Deployed**.

### Initialize the Commerce Scale Unit (cloud)

To initialize the CSU, follow these steps.

1. In the **Cloud-hosted environments** view, select your environment in the list.
2. In the environment view on the right, select **Full details**. The environment details view appears.
3. Under **Environment features**, select **Manage**.
4. On the **Commerce** tab, select **Initialize**. The CSU initialization parameters view appears.
5. In the **Region** field, select the region that is the same or close to the region that you deployed the

environment to.

6. Leave the **Version** field as it is.
7. Select **Initialize**.
8. On the deployment confirmation page, verify that the details are correct, and then select **Yes**. The **Commerce management** view displays again, where the **Commerce** tab is selected. Your CSU has been queued for provisioning.
9. Before you continue, make sure that the status of your CSU is **Success**. Initialization takes approximately two to five hours.

If you can't find the **Manage** link in the environment details view, reach out to your Microsoft contact for assistance.

During the deployment process, you might receive the following error message:

Evaluation (demo/test) environments need to register the scale unit connector application <application ID> in headquarters.

If the CSU initialization fails and you receive this error message, make a note of the application ID, which is a globally unique identifier (GUID), and then follow the steps in the next section to register the CSU deployment application in Commerce headquarters.

### **Register the CSU deployment application in Commerce headquarters (if required)**

To register the CSU deployment application in Commerce headquarters, follow these steps.

1. In Commerce headquarters, go to **System administration > Setup > Azure Active Directory applications**.
2. In the **Client Id** column, enter the application ID from the CSU initialization error message that you received.
3. In the **Name** column, enter any descriptive text (for example, **CSU Eval**).
4. In the **User ID** column, enter **RetailServiceAccount**.
5. Retry the CSU initialization and deployment from LCS.

### **Initialize e-Commerce**

To initialize e-Commerce, follow these steps.

1. On the **e-Commerce** tab, review the evaluation consent, and then select **Setup**.
2. In the **e-Commerce environment name** field, enter a name. Be aware that this name will appear in some of the URLs that point to your e-Commerce instance.
3. In the **Commerce Scale Unit name** field, select your CSU in the list. (The list should have only one option.)

The **e-Commerce geography** field is set automatically.

4. Select **Next** to continue.
5. In the **Supported host names** field, enter any valid domain, such as `www.fabrikam.com`.
6. In the **AAD security group for system admin** field, enter the first few letters of the name of the security group that you want to use, and then select the magnifying glass symbol to view the search results. Select the correct security group in the list.
7. In the **AAD security group for ratings and review moderator** field, enter the first few letters of the name of the security group that you want to use, and then select the magnifying glass symbol to view the search results. Select the correct security group in the list.
8. Leave the **Enable ratings and review service** option set to **Yes**.

9. Select **Initialize**. The **Commerce management** view displays again, where the **e-Commerce** tab is selected. E-Commerce initialization has started.
10. Before you continue, wait until the status of e-Commerce initialization is **Initialization successful**.
11. Under **Links** in the lower right, make a note of the URLs for the following links:
  - **e-Commerce site** – The link to the root of your e-Commerce site.
  - **Commerce site builder** – The link to the site management tool.

## Next steps

To continue the process of provisioning and configuring your Commerce evaluation environment, see [Configure a Commerce evaluation environment](#).

## Additional resources

[Dynamics 365 Commerce evaluation environment overview](#)

[Configure a Dynamics 365 Commerce evaluation environment](#)

[Configure BOPIS in a Dynamics 365 Commerce evaluation environment](#)

[Configure optional features for a Dynamics 365 Commerce evaluation environment](#)

[Dynamics 365 Commerce evaluation environment FAQ](#)

[Microsoft Lifecycle Services \(LCS\)](#)

[Commerce Scale Unit \(cloud\)](#)

[Microsoft Azure portal](#)

[Dynamics 365 Commerce website](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure a Dynamics 365 Commerce evaluation environment

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how to configure a Microsoft Dynamics 365 Commerce evaluation environment after it's provisioned.

## Overview

Complete the procedures in this topic only after your Commerce evaluation environment has been provisioned. For information about how to provision your Commerce evaluation environment, see [Provision a Commerce evaluation environment](#).

After your Commerce evaluation environment has been provisioned end to end, additional post-provisioning configuration steps must be completed before you can start to evaluate the environment. To complete these steps, you must use Microsoft Dynamics Lifecycle Services (LCS) and Dynamics 365 Commerce.

## Before you start

1. Sign in to the [LCS portal](#).
2. Go to your project.
3. On the top menu, select **Cloud-hosted environments**.
4. Select your environment in the list.
5. In the environment information on the right, select **Log on to environment**. You will be sent to Commerce headquarters.
6. Make sure that the **USRT** legal entity is selected in the upper-right corner.

During post-provisioning activities in Commerce headquarters, make sure that the **USRT** legal entity is always selected.

## Configure the point of sale

### Associate a worker with your identity

To associate a worker with your identity, follow these steps in Commerce headquarters.

1. Use the menu on the left to go to **Modules > Retail and commerce > Employees > Workers**.
2. In the list, find and select the following record: **000713 - Andrew Collette**.
3. On the Action Pane, select **Commerce**.
4. Select **Associate existing identity**.
5. In the **Email** field to the right of **Search using email**, enter your email address.
6. Select **Search**.
7. Select the record that has your name.
8. Select **OK**.
9. Select **Save**.

### Activate Cloud POS

To activate Cloud POS, follow these steps in LCS.

1. On the top menu, select **Cloud-hosted environments**.
2. Select your environment in the list.
3. In the environment information on the right, select **Log on to Cloud Point of Sale**.
4. Select **Next** to open the **Before you start** dialog box.
5. Leave the **Server URL** field as it is. Select **Next**.
6. Sign in by using your Microsoft Azure Active Directory (Azure AD) account.
7. Under **Store name**, select **San Francisco**, and then select **Next**.
8. Under **Register and device**, select **SANFRAN-1**.
9. Select **Activate**. You're signed out and taken to the POS sign-in page.
10. You can now sign in to the Cloud POS experience by using operator ID **000713** and password **123**.

## Set up your site in Commerce

To start to set up your evaluation site in Commerce, follow these steps.

1. Sign in to site builder by using the URL that you made a note of when you initialized e-Commerce during provisioning (see [Initialize e-Commerce](#)).
2. Select the **Fabrikam** site to open the site setup dialog box.
3. Select the domain that you entered when you initialized e-Commerce.
4. Select **Fabrikam extended online store** as the default channel. (Make sure that you select the **extended online store**.)
5. Select **en-us** as the default language.
6. Leave the value of the **Path** field as it is.
7. Select **OK**. The list of pages on the site appears.

## Enable jobs

To enable jobs in Commerce, follow these steps.

1. Sign in to the environment (HQ).
2. Use the menu on the left to go to **Retail and commerce > Inquiries and reports > Batch jobs**.

The remaining steps of this procedure must be completed for each of the following jobs:

- Process retail order email notification
  - Product availability
  - P-0001
  - Synchronize orders job
3. Use the Quick Filter to search for the job by name.
  4. If the status of the job is **Executing**, follow these steps:
    - a. Select the record.
    - b. On the Action Pane, on **Batch job** tab, select **Change status**.
    - c. Select **Canceling**, and then select **OK**.

Optionally, you can also set the recurrence interval to one (1) minute for the following jobs:

- Process retail order email notification job
- P-0001 job
- Synchronize orders job

### Run full data synchronization



To run full data synchronization in Commerce, follow these steps in Commerce headquarters.

1. Use the menu on the left to go to **Modules > Retail and commerce > Headquarters setup > Commerce scheduler > Channel database**.
2. Select the channel that is named **scXXXXXXXXXX**.
3. On the Action Pane, select **Full data sync**.
4. Enter **9999** as the distribution schedule.
5. Select **OK**.
6. Select **OK**.

### Test credit card information to do test purchases

To perform test transactions on the site, you can use the following test credit card information:

- **Card number:** 4111-1111-1111-1111
- **Expiration date:** 10/20
- **Card verification value (CVV) code:** 737

#### IMPORTANT

Never, under any circumstances, try to use actual credit card information on the test site.

## Next steps

After the provisioning and configuration steps are completed, you can start to use your evaluation environment. Use the Commerce site builder URL to go to the authoring experience. Use the Commerce site URL to go to the retail customer site experience.

To configure optional features for your Commerce evaluation environment, see [Configure optional features for a Commerce evaluation environment](#).

## Additional resources

[Dynamics 365 Commerce evaluation environment overview](#)

[Provision a Dynamics 365 Commerce evaluation environment](#)

[Configure optional features for a Dynamics 365 Commerce evaluation environment](#)

[Configure BOPIS in a Dynamics 365 Commerce evaluation environment](#)

[Dynamics 365 Commerce evaluation environment FAQ](#)

[Microsoft Lifecycle Services \(LCS\)](#)

[Retail Cloud Scale Unit \(RCSU\)](#)

[Microsoft Azure portal](#)

[Dynamics 365 Commerce website](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure BOPIS in a Dynamics 365 Commerce evaluation environment

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how to configure buy online, pickup in store (BOPIS) in a Microsoft Dynamics 365 Commerce evaluation environment after the environment has been provisioned.

## Prerequisite

Complete the procedures in this topic only after your Commerce evaluation environment has been provisioned and configured. For information about how to provision and configure your environment, see [Provision a Dynamics 365 Commerce evaluation environment](#) and [Configure a Dynamics 365 Commerce evaluation environment](#).

After your Commerce environment has been provisioned and configured end to end, you can use this topic to enable BOPIS scenarios.

## Configure the POS

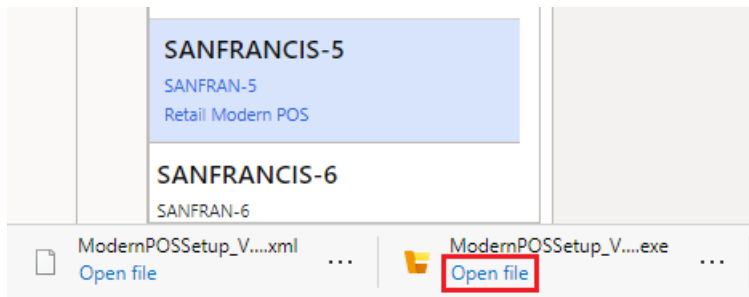
### Configure Modern POS

BOPIS scenarios that involve a credit card payment require a hardware station. The hardware station is built into Modern POS for Windows and Android clients. If you're using Cloud POS or Modern POS for iOS, the point of sale (POS) client must be paired with a shared hardware station. This topic explains how to configure BOPIS for Windows and Android clients. For information about how to set up a shared hardware station, see [Configure and install Retail hardware station](#).

1. Go to **Retail and Commerce > Channel setup > POS setup > Registers**.
2. Select register **SANFRAN-5**, and then select **Edit**.
3. Change the value of the **Hardware profile** field from **HW002** to **HW001**, and then select **Save**.
4. To synchronize the changes, go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
5. Select distribution schedule **1090**, and then, on the Action Pane, select **Run now**.
6. Select **Yes** and then **OK** to initiate data synchronization.

### Install Modern POS

1. Go to **Retail and Commerce > Channel setup > POS setup > Devices**.
2. Select device **SANFRANCIS-5**.
3. On the Action Pane, select **Download**, and then select **Configuration file**.
4. Select **Download**, and then select **Retail Modern POS**.
5. When download of the **ModernPOSSetup.exe** file is completed, select **Open file**.



6. Select **Next** to go through the installation process. When installation is completed, select **Close**.

### Activate Modern POS

1. On the Windows desktop, select the **Start** button, and enter **Retail Modern POS**.
2. Select the **Retail Modern POS** application to initiate activation.
3. Select **Next**. The **Server URL**, **Device ID**, and **Register number** fields should be preset by using information from the configuration file that you downloaded in the previous procedure.
4. Select **Activate**.
5. An authentication dialog box appears. Select the account that uses the email address that was previously associated with worker **000713 - Andrew Collette**.

#### NOTE

If you haven't yet associated a worker with your identity, activation will be unsuccessful. In this case, follow the steps under the "Associate a worker with your identity" section in the [Configure a Dynamics 365 Commerce evaluation environment](#) topic.

6. When you're prompted to let your organization manage the device, select **This app only**.
7. When activation is completed, select **Get started**.

### Enable BOPIS in Modern POS

1. Sign in to Modern POS by using **000713** as the operator ID and **123** as the password.
2. While the introductory walkthrough video is playing, select the two check boxes in the lower-left corner of the dialog box, and then close the dialog box.
3. If you aren't prompted to close the shift, scroll to the right on the **Welcome** page, select **Close shift**, and then sign back in to the POS.
4. After you're signed in, when you're prompted, select **Perform a non-drawer operation**.
5. On the **Welcome** page, scroll to the right, and select the **Select hardware station** operation.
6. Select **Manage**, set the **Use hardware station** option to **On**, and then select **OK**.
7. Sign out of the POS, and then sign back in.
8. After you're signed in, select **Open a new shift**, and then select **Drawer**.

## Complete a BOPIS scenario

### Create a storefront order for in-store pickup

1. Go to the URL that you specified in the [Initialize e-Commerce](#) step during environment configuration.
2. Select an item, and select **Add to cart**.
3. On the shopping bag page, select **Pick this up** for the order line that you just added.
4. In the **Select a store** dialog box, enter **San Francisco**, and then select the **Search** button.

- In the list of results, find the San Francisco store, and select **Pick up here**.
- On the shopping bag page, select **Checkout as Guest**.

#### **NOTE**

To continue with checkout, you must accept the cookie notice. This notice appears as a banner at the top of the checkout page.

- For the credit card payment method, enter the following details:

- **Cardholder name:** Enter any name.
- **Card number:** Enter 4111-1111-1111-1111.
- **Expiration date:** Enter 10/20.
- **Card verification value (CVV) code:** Enter 737.

#### **IMPORTANT**

Never, under any circumstances, try to use actual credit card information on the test site.

- Continue with checkout by entering details of the billing address, and then select **Save and continue**.
- When the order is ready to be placed, select **Checkout**.

### **Synchronize online orders to the back office**

For information about how to synchronize online orders, see [Posting of online sales and payments](#).

### **Pick up an order in the store**

- Sign in to the POS.
- On the **Welcome** screen, select **Order fulfillment**
- In the list of items for pickup, select the line from the order that was placed online.
- While the order line is selected, select **Pick up**.

The line item is added to the transaction screen, and **\$0.00** is shown as the balance that is due.

- Select the balance due of **\$0.00**, or select any payment method to conclude the transaction.

## Troubleshooting

### **Online orders that are retrieved in the POS have a non-zero balance due**

When an order is retrieved for in-store pickup, if the balance due isn't 0 (zero), make sure that Modern POS is being used, and that the hardware station is active. If Cloud POS or Modern POS for iOS is being used, make sure that a shared hardware station is active. Some form of active hardware station is required to retrieve payments that were made online.

### **General issues with payment capture**

For all general issues, you should always consult the Modern POS or Internet Information Services (IIS) Hardware Station event logs as a first step. You can find these logs under the following nodes in the Windows event log:

- Application and Services Logs > Microsoft > Dynamics > Commerce-ModernPOS
- Application and Services Logs > Microsoft > Dynamics > Commerce-Hardware Station

# Additional resources

[Dynamics 365 Commerce evaluation environment overview](#)

[Provision a Dynamics 365 Commerce evaluation environment](#)

[Configure optional features for a Dynamics 365 Commerce evaluation environment](#)

[Dynamics 365 Commerce evaluation environment FAQ](#)

[Microsoft Lifecycle Services \(LCS\)](#)

[Retail Cloud Scale Unit \(RCSU\)](#)

[Microsoft Azure portal](#)

[Dynamics 365 Commerce website](#)

[Adyen payment connector](#)

[Saving online payment instruments with the Adyen connector](#)

[Omni-channel payments overview](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure optional features for a Dynamics 365 Commerce evaluation environment

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how to configure optional features for a Microsoft Dynamics 365 Commerce evaluation environment.

## Prerequisites

If you want to evaluate the transactional email features, the following prerequisites must be met:

- You have an available email server (Simple Mail Transfer Protocol [SMTP] server) that can be used from the Microsoft Azure subscription where you provisioned the evaluation environment.
- You have the server's fully qualified domain name (FQDN)/IP address, SMTP port number, and authentication details available.

## Configure the image back end

### Find your media base URL

#### NOTE

Before you can complete this procedure, you must complete the steps in [Set up your site in Commerce](#).

1. Sign in to the Commerce site builder by using the URL you made a note of when you initialized e-Commerce during provisioning (see [Initialize e-Commerce](#)).
2. Open the **Fabrikam** site.
3. On the menu on the left, select **Media Library**.
4. Select any single image asset.
5. In the property inspector on the right, find the **Public URL** property. The value is a URL. Here is an example:

```
https://images-us-sb.cms.commerce.dynamics.com/cms/api/fabrikam/imageFileData/MA1nQC .
```

6. Replace the last identifier in the URL (**MA1nQC** in the preceding example) with the string **search?fileName=**. Here is what the example URL looks like after this change is made:

```
https://images-us-sb.cms.commerce.dynamics.com/cms/api/fabrikam/imageFileData/search?fileName=
```

This URL is your media base URL. Make a note of it.

### Update the media base URL

1. Sign in to Commerce headquarters.
2. Use the menu on the left to go to **Modules > Retail and commerce > Channel setup > Channel profiles**.
3. Select **Edit**.
4. Under **Profile properties**, replace the value for the **Media Server Base URL** property with the media base URL that you created earlier.

5. Select the channel that is named `scXXXXXXXXXX`.
6. Under **Profile properties**, select **Add**.
7. For the property that was added, select **Media Server Base URL** as the property key. As the property value, enter the media base URL that you created earlier.
8. Select **Save**.

## Configure and test the email server

### NOTE

The SMTP server or email service that you enter here must be accessible from the Azure subscription that you're using for the environment.

1. Sign in to Commerce headquarters.
2. Use the menu on the left to go to **Modules > Retail and Commerce > Headquarters setup > Parameters > Email parameters**.
3. On the **SMTP settings** tab, in the **Outgoing mail server** field, enter the FQDN or IP address of your SMTP server or email service.
4. In the **SMTP port number** field, enter the port number. (If you aren't using Secure Sockets Layer [SSL], the default port number is 25.)
5. If authentication is required, enter values in the **User name** and **Password** fields.
6. Select **Save**.
7. Select **Refresh**.
8. On the **Test email** tab, in the **Email provider** field, select **SMTP**.
9. In the **Send to** field, enter the email address that the test email should be delivered to.
10. Select **Send test email**.

## Configure email templates

For each transactional event that you want to send emails for, you must update the email template with a valid sender email address.

1. Sign in to Commerce headquarters.
2. Use the menu on the left to go to **Modules > Retail and Commerce > Headquarters setup > Parameters > Organization email templates**.
3. Select **Show list**.
4. For each template in the list, follow these steps:
  - a. In the **Sender email** field, enter the sender email address.
  - b. Optional: In the **Sender name** field, enter the name that should be used as the sender name.
5. Select **Save**.

## Customize email templates

You might want to customize the email templates so that they use different images. Or you might want to update the links in the templates so that they go to your evaluation environment. This procedure explains how to download the default templates, customize them, and update the templates in the system.

1. In a web browser, download the [Microsoft Dynamics 365 Commerce Evaluation default email templates zip file](#) to your local computer. This file contains the following HTML documents:

- Order confirmation template
  - Issue gift card template
  - New order template
  - Pack order template
  - Pick order template
2. Customize the templates by using a text or HTML editor. See the list of [supported tokens](#) later in this topic.
  3. Sign in to Commerce.
  4. Use the menu on the left to go to **Modules > Organization administration > Setup > Organization email templates**.
  5. Expand the list on the left to see all the templates.
  6. For each template that you want to customize, follow these steps:
    - a. Select the template in the list.
    - b. Under **Email message content**, select the appropriate language version in the list. (The default language is **en-us**.)
    - c. Under **Email message content**, select **Edit**. The **Upload email template** pane appears.
    - d. Select **Browse**, and find the HTML file that has the customized content.
    - e. Select **Upload**. The template is uploaded into the system, and a preview is shown.
    - f. Select **OK**.
    - g. Optional: Customize the **Subject** property of the template.
    - h. Select **Save**.

### Supported tokens in the email template

When the email is rendered, these tokens are replaced with actual values that apply to the customer and the customer's order.

#### Sales order

The following tokens apply to the overall sales order.

NAME OF THE TOKEN	TOKEN
Order number	%salesid%
Customer's name	%customername%
Delivery address	%deliveryaddress%
Billing address	%customeraddress%
Order date	%shipdate%
Delivery mode	%modeofdelivery%
Discount	%discount%
Sales tax	%tax%
Order total	%total%



## Sales line

The following tokens are replaced with values for each product in the order.

### NOTE

Put the **Product list - start** token at the beginning of the HTML block that is repeated for every product, and put the **Product list - end** token at the end of the block.

NAME OF THE TOKEN	TOKEN
Product list - start	<!--%tablebegin.salesline% -->
Product list - end	<!--%tableend.salesline%-->
Product name	%lineproductname%
Description	%lineproductdescription%
Quantity	%linequantity%
Line unit price	%lineprice% (verify)
line item total	%linenetamount%
line discount	%linediscount%
Ship date	%lineshipdate%
Procurement method	%linedeliverymode%
delivery address	%linedeliveryaddress%
Sales unit of the line	%lineunit%

## Additional resources

[Dynamics 365 Commerce evaluation environment overview](#)

[Provision a Dynamics 365 Commerce evaluation environment](#)

[Configure a Dynamics 365 Commerce evaluation environment](#)

[Configure BOPIS in a Dynamics 365 Commerce evaluation environment](#)

[Dynamics 365 Commerce evaluation environment FAQ](#)

[Microsoft Lifecycle Services \(LCS\)](#)

[Retail Cloud Scale Unit \(RCSU\)](#)

[Microsoft Azure portal](#)

[Dynamics 365 Commerce website](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Commerce evaluation environment FAQ

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides answers to frequently asked questions about the Microsoft Dynamics 365 Commerce evaluation environment.

## **Can we use the Commerce evaluation environment as an e-Commerce storefront for customers that currently implement Retail?**

No. The Commerce evaluation environment is only for evaluation. If you require an environment for a customer that implements Retail, contact Microsoft.

## **Can the Commerce evaluation environment be used to provision the e-Commerce features on top of an existing application/environment that implements Retail?**

No (mostly). The Commerce evaluation components are available only to environments that match the configurations that are specified in the prerequisites and provisioning guide. Additionally, the required base demo data won't be available in environments that were deployed with an initial release that is earlier than 10.0.8.

## **What costs are involved in deploying the Commerce evaluation environment on Microsoft Azure via Microsoft Dynamics Lifecycle Services (LCS)?**

A traditional Dynamics 365 Finance/Dynamics 365 Supply Chain Management/Dynamics 365 Commerce headquarters demo environment (virtual machine [VM]) will be hosted in your Azure subscription. You can use the [Azure pricing calculator](#) to estimate this cost.

Other components such as Commerce Scale Unit, Commerce site builder, and your e-Commerce site will be available as software as a service (SaaS) and hosted by Microsoft.

## **Which Azure geographies are currently supported for the Commerce evaluation environment?**

The Commerce evaluation environment can be deployed only in the North America geography.

## **Is there a downloadable virtual hard disk (VHD) that has the complete OneBox virtual machine (VM) option?**

Dynamics 365 Commerce and Commerce Scale Unit are completely software as a service (SaaS) and must be cloud-hosted.

## **How long can the Commerce evaluation environment be used?**

The Commerce evaluation environment has a 30-day time limit from the date when SaaS components such as Commerce Scale Unit, Commerce site builder, and your e-Commerce site are provisioned.

## **Can I extend the time limit for my Commerce evaluation environment?**

Extension of the time limit is an exception to the norm and is considered on a case-by-case basis. You should reach out to your Microsoft partner contact for assistance.

## **Additional resources**

[Dynamics 365 Commerce evaluation environment overview](#)

Provision a Dynamics 365 Commerce evaluation environment

Configure a Dynamics 365 Commerce evaluation environment

Configure BOPIS in a Dynamics 365 Commerce evaluation environment

Configure optional features for a Dynamics 365 Commerce evaluation environment

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Deployment options

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can deploy Finance and Operations apps in the cloud or on-premises. Cloud deployments offer an ERP service that is fully managed by Microsoft, while on-premises deployments are deployed locally within a customer's data center.

## IMPORTANT

On-premises deployments are not supported on any public cloud infrastructure, including Azure.

The following table provides a comparison of the capabilities provided by the two deployment options.

Dynamics 365 deployment options		
Capability	Cloud	On-premises
Infrastructure and data location	<ul style="list-style-type: none"><li>Full Microsoft managed cloud service</li><li>Microsoft managed data centers</li></ul>	<ul style="list-style-type: none"><li>Customer or partner managed infrastructure</li><li>Disconnected data center</li><li>Local data residency</li></ul>
Data trustee	<ul style="list-style-type: none"><li>Microsoft</li></ul>	<ul style="list-style-type: none"><li>Customer</li></ul>
Application lifecycle management (ALM)	<ul style="list-style-type: none"><li>Managed by Microsoft</li><li>Customer access to ALM and telemetry using Lifecycle Services (LCS)</li></ul>	<ul style="list-style-type: none"><li>Managed by customer or partner with cloud-based ALM and telemetry using LCS</li></ul>
Cloud capabilities	<ul style="list-style-type: none"><li>High availability and disaster recovery included</li><li>Sandbox environments</li></ul>	<ul style="list-style-type: none"><li>Automated deployment and telemetry using LCS</li></ul>
Intelligence and analytics	<ul style="list-style-type: none"><li>Author and publish Power BI reports</li><li>Ready-made Analytical workspaces</li><li>Pinning tiles and Reports from PowerBI.com</li></ul>	<ul style="list-style-type: none"><li>Author and publish PowerBI reports</li></ul>
Updates, health monitoring	<ul style="list-style-type: none"><li>Provided through LCS</li></ul>	<ul style="list-style-type: none"><li>Provided through LCS</li></ul>
Licensing	<ul style="list-style-type: none"><li>Subscription</li></ul>	<ul style="list-style-type: none"><li>License with Software Assurance/Business Ready Enhancement Plan, or subscription</li></ul>

## Why cloud

Cloud deployments provide a cloud service that is easy to scale up or down as needed, as well as data centers that are fully managed by Microsoft. The time spent implementing Finance and Operations apps can be significantly shortened, fewer customizations may be required, and the costs of IT hardware and infrastructure are lower.

Cloud deployments include high availability, disaster recovery, sandbox environments, and application lifecycle management combined with cloud-based systems of intelligence, infrastructure, compute, and database services in a single offering. When needed, data failover in the cloud, automated deployment and continuous updates, and elastic compute capacity are available. A cloud deployment also provides data aggregation, financial reporting, and intelligence.

The cloud service provides customers with the greatest value, the broadest range of functionality, the best application lifecycle experience, the easiest and broadest integration with Microsoft Azure services, the best option for business insights and intelligence, and the most value for customers' technology investments.

## Why on-premises

With an on-premises deployment, existing data center investments can be leveraged. Customers can also configure their enterprise preferences to meet the regulatory and compliance needs of their business, comply with data sovereignty rules in regions where there are no Azure Data Centers, or ensure business continuity in areas with limited public infrastructure.

A customer's business data and processes are disconnected from the cloud and are stored and run locally in the customer's or their partner's data center. Some connectivity is required for system management and updates which are enabled through Microsoft Dynamics Lifecycle Services (LCS), a cloud-based application lifecycle management service. Customer data that is related to the configuration and application customization may be stored in the cloud.

For customers who choose to run Finance and Operations apps in their own data center, the on-premises deployment option will have a similar user-interface and application functionality as other deployment options. However, customers must take on the following responsibilities:

- Stand up their own infrastructure.
- Configure their own high-availability and disaster recovery solutions.
- Stand up sandbox environments.
- Manage their infrastructure, including scheduling operating system updates.

The additional costs to deploy and manage these capabilities might lead to higher deployment costs and a greater Total Cost of Ownership (TCO). Tools for deploying the Finance and Operations apps and updates will be available to partners and customers via Lifecycle Services. Unlike the cloud deployment option, Advanced Analytics and Azure Machine Learning are not included in the on-premises deployment option.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Comparison of cloud and on-premises features

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic shows a comparison of features available in cloud vs. on-premises for the following applications:

- [Dynamics 365 Finance](#)
- [Dynamics 365 Supply Chain Management](#)
- [Dynamics 365 Commerce](#)
- [Dynamics 365 Human Resources](#)

Information about the [development and administration features](#) is included, as well.

The following tables list the application areas. Cloud and on-premises support is listed for the feature as a whole. Where specific features differ from the area overall, the features are listed on a separate line in the Feature column.

## Dynamics 365 Finance

AREA	FEATURE	CLOUD	ON-PREMISES
Compliance and certifications		Yes	Yes
	SOC 1 Type 1 certification	Yes	No
Data management and integration		Yes	Yes
	Export data to your own data warehouse	Yes	Yes
	Enable the export of incremental updates to a data entity	Yes	Yes
	Data integrations	Yes	Yes
Document management		Yes	Yes
Financial management		Yes	Yes
Help		Yes	No
Human resources		Yes	Yes
Intelligence		Yes	Yes
	Electronic reporting (ER)	Yes	Yes
	ER: Integration with LCS	Yes	No

AREA	FEATURE	CLOUD	ON-PREMISES
	ER: Integration with SharePoint	Yes	No
	ER: Integration with Regulatory Configuration Services (RCS)	Yes	No
	ER: Uses local file system as storage of ER configurations accessible via ER repositories	No	Yes
	Integration with PowerBI.com	Yes	No
	Integration with PowerBI Desktop	No	Yes
	Analytical workspaces	Yes	No
	Intelligent business process: Recommendations	Yes	No
	Authoring Power BI reports with OData using Power BI desktop or Excel PowerQuery tools	Yes	No
	SQL Server Reporting Services (SSRS) supports scaling out	Yes	No
	Telemetry is transferred into the cloud	Yes	No
Lifecycle services		Yes	Yes
	Configurable business processes	Yes	No
Localizations		Yes	Yes
Mobile app, workspaces, and platform		Yes	Yes
Office integration		Yes	Yes
Organization administration		Yes	Yes
Payroll		Yes	Yes
	Direct deposit	Yes	No



AREA	FEATURE	CLOUD	ON-PREMISES
Project management and accounting		Yes	Yes
Security		Yes	Yes
Service management		Yes	Yes
Web client		Yes	Yes
	Task recorder - Save or load task recordings from the BPM library	Yes	No
Support		Yes	Yes
	Access to Support via the Help & Support menu	Yes	No
	Business events	Yes	Yes (either internet connectivity is required or custom endpoints must be implemented to send/receive business events within intranet)

## Dynamics 365 Supply Chain Management

AREA	FEATURE	CLOUD	ON-PREMISES
Asset management		Yes	No
Compliance and certifications		Yes	Yes
	SOC 1 Type 1 certification	Yes	No
Cost accounting		Yes	Yes
	Cost accounting content pack for Power BI	Yes	No
	Cost accounting workspace for mobile app	Yes	No
Cost management		Yes	Yes
	Cost management content pack for Power BI	Yes	No
Data management and integration		Yes	Yes

AREA	FEATURE	CLOUD	ON-PREMISES
	Configuration-driven extension	Yes	No
	Export data to your own data warehouse	Yes	Yes
	Enable the export of incremental updates to a data entity	Yes	Yes
	Data integrations	Yes	Yes
Document management		Yes	Yes
Help		Yes	No
Intelligence		Yes	Yes
	Electronic reporting (ER)	Yes	Yes
	ER: Integration with LCS	Yes	No
	ER: Integration with SharePoint	Yes	No
	ER: Integration with Regulatory Configuration Services (RCS)	Yes	No
	ER: Uses local file system as storage of ER configurations accessible via ER repositories	No	Yes
	Integration with PowerBI.com	Yes	No
	Integration with PowerBI Desktop	No	Yes
	Analytical workspaces	Yes	No
	Intelligent business process: Recommendations	Yes	No
	Authoring Power BI reports with OData using Power BI desktop or Excel PowerQuery tools	Yes	No
	SQL Server Reporting Services (SSRS) supports scaling out	Yes	No

AREA	FEATURE	CLOUD	ON-PREMISES
	Telemetry is transferred into the cloud	Yes	No
Inventory management		Yes	Yes
Lifecycle services		Yes	Yes
	Configurable business processes	Yes	No
Localizations		Yes	Yes
Manufacturing		Yes	Yes
Master planning and forecasting		Yes	Yes
Planning optimization		Yes	No
Mobile app, workspaces, and platform		Yes	Yes
Office integration		Yes	Yes
Organization administration		Yes	Yes
Procurement and sourcing		Yes	Yes
	Punch-out to external catalog from purchase requisition	Yes	No
	Purchase spend analysis Power BI reports	Yes	No
Product information management		Yes	Yes
Product master data		Yes	Yes
Production		Yes	Yes
	Production performance Power BI reports	Yes	No
Project management and accounting		Yes	Yes
Sales		Yes	Yes

AREA	FEATURE	CLOUD	ON-PREMISES
	Sales and profitability performance Power BI reports	Yes	No
Security		Yes	Yes
Service management		Yes	Yes
Supply chain management		Yes	Yes
Transportation management		Yes	Yes
Vendor collaboration		Yes	No
Warehouse management		Yes	Yes
	Mobile warehouse app	Yes	Yes
	Warehousing Power BI reports	Yes	No
Web client		Yes	Yes
	Task recorder - Save or load task recordings from the BPM library	Yes	No
Support		Yes	Yes
	Access to Support via the Help & Support menu	Yes	No

## Dynamics 365 Commerce

To see a list of capabilities that are available in on-premises deployments, see [Commerce capabilities that are available in on-premises deployments](#).

## Dynamics 365 Human Resources

AREA	FEATURE	CLOUD	ON-PREMISES
All Human Resources areas	All Human Resources features	Yes	No

## Development and administration features

AREA	FEATURE	CLOUD	ON-PREMISES
Build and test		Yes	Yes

AREA	FEATURE	CLOUD	ON-PREMISES
Extensibility		Yes	Yes
Monitoring and telemetry		Yes	Yes
Platform compatibility		Yes	Yes
Servicing		Yes	Yes
	Servicing environments	Yes	No
Trace Parser		Yes	Yes
PerfTimer		Yes	Yes*
Upgrade		Yes	Yes
	Upgrade	Yes	No
	Upgrade and support for previous versions	Yes	No
Visual Studio development		Yes	Yes

\* In on-premises environments, PerfTimer only shows results for the client.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cloud deployment overview

2/18/2021 • 15 minutes to read • [Edit Online](#)

Working with Microsoft to deploy Finance and Operations apps in the cloud requires that you understand the environment and subscription that you are deploying to, who can perform which tasks, and the data and customizations that you need to manage. We recommend that you sign up for the Full Microsoft FastTrack for Dynamics 365 to help speed your deployment and implementation - it's a program that provides training and consulting to help you realize business value faster. For more information, see [Microsoft FastTrack](#). If you choose to use the Essentials FastTrack program instead, you will be using the Implementation Project Methodology in Lifecycle Services (LCS) to help you manage your implementation project.

## Customer lifecycle, subscriptions, and environment types

Microsoft assumes that all customers will follow a lifecycle similar to the following for all cloud deployments, and therefore need different environment topologies at each phase.

- Evaluate
- Develop customizations, if needed.
- Curate a "golden configuration" environment that contains only module configurations without master or transactional data. This is to be the baseline for your data migration testing and eventual go live.
- Install and test customizations and partner solutions on a tier-1 sandbox (Development or test environment).
- Test customizations, partner solutions and data configuration on a tier-2 sandbox environment.
- Deploy customizations and data configurations to a production environment with high availability.

At some phases of a project, you may have all of the environments live at once. For more information, about the default licenses and tiers that are available, see the [Dynamics 365 Licensing Guide](#).

You may notice the terms cloud hosted or Microsoft subscriptions. A *cloud hosted subscription* means that the customer or partner brings their own Azure subscription and deploys Finance and Operations apps to it, for evaluation and development purposes only. The customer or partner pays for the resources deployed to their Azure subscription based on the Azure price list. A *Microsoft subscription* means that the customer purchases Finance and Operations licenses, which will then allow them to deploy environments to an Azure subscription which is managed by Microsoft, therefore, the customer has no separate Azure billing.

With each Enterprise offer, two environments are included by default:

- One Tier 2 sandbox (multi-box environment) for user acceptance testing (UAT).
- One production environment with high availability (HA).

Additional environments may be purchased as add-ons. For information about licensing and what is included in Microsoft Dynamics 365, see the [Dynamics 365 Licensing Guide](#).

Here's how the lifecycle maps to the available environments. If you already have environments deployed in your Lifecycle Services project, you can find the Environment Type and Environment Sub type on each environment's details page.

LIFECYCLE PHASE	ENVIRONMENT TIER	SUBSCRIPTION	ENVIRONMENT TYPES	ENVIRONMENT SUB-TYPE
Evaluation and analysis	Tier 1 Sandbox	Cloud hosted	Customer Managed	Demo

LIFECYCLE PHASE	ENVIRONMENT TIER	SUBSCRIPTION	ENVIRONMENT TYPES	ENVIRONMENT SUB-TYPE
Customize	Tier 1 Sandbox	Cloud hosted or VHD	Customer Managed	Develop
Golden configuration	Tier 1 Sandbox	Cloud hosted	Customer Managed	Develop
User acceptance testing (UAT)	Tiers 2-5 Sandbox	Microsoft	Microsoft Managed or Self-service	Not applicable
Go live	Production	Microsoft	Microsoft Managed or Self-service	Not applicable

*Tiers 2-5 can be purchased to increase performance of the environment. The higher the tier, the more compute and database capacity is reserved for your use. For more information about Self-service environment types, check out the [Self-service deployment overview](#).*

### Environment lifecycle operations

Users with the Environment Administrator or Project Owner roles in Lifecycle Services can perform various lifecycle operations on their environments. These operations often involve downtime on the environment until the task is finished. Each of these operations are located under or next to the **Maintain** button on each environment details page.

LIFECYCLE OPERATION	DESCRIPTION	LEARN MORE
Apply software	Install Microsoft updates, ISV solutions, or your own customization packages.	<a href="#">Apply updates to cloud environments</a>
Enable access	Allow list your IP for Remote Desktop or database access	See the <a href="#">Remote Desktop</a> section later in this topic
Restart services	Ability to restart components of your environment	<a href="#">Restart environment services</a>
Move database	Full data lifecycle management	<a href="#">Database movement operations</a>
Maintenance mode	Ability to change configuration with only admin access	<a href="#">Maintenance mode</a>
Upgrade	Upgrade code and data from 7.x to the latest version	<a href="#">Process for moving to the latest update</a>
Deallocate	Ability to turn off an environment not being used, or to troubleshoot a failed action	Not applicable
Start	Ability to turn on an environment for use	Not applicable
Delete	Ability to delete an environment previously deallocated	Not applicable

## Security and compliance

Finance and Operations is PA-DSS 3.1 certified which means that all communications between components are secured out-of-the-box.

All Finance and Operations front-end virtual machines in Microsoft Azure are configured during deployment to only accept TLS 1.2.

#### **IMPORTANT**

Customers who have administrator access to Microsoft-managed sandboxes, including any add-on sandboxes purchased, must follow these guidelines:

- By default, automatic Windows update is enabled for all Tier 1 - 5 sandboxes and should NOT be disabled. This ensures that any time that Microsoft pushes security or critical infrastructure updates to your environment, your environment receives the latest set of updates and is updated each month with the operating system fixes that Microsoft releases.
- Admin passwords on these environments should NOT be changed. Environments that have admin passwords changed will be flagged by Microsoft. Microsoft reserves the right to, and will reset the admin password.
- Adding new user accounts to any Microsoft managed VM is NOT permitted. Microsoft reserves the right to, and will remove the newly added user accounts without providing notice.

Finance and Operations is not covered by a FedRAMP ATO at this time. If Finance and Operations is provisioned in the United States, all customer data at rest is stored in data centers located in the United States, as described in [International availability of Dynamics 365](#). Finance and Operations does not support any other Dynamics 365 US Government or Microsoft 365 GCC compliance attributes (for example, access by US screened personnel, and support for CJIS and IRS 1075).

## Remote Desktop

### **Microsoft-managed environments**

#### **WARNING**

Microsoft will be removing the use of Remote Desktop by customers and partners. Each environment will first have administrator access removed, but still allow non-administrator access to the virtual machines. After this, all access will be removed. For each step of this phased removal, an email notification will be sent to the Notification list setup for each environment. All Remote Desktop access will be removed by November 2020.

Customers are required to complete additional setup to connect to virtual machines (VMs) through Microsoft Remote Desktop (RDP). This additional setup applies to all Microsoft-managed environments, including Tier 1 through Tier 5 sandboxes and add-ons. In order to connect to Tier 1 through Tier 5 sandbox environments, you must explicitly enable access (safe list) from your organization's IP address space. This can be done by a Lifecycle Services (LCS) user who has access to the **Environment** page (**Maintain** > **Enable Access**) where they can enter the IP address space that will be used to connect to the virtual machines through Remote Desktop. Access rules are either a single IP address (example: 10.10.10.10) or an IP address range (example: 192.168.1.0/24). You may add multiple entries at once as a semi-colon(;) separated list (example: 10.10.10.10;20.20.20.20;192.168.1.0/24). These entries are used to configure the Azure Network Security Group that is associated with your environment's virtual network. For more information, see [Security rules](#).



### IMPORTANT

Customers need to ensure that RDP endpoints are secured through explicit IP safe list rules as mentioned above. The IP safe list rules must adhere to the following conditions.

- IP safe list rules must NOT use asterisk/zero.
- Wide IP address ranges must NOT be used.
- IP address ranges must restrict to the customer's CORPNET.
- If computers outside the customer's CORPNET (such as a home office) are used to connect to sandbox environments, only the specific IP addresses of the computers used to connect to the sandbox environments must be added.
- Azure Datacenter IP address ranges must NOT be added.
- Public IP addresses, such as a coffee shop location, must NOT be added.
- IP safe list rules should be removed when not in use. Periodic review of environment IP safe list rules is recommended.

Microsoft will run periodic tests on the Microsoft Managed environments validating that the environments are sufficiently restricted. Microsoft reserves the right to and will remove any IP Address safe list rules that violate the above guidelines, immediately without providing notice.

### Partner/Customer managed environments

By default, Remote Desktop is enabled for all non-Microsoft managed environments. We recommend that customers restrict access to any environments that belong to their subscriptions. This can be done by configuring Network Security Group rules on the environments directly in Azure Portal.

## Windows Remoting (WinRM)

Windows Remoting (WinRM) is disabled on all environments. Although you can enable WinRM on environments that belong to your subscriptions through Azure Portal, we strongly recommend that you do not do this.

### WARNING

Exceptions to enable WinRM will not be granted for any Microsoft-managed environments.

## Availability

The guaranteed uptime for Finance and Operations apps is 99.9%. Planned downtime occurs once a month and lasts no longer than eight hours. Because the work completed during the downtime doesn't always take eight hours, we will always communicate the estimated amount of time that your environments will be down. For more information, see [Get support for Finance and Operations apps or Lifecycle Services \(LCS\)](#).

### High-availability features

To ensure service availability, all production environments are protected by using default Azure high availability (HA) features. HA functionality provides ways to avoid downtime caused by the failure of a single node within a datacenter, and DR features protect against outages broadly impacting an entire datacenter. Azure availability sets are used to prevent single-point-of-failure events. For more information about Azure availability sets, see [Use availability zones to protect from datacenter level failures](#). High availability for databases is supported through Azure SQL. For more information, see [Overview of business continuity with Azure SQL Database](#).

### Disaster recovery features

Production environments are configured with Azure disaster recovery support that includes the following:

- Azure SQL active-geo replication is configured for the Finance and Operations database of the production

environment. For more information about SQL replication, see [Compare geo-replication with failover groups](#).

- Geo-redundant copies of Azure blob storage (containing document attachments) in other Azure regions. For more information, see [Azure Storage redundancy](#).
- Same secondary region for the Azure SQL and Azure blob storage replication.

Only primary data stores are supported by replication. The Financial reporting services and Entity store database use transformed data from the primary database and must be generated after the recovery site has been set up and the Finance and Operations service has started.

## Service availability in Azure Regions

Finance and Operations apps can be deployed into a subset of Microsoft Azure datacenters using Dynamics Lifecycle Services (LCS). Azure is generally available in datacenters and geographical locations around the world. With Finance and Operations apps, customers can specify the region or datacenter where their customer data will be stored. Microsoft may replicate data to other regions for data durability, but we will not replicate or move customer data outside the geographical location. For more details, see the [Service description white paper](#).

### IMPORTANT

Regardless of where customer data is stored, Microsoft does not control or limit the locations from which customers or their end-users may access it. For more information, see [International availability of Dynamics 365](#).

### Upcoming changes to region availability

Dynamics 365 solutions consist of a collection of multiple services. Looking across Dynamics 365 applications, the Power Platform and the Azure services that they both depend on, the required matrix of services is quite large and growing. We have locked on a strategy of selecting a subset of data center regions across the globe to simplify ensuring that we have availability of the full portfolio of required services. Our plan is to optimize to have minimal latency between the component services of a solution and as a result, we are focused on having the full portfolio of services available in each of the designated data centers.

Additionally, the Finance and Operations architecture is being enhanced to build on self-service for greater elasticity, stronger reliability, and more seamless maintenance. Customers gain material efficiency by having deeper self-service deployments in fewer data centers. This transition also benefits from selecting a subset of Azure regions. To that effect, the regional availability of Finance and Operations apps will now be **limited to East US, West US, and Central US in North America** for all new projects. For a list of the latest supported regions, see [International availability of Dynamics 365](#).

Support for East US2, West US2, West Central US, North Central US, and South Central US will continue to be available for projects and environments that currently have their data stored in those regions on Microsoft-managed environments.

### NOTE

Microsoft will work with customers to move them to an appropriate data center beginning October 19, 2020. This will happen in a phased approach. Select customers will receive advance notification before we migrate them to a supported region.

If there are other customer workloads that are not part of the Dynamics 365 or Power Platform family that also require proximity to the Dynamics 365 and Power Platform services, Microsoft will work with customers to coordinate a plan for the overall migration. For more information, see [Cloud deployment overview: Frequently asked questions](#).

# Frequently asked questions

## **Why does the status display 'Maintenance' on my environment in LCS?**

To provide the best experience and performance, Microsoft performs maintenance operations on your environment. During some of these maintenance operations, your environment status may display one of the following statuses:

- Preparing for maintenance
- Prepared for maintenance
- Maintenance in progress

While your environment is in this state and until the status returns to 'Deployed', you will not be able to perform any lifecycle operations, such as package applications. There will be no impact to Finance and Operations apps. Users can continue with normal operations without any service interruption. You will receive an email notification before any maintenance operation puts your environment in this state.

## **How do I connect to the SQL database on my Sandbox environment?**

To connect to the SQL database in your Sandbox environment, follow the steps in [Enable just-in-time access](#).

## **How do I access a development instance?**

For information about how to access development instances, configure on-premises development VMs, and find configurations settings for developers and administrators, see [Deploy and access development environments](#).

## **How do I deploy a demo environment?**

A demo environment includes only Microsoft demo data. You can use a demo environment to explore default features and functionality. For more information, see [Deploy a demo environment](#).

## **How do I move my customizations between environments?**

To move customizations from a development to a sandbox or production environment, see [Create deployable packages of models](#)

## **Can I bring my own domain name?**

You can bring your own domain name if it is running Azure Active Directory (AAD), and the administrator of your AAD instance has enabled the Finance and Operations apps within their AAD. This is usually done through the office email, after you buy a license. When you click the link to accept the offer, AAD is set up for you.

## **Can I add guest AAD accounts as users?**

You can add guest AAD accounts if you have correctly configured them within Azure Active Directory, and enabled the Finance and Operations apps within your AAD.

## **Why am I no longer able to see the Private AOS machines in one or more of my Tier 2 through Tier 5 Sandbox environments?**

The Private AOS VMs were part of your environment configuration as they were needed to secure communication between the AOS and BI machines in the past. With recent updates, all communication between AOS and BI machines are secure directly and no longer need the intermediary Private AOS machines. Therefore, we are in the process of rolling out removing the Private AOS machines. As we are removing the machines in batches, you may notice that only some of your environments have the Private AOS machines removed. This change will not impact functionality or security in any way and will be transparent to you.

## **Why am I no longer able to Remote Desktop into one or more of my Tier 1 through Tier 5 Microsoft managed Sandbox environments?**

Microsoft managed Tier 1 through Tier 5 sandbox environments require Remote Desktop management endpoints to be restricted to specific IP Address sets (safe list). Microsoft regularly validates that the environments are sufficiently restricted. Microsoft reserves the right to immediately remove any IP Address safe list rules that violate the above guidelines without notice. You may not be able to Remote Desktop into your

environment for one of these reasons:

- Your current IP address is not in the safe list.
- Your IP has changed from the IP address listed in the safe list.
- Microsoft deleted the rule containing your IP address from the safe list because it violated a guideline.

To regain access to the environment, you will need to add the IP address of the computer from which you are connecting to. To do this, complete the steps [Remote Desktop](#) section earlier in this topic.

### **When will the availability of reduced regions go into effect for new onboarding?**

Beginning August 1, 2020, new projects for Finance and Operations will be onboarded to the following regions:

- East US
- West US
- Central US

### **My environments are currently in the regions that will be deprecated. How will this change affect me?**

We will deprecate support for the following regions only for new projects that will be onboarded on or after August 1, 2020:

- East US2
- West US2
- West Central US
- North Central US
- South Central US

This will not affect any environments that have their data stored in the deprecated regions before August 2020. In the near future there is a transition plan to move customers in the deprecated regions into the reduced regions.

### **I'm unable to redeploy an environment after deleting it, the environment slot is missing.**

This is due to the license expiring, which means that you no longer have the minimum required licenses to obtain an environment slot. Please review your [subscription status](#) and then reactivate the expired license to enable the redeployment.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# System requirements for cloud deployments of Dynamics 365 Commerce

2/18/2021 • 10 minutes to read • [Edit Online](#)

This topic lists the system requirements for cloud deployments of the current version of Dynamics 365 Commerce. If this step is appropriate, before you install Commerce, you should verify that the system that you're working with meets or exceeds the minimum network, hardware, and software requirements.

## Supported web browsers

The web application can run in any of the following web browsers that run on the specified operating systems:

- Microsoft Edge (latest publicly available version) on Windows 10
- Internet Explorer 11 on Windows 10, Windows 8.1, or Windows 7
- Google Chrome (latest publicly available version)
- Apple Safari (latest publicly available version)

### NOTE

It is possible for the Safari browser to show an error during device activation of a Cloud POS device due to an Azure Active Directory token being unattainable. You can resolve this issue by utilizing the [Microsoft Enterprise SSO plug-in for Apple devices](#).

To find the latest release for each web browser, go to the software manufacturer's website.

### NOTE

- To enable Task Recorder to capture screenshots and include them in Microsoft Word documents that are generated, you must install a pre-release Chrome extension.
- The Workflow Editor and Report Designer for Financial reporting are started as ClickOnce applications. They require a 64-bit-compatible operating system. Only Microsoft Edge and Internet Explorer (on a supported version of Microsoft Windows) support ClickOnce applications out of the box. If you're using Chrome, you must install a ClickOnce extension, such as [Meta4](#) to use ClickOnce applications. If you use Chrome in incognito mode, make sure that the ClickOnce extension is also enabled for incognito mode.
- To preview PDF files, we recommend that you use browsers such as Microsoft Edge (latest publicly available version) on Windows 10, or Google Chrome (latest publicly available version) on Windows 10, Windows 8.1, Windows 8, Windows 7, or Google Nexus 10 tablet.

## Supported web browsers for Cloud POS

Cloud point of sale (POS) can run in any of the following web browsers that run on the specified operating systems:

- Microsoft Edge (latest publicly available version) on Windows 10
- Internet Explorer 11 on Windows 10, Windows 8.1, or Windows 7

### NOTE

Beginning with release 10.0.17, Internet Explorer will no longer be supported.

- Chrome (latest publicly available version) on Windows 10, Windows 8.1, or Windows 7

## Network requirements

- Commerce is designed for networks that have a latency of 250–300 milliseconds (ms) or less. This latency is the latency from a browser client to the Microsoft Azure datacenter that hosts Commerce. We recommend that you test network latency at [AzureSpeed.com](https://azure.microsoft.com/en-us/learn/azure-speed/).
- Bandwidth requirements for Commerce depend on your scenario. Most typical scenarios require a bandwidth that is more than 50 kilobytes per second (KBps). However, we recommend more bandwidth for scenarios that have high payload requirements, such as scenarios that involve workspaces or extensive customization.

In general, Commerce is optimized for the internet. The number of round trips from a browser client to the Azure datacenter is very small, and the whole payload is compressed.

### WARNING

Don't calculate bandwidth requirements from a client location by multiplying the number of users by the minimum bandwidth requirements. The concurrent usage of a given location is very difficult to calculate. Customers who are concerned about bandwidth requirements should use a preview version of Commerce.

## .NET Framework requirements

Commerce requires Microsoft .NET Framework 4.7.1 or later for all ClickOnce applications, such as the document routing agent. For installation instructions, see [Install the .NET Framework for developers](#).

## Supported Microsoft Office applications

The following Microsoft Office applications are supported:

- To run the Microsoft Excel and Word add-ins, you must have Microsoft Office 2016 for Windows installed. For more information about version requirements, see [Troubleshoot the Office integration](#).
- To view documents that are generated by the Export to Excel or Export to Word functionality, you must have Microsoft Office 2007 or later installed.

## System requirements for Commerce client components

It is critical to perform proper performance testing prior to going live in production. The following are considered minimum system requirements for applications to function. To achieve desired performance, consider concepts like data volumes, transactional load per hour, and customization impact. Proper performance testing both early into implementation and again prior to final testing will allow for any necessary performance improvements to be made and to validate that the base solution meets the expected operation times required.

[!WARNING] The Microsoft Windows 7 operating system is no longer supported for anything other than security-related fixes. As a result, while Dynamics 365 Commerce components may function on Windows 7, there will be no bug fixes that specifically relate to supporting this operating system. Workarounds may be required for components to function properly on Windows 7, so it is highly recommended to upgrade to a supported operating system.

## Modern POS for Windows requirements

#### NOTE

- If Modern POS will use an offline database, the computer must meet all system requirements for Microsoft SQL Server and the system must have no less than 10 GB of disk space available. It is recommended to have no less than 20 GB of disk space available. An offline database for Modern POS requires SQL Server 2014 with Service Pack 3 or later, SQL Server 2016 with Service Pack 2, SQL Server 2017, or newer. The SQL Server version used must have the Full-Text Search feature installed. We recommend that you always use the latest version that is available, and that you install all the latest service packs. By following these recommendations, you can help to ensure both compatibility and security.
- Starting August 1, 2019, Modern POS and other client-side components require that the Microsoft .NET Framework version 4.7.1 or later be installed. For installation instructions, see [Install the .NET Framework for developers](#).

### Supported Windows operating systems

- Modern POS is a 32-bit application, but it will run on both x86 and x64 architectures.
- Modern POS is supported on Windows Server 2016, Windows 10 Pro, Windows 10 Enterprise, Windows 10 Enterprise Long Term Service Branch (LTSB), and Windows 10 IOT Enterprise editions. At a minimum, the Windows 10 Anniversary Update (version 1607), build 14393, must be installed.
- It is not recommended to use Modern POS and other Commerce components on Windows 10 Pro unless within a domain as Windows 10 Pro doesn't allow for advanced management of updates to the operating system.
- It is not recommended to use Modern POS and any other Commerce component together on the same computer.

### Minimum system requirements

- The minimum supported effective resolution for POS Full layout (PCs and tablets) is 1,024 × 768 (recommended 1366 x 768 or greater)
- The minimum supported effective resolution for POS Compact layout (phones and small tablets) is 320 x 568 (recommended 360x640 or greater)
- The computer that Modern POS runs on must meet these requirements:
  - It must have, at a minimum, a dual-core processor that runs at no less than 2 gigahertz (GHz).
  - It must have, at a minimum, 3 gigabytes (GB) of random-access memory (RAM). When combining with SQL Server for offline, no less than 4 GB of RAM is required.
  - It must have internet access.

## Modern POS for Apple iPhone or iPad requirements

- iOS 11 or later

## Modern POS for Android phone or tablet requirements

- Android OS 6.0 or later

## Retail hardware station requirements

#### NOTE

Starting August 1, 2019, Retail hardware station and other client-side components require that the .NET Framework version 4.7.1 or later be installed. For installation instructions, see [Install the .NET Framework for developers](#). It is critical to note that this component utilizes a server certificate. Server certificates must be managed for expiration. By default, a certificate expires in one calendar year (365 days).

## Supported operating systems

- Retail hardware station is a 32-bit application, but it will run on both x86 and x64 architectures.
- Retail hardware station is supported on the following operating systems:
  - Windows 8.1 Update 1 Professional, Enterprise, and Embedded editions.
  - Windows 10 Pro, Enterprise, Enterprise LTSB, and IOT Enterprise editions.
  - Windows Server 2016 and Windows Server 2019.
  - It is not recommended to use Retail hardware station and other Commerce components on Windows 10 Pro unless within a domain as Windows 10 Pro doesn't allow for advanced management of updates to the operating system.

## Minimum system requirements

The computer must meet all system requirements for installing and using the following items:

- Microsoft Internet Information Services (IIS)
- Third-party hardware

## Commerce Scale Unit (self-hosted) requirements

### NOTE

Starting August 1, 2019, Commerce Scale Unit and other client-side components require that the .NET Framework version 4.7.1 or later be installed. For installation instructions, see [Install the .NET Framework for developers](#).

It is critical to note that this component utilizes a server certificate in addition to Azure Service to Service authentication. Both the generated Azure web application keys (formerly called *secrets*) and the server certificate must be managed for expiration. By default, a certificate and a generated Azure web application key expires in one calendar year (365 days).

Take note that the minimum system requirements listed below are the bare minimum necessary to get a Commerce Scale Unit to function in a test scenario. The following is not representative of a realistic production environment. It is critical to perform proper performance testing and validate that the hardware used will meet the needs of the users.

## Supported operating systems

- Commerce Scale Unit is a 32-bit application, but it will run on both x86 and x64 architectures.
- Commerce Scale Unit is supported on the following operating systems:
  - Windows 8.1 Update 1 Professional, Enterprise, and Embedded editions.
  - Windows 10 Pro, Enterprise, Enterprise LTSB, and IOT Enterprise editions.
  - Windows Server 2016 and Windows Server 2019.
  - It is not recommended to use Commerce Scale Unit and other Commerce components on Windows 10 Pro unless within a domain as Windows 10 Pro doesn't allow for advanced management of updates to the operating system.

## Minimum system requirements

### NOTE

The following are the minimum system requirements for Commerce Scale Unit. Both these and the recommended requirements are the minimum possible for testing and basic functionality. It is crucial to perform performance testing and validate that the hardware used for Commerce Scale Unit meets expectations.

- 4 GB of RAM



- 1.6 GHz i5 (or equivalent) minimum CPU speed per core (2 cores are the minimum).
- At least 15 GB of free space (the channel database can require a large amount of space).

### **Recommended system requirements**

- 6 GB of RAM
- 2.4 GHz i7 (or equivalent) minimum CPU speed per core (4 cores are recommended).
- At least 20 GB of free space (the channel database can require a large amount of space).

It would be in an organization's best interest to also take the following items into consideration when determining personal hardware needs:

- Number of physical network ports (more ports enhances throughput per second).
- SQL Server log flush size (this directly impacts SQL Server performance).
- Data read and write capabilities (this directly impacts SQL Server performance).
- Number of CPU(s) core, number of simultaneous threads per core, and speed per core (this impacts overall throughput of the system).
- Whether load balancing will be required.

## Connector requirements

### **Supported operating systems**

- The Connector for Microsoft Dynamics AX has two separate installers, one for Async Server Connector service and one for Real-time service for Microsoft Dynamics AX 2012 R3.
- Both components are 32-bit applications, but they will run on both x86 and x64 architectures.
- Both components are supported on the following operating systems:
  - Windows 8.1 Update 1 Professional, Enterprise, and Embedded editions.
  - Windows 10 Pro, Enterprise, and Enterprise LTSB editions.
  - Windows Server 2016 and Windows Server 2019.
  - It is not recommended to use Commerce components on Windows 10 Pro unless within a domain as Windows 10 Pro doesn't allow for advanced management of updates to the operating system.

### **Minimum system requirements**

- 2 GB of RAM (4 GB of RAM are recommended).
- 1.6 GHz peak CPU speed per core (2 cores are the minimum).
- At least 10 GB of free space (the channel database can require a large amount of space).

## Requirements for development on local VMs

For information about the requirements for development on local virtual machines (VMs), see [VM that is running on-premises](#).

## Database collation

The only supported collation for Commerce databases in the cloud is `SQL_Latin1_General_CP1_CI_AS`. Please ensure that your SQL Server and database collations in development environments are set to this. Also ensure that any configuration environments that are published to Sandbox have this same collation.

## Additional resources

[Get an evaluation copy](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# On-premises deployment home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can deploy Dynamics 365 Finance + Operations (on-premises). When you choose an on-premises deployment type, the system requirements, hardware sizing, and functionality differ from a cloud deployment. This topic provides links to content that contains information specific to on-premises deployments.

## Get started

- [On-premises deployment overview](#)
- [Plan and prepare for on-premises deployments](#)
- [System requirements for on-premises deployments](#)
- [Hardware sizing requirements for on-premises environments](#)
- [Buy Finance + Operations \(on-premises\)](#)
- [Comparison of cloud and on-premises features](#)

## Onboard

- [Set up on-premises projects in Lifecycle Services \(LCS\)](#)
- [Set up and deploy on-premises environments \(Platform update 12 and later\)](#)
- [Install network printer devices in on-premises environments](#)
- [Configure SQL Server Reporting Services for on-premises deployments](#)
- [Develop and deploy custom models to on-premises environments](#)

## Work in your on-premises deployment

- [Configure document management](#)
- [Import Electronic reporting \(ER\) configurations](#)
- [Document generation, publishing, and printing in on-premises deployments](#)
- [Configure proxies for on-premises environments](#)
- [Set up technical support for Finance and Operations apps](#)
- [Client internet connectivity](#)
- [Apply updates to on-premises deployments](#)
- [Redeploy on-premises environments](#)
- [Reuse the same AD FS instance for multiple environments](#)

## Commerce

- [Commerce capabilities that are available in on-premises deployments](#)
- [Installation steps for Retail channel components in an on-premises environment](#)
- [Configure, install, and activate Modern POS \(MPOS\)](#)
- [Configure and install Commerce Scale Unit](#)

## Upgrade

- [In-place upgrade process for on-premises environments](#)

## Other resources

- [Troubleshoot on-premises deployments](#)
- [Scripts for resolving issues in on-premises environments](#)
- [Certificate rotation](#)
- [On-premises diagnostics](#)
- [Removed or deprecated features for Finance and Operations](#)
- [Software lifecycle policy and on-premises releases](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# System requirements for on-premises deployments

2/18/2021 • 10 minutes to read • [Edit Online](#)

This topic lists the system requirements for the current version of Microsoft Dynamics 365 Finance + Operations (on-premises) deployments. Before you install, when this step is appropriate, verify that the system that you're working with meets or exceeds the minimum network, hardware, and software requirements.

## IMPORTANT

Dynamics 365 Finance + Operations (on-premises) deployments are not supported on any public cloud infrastructure, including Azure.

## Network requirements

Dynamics 365 Finance + Operations (on-premises) can work on networks that use Internet Protocol Version 4 (IPv4) or Internet Protocol Version 6 (IPv6). Consider the network environment when you plan your system, and use the following guidelines.

### Network response time

The following table lists the minimum network requirements for the connection between the web browser and Application Object Server (AOS), and for the connection between AOS and the database in an on-premises system.

VALUE	WEB BROWSER TO AOS	AOS TO DATABASE
Bandwidth	50 kilobytes per second (KBps) per user	100 megabytes per second (MBps)
Latency	Less than 250–300 milliseconds (ms)	Less than 1 ms (local area network [LAN] only). AOS and the database must be co-located.

- Finance + Operations is designed for networks that have a latency of 250–300 milliseconds (ms) or less. This latency is the latency from a browser client to the datacenter that hosts Finance + Operations.
- Bandwidth requirements depend on your scenario. Typical scenarios require a bandwidth of more than 50 KBps between the browser and the server. However, we recommend higher bandwidth for scenarios that have high payload requirements, such as scenarios that involve workspaces or extensive customization. The specific amount of bandwidth depends on use.

Deployments where AOS and the Microsoft SQL Server database are in different datacenters aren't supported. AOS and the SQL Server database must be co-located.

In general, Finance + Operations is optimized to reduce browser-to-server round trips. The number of round trips from a browser client to the datacenter is either zero or one for each user interaction, and the payload is compressed.

## WARNING

Don't calculate bandwidth requirements from a client location by multiplying the number of users by the minimum bandwidth requirements. The concurrent usage of a given location is very difficult to calculate. We recommend that you use a real-life simulation against a non-production environment as the best gauge of performance for your specific case.

### LAN environments

In LAN environments, Microsoft Remote Desktop in Microsoft Windows Server isn't required in order to connect to Finance + Operations. However, Remote Desktop might be required for servicing operations on the virtual machines (VMs) that make up the server deployments.

### WAN environments

In wide area network (WAN) environments, Remote Desktop in Windows Server isn't required in order to connect to Finance + Operations.

### Internet connectivity requirements

Finance + Operations doesn't require internet connectivity from user workstations. However, some features won't be available if there is no internet connectivity.

Browser client	An intranet scenario without internet connectivity is a design point for the on-premises deployment option. Some features that require cloud services won't be available, such as Help and Task guide libraries in Microsoft Dynamics Lifecycle Services (LCS).
Server	The AOS or Microsoft Azure Service Fabric tier must be able to communicate with LCS. The on-premises browser-based client doesn't require internet access.
Telemetry	Telemetry data might be lost if there are long interruptions in connectivity. Interruptions in connectivity to LCS don't affect the on-premises application functionality.
LCS	Connectivity to LCS is required for deployment, code deployment, and servicing operations.

## Telemetry data transfer to the cloud

Most telemetry data is stored locally and can be accessed by using Event Viewer in Microsoft Windows. A small subset of telemetry events is transferred to the Microsoft telemetry pipeline in the cloud for diagnostics. Customer data and user-identifiable data aren't part of the telemetry data that is sent to Microsoft. VM names are sent to Microsoft to help with environment management and diagnostics from the LCS portal.

## Domain requirements

Consider the following domain requirements when you install Finance + Operations:

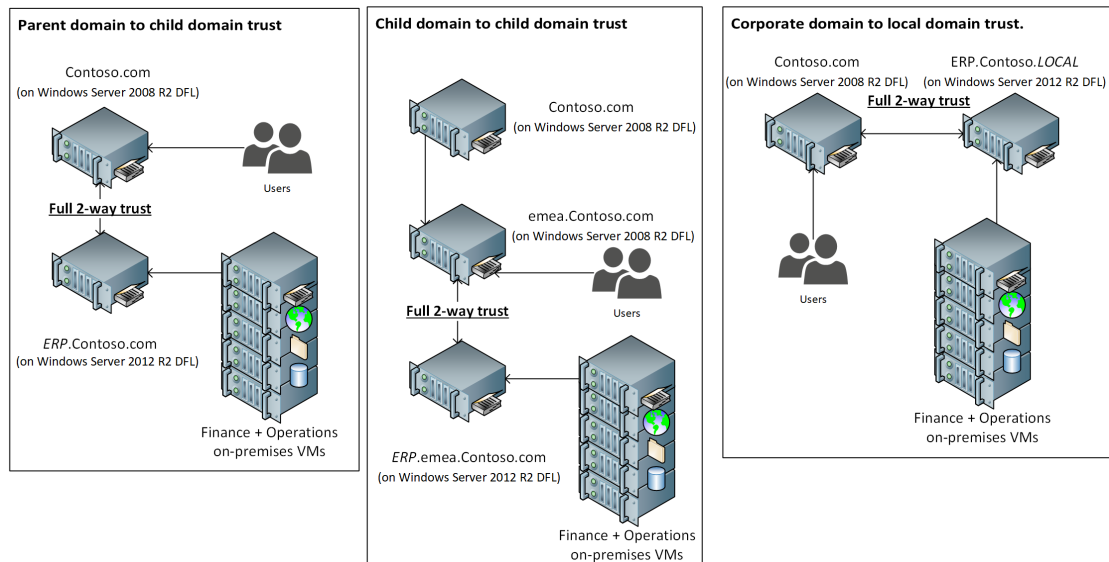
- VMs that host Finance + Operations components must belong to an Active Directory domain. Active Directory Domain Services (AD DS) must be configured in native mode.
- VMs that run Finance + Operations components must have access to each other. This access is configured in AD DS.
- The domain controller must be Microsoft Windows Server 2012 R2 or later, and the domain functional level must be 2012 R2 or more.

## Full 2-way trust

For compatibility with corporate domain controllers on Windows Server 2008 R2 domain functional level (DFL), a full 2-way trust between the Windows Server 2008 R2 DFL user domain and the Windows Server 2012 R2 DFL Finance + Operations service domain is supported in Platform update 33 and later.

This means that users of the Finance + Operations (on-premises) application will come from the Windows Server 2008 R2 DFL domain, and the resources and service accounts hosting the Finance + Operations (on-premises) infrastructure and services will come from the Windows Server 2012 R2 DFL domain.

Examples for a full 2-way trust setup could be.



### Known limitations with using the full 2-way trust setup

- Import of security groups from the Windows Server 2008 R2 user domain is not supported.

## Hardware requirements

This section describes the hardware that is required in order to run Finance + Operations.

The actual hardware requirements vary, based on the system configuration, the data composition, and the features that you decide to use. Here are some of the factors that can affect the choice of appropriate hardware:

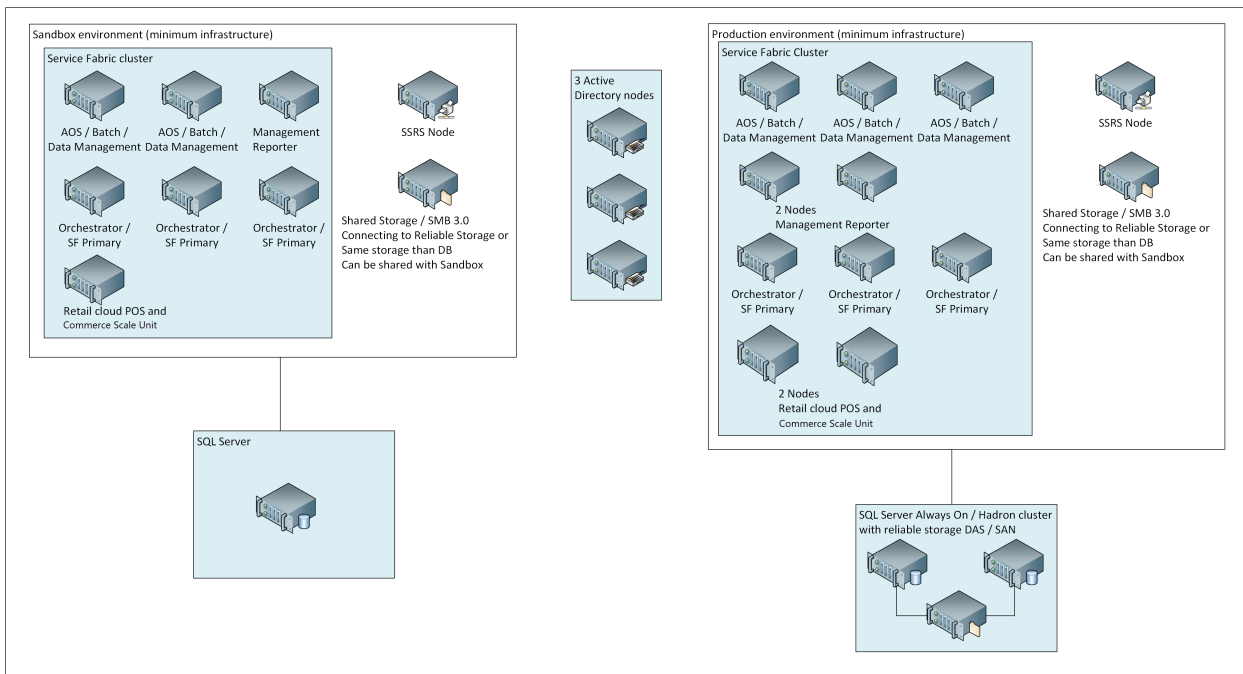
- The number of transactions per hour
- The number of concurrent users

## Minimum infrastructure requirements

Finance + Operations uses Service Fabric to host the AOS, Batch, Data management, Management reporter, and Environment orchestrator services.

SQL Server must have a high-availability HADRON setup that has at least two nodes for production use.

The following illustration shows the minimum number of nodes that is recommended for your Service Fabric cluster.



## Processor and RAM requirements

The following tables list the number of processors and the amount of random-access memory (RAM) that are required for each role that is required in order to run this deployment option. For more information, see the recommended minimum requirements for a Service Fabric standalone cluster in [Plan and prepare your Service Fabric cluster](#).

### NOTE

If other Microsoft software is installed on the same computer, the system must also comply with the hardware requirements for that software. If other server applications are installed on the same computer as AOS, we recommend that you limit those server applications 1 gigabyte (GB) of RAM.

### Sizing by role and topology type

TOPOLOGY	ROLE (NODE TYPE)	RECOMMENDED PROCESSOR CORES	RECOMMENDED MEMORY (GB)
Production	AOS, Data management, Batch	8	24
	Management Reporter	4	16
	SQL Server Reporting Services	4	16
	Orchestrator	4	16
	SQL Server	8	32
Sandbox	AOS, Data management, Batch	4	24
	Management Reporter	4	16



TOPOLOGY	ROLE (NODE TYPE)	RECOMMENDED PROCESSOR CORES	RECOMMENDED MEMORY (GB)
	SQL Server Reporting Services	4	16
	Orchestrator	4	16
	SQL Server	8	32

### Minimum sizing estimates for production and sandbox deployments

TOPOLOGY	ROLE	NUMBER OF INSTANCES
Production	AOS (Data management, Batch)	3
	Management Reporter	2
	SQL Server Reporting Services	1
	Orchestrator**	3
	SQL Server	2
Sandbox	AOS, Data management, Batch	2
	Management Reporter	1
	SQL Server Reporting Services	1
	Orchestrator	3
	SQL Server	1
<i>Summary for production and sandbox topologies</i>		<i>19</i>

\* The numbers in this table are being validated by our preview customers and might be adjusted based on the feedback from those customers.

\*\* Orchestrator is designated as the primary node type and will also be used to run the Service Fabric services.

### Initial estimates for the back-end SQL Server and AD DS

	ROLE	VMS/INSTANCES	CORES	TOTAL CORES	MEMORY PER INSTANCE (GB)	TOTAL MEMORY (GB)
Shared infrastructure	SQL Server*	2	8	16	32	64

	ROLE	VMS/INSTANCES	CORES	TOTAL CORES	MEMORY PER INSTANCE (GB)	TOTAL MEMORY (GB)
	File server/Storage area network/Highly available storage	The back-end storage must be based on solid-state drives (SSDs) on a runtime storage area network (SAN). Size and input/output operations per second (IOPS) throughput is based on the size of the workload.				
	Active Directory	3	4	12	16	48
<i>Summary for shared infrastructure</i>		5		28		112

\* SQL Server sizes are highly dependent on workloads. For more information, see [Hardware sizing requirements for on-premises environments](#). Separate SQL Server machines for sandbox and production environments must be used. However, SQL Server can be shared in all sandbox environments.

## Storage

- AOS – Finance + Operations uses a Server Message Block (SMB) 3.0 share to store unstructured data. For more information, see [Storage Spaces Direct in Windows Server 2016](#).
- SQL – The following options are viable:
  - A highly available SSD setup
  - A SAN that is optimized for online transaction processing (OLTP) throughputs
  - High-performance direct-attached storage (DAS)
- SQL Server and data management IOPS – The storage for both data management and SQL Server should have at least 2,000 IOPS. Production IOPS depends on many factors. For more information, see [Hardware sizing requirements for on-premises environments](#).
- VM IOPS – Each VM should have at least 100 write IOPS.

## Virtual host requirements

When you set up the virtual hosts for an environment, see the guidelines in [Plan and prepare your Service Fabric cluster](#) and [Describing a service fabric cluster](#). Each virtual host should have enough cores for the infrastructure that is being sized. Multiple advanced configurations are possible, where SQL Server resides on physical hardware but everything else is virtualized. If SQL Server is virtualized, the disk subsystem should be a fast SAN or the equivalent. In all cases, make sure that the basic setup of the virtual host is highly available and redundant. In all cases, when virtualization is used, no VM snapshots should be taken.

Finance + Operations falls under Microsoft's standard support policy regarding operation on non-Microsoft virtualization platforms – specifically VMWare. For more information, read [Support policy for Microsoft software](#). In short, we support our products in this environment, but if we are asked to investigate an issue, we may ask the customer to first reproduce the problem without the virtualization platform or on the Microsoft virtualization platform.

## Software requirements for all server computers

The following software must be present on a computer before any Finance + Operations components can be

installed:

- The Microsoft .NET Framework. See [Deployment setup](#) for version information.
- Service Fabric

For more information, see [Plan and prepare your Service Fabric cluster](#).

## Supported server operating systems

The following table lists the server operating systems that are supported.

OPERATING SYSTEM	NOTES
Microsoft Windows Server 2016 Datacenter or Standard	These requirements are for the database and the Service Fabric cluster that hosts AOS. Only en-US OS installations are supported.

## Software requirements for database servers

- Only 64-bit versions of SQL Server 2016 are supported.
- Only `SQL_Latin1_General_CP1_CI_AS` is valid for the server and database collation. For more information about how to select a collation for a SQL Server database, see the [SQL Server documentation](#).
- In a production environment, we recommend that you install the latest cumulative update (CU) for the version of SQL Server that you're using.

The following table lists the SQL Server versions that are supported for the databases. For more information, see the minimum hardware requirements for [SQL Server](#).

REQUIREMENT	NOTES
Microsoft SQL Server 2016 Standard Edition or Enterprise Edition	For the hardware requirements for SQL Server 2016, see <a href="#">Hardware and Software Requirements for Installing SQL Server 2016</a> .

## Software requirements for Application Object Server (AOS)

- SQL Server Integration Services (SSIS)

## Software requirements for Reporting Server (BI)

- SQL Server Reporting Services (SSRS)

## Software requirements for client computers

Users can access Finance + Operations by using the most recent versions of these popular browsers:

- Microsoft Edge (recommended: [Chromium-based Edge](#))
- Google Chrome
- Apple Safari
- Internet Explorer 11 (deprecated, not recommended)

#### NOTE

For optimal performance and an optimal experience, we recommend that you use the latest version of a modern browser, especially Microsoft Edge. Support for Internet Explorer 11 is deprecated. For more information, see the [Internet Explorer deprecation announcement](#).

## Software requirements for Active Directory Federation Services

Active Directory Federation Services (AD FS) on Windows Server 2016 is required.

The domain controller must be Windows Server 2012 R2 or later, and the domain functional level must be 2012 R2 or more. For more information about domain functional levels, see the following pages:

- [What Are Active Directory Functional Levels](#)
- [Understanding Active Directory Domain Services Functional Levels](#)
- [Full 2-way trust](#)

## Supported Microsoft Office applications

The following Microsoft Office applications are supported in on-premises deployments:

- To run the Microsoft Excel and Microsoft Word add-ins, you must have Microsoft Office 2016 for Windows (or newer) installed. For more information about version requirements, see [Troubleshoot the Office integration](#).
- To view documents that are generated by the Export to Excel or Export to Word functionality, you must have Microsoft Office 2007 or later installed.

## Hardware and software requirements for Commerce components

Currently, Finance + Operations doesn't include the Commerce components.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Installation steps for Retail channel components in an on-premises environment

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic covers the installation steps for Commerce channel components in an on-premises environment.

## Overview

Channel functionality, in an on-premises environment, is enabled exclusively via use of Commerce Scale Unit (self-hosted). For an overview, see [Commerce Scale Unit \(self-hosted\)](#).

Unlike a cloud deployment, an on-premises environment does not enable seamless, high-availability deployment of channel components via Lifecycle Services (LCS). The only way to use channel components is by installing Commerce Scale Unit (self-hosted).

## Prerequisites

Before you can start installation of channel components, you must first complete all prior installation steps for an on-premises environment. These steps are listed in [Set up and deploy on-premises environments \(Platform update 12 and later\)](#). In addition, version 8.1.1 must be installed in order for Commerce have full functionality. We recommend that you update to version 8.1.2.

### NOTE

It is critical to ensure that a secure network, that is not publicly accessible, is used to connect Commerce Scale Unit to Headquarters. You must also restrict network access to Headquarters, so access is only allowed to known Commerce Scale Unit devices via network filtering or other means. This means that a firewall must exist and using a safe list is highly recommended.

## Installation steps

1. On the previously created [Application share](#), (not the **LocalAgent** share folder), create a new folder called **selfservicepackages** in the root directory of the share location.
2. On each AOS computer, create an easily accessible directory, such as **C:/selfservicepackages**.
3. On one AOS computer (which one does not matter), run the following PowerShell script.

```
.\RetailUpdateDatabase.ps1 -envName '<Environment name>' -AosUrl 'https://<My Environment Name>.com/namespaces/AXSF/' -SendProductSupportTelemetryToMicrosoft
```

## IMPORTANT

The above steps apply to version 10.0 and later. For the original 8.1.3 release of Retail on-premises functionality, the original version of the script delimiters must be used.

```
.\RetailUpdateDatabase.ps1 -DatabaseServer '<Database server name for AOS database>' -  
DatabaseName '<Database name for AOS database>' -envName '<Environment name>' -  
RetailSelfServicePackages '<Local path of Retail self-service packages, such as  
**C:/selfservicepackages**>' -SendProductSupportTelemetryToMicrosoft
```

- The parameter **-envName** should be known based on creation when the environment is generated.
- The legacy parameters **-DatabaseServer** and **-DatabaseName** should be known based on the environment setup.
- The parameter **-SendProductSupportTelemetryToMicrosoft** is a required value to enable telemetry to Microsoft. This is critical to maximize support from Microsoft.
- This script will perform a variety of actions, including updating the Service user and role and updating registry keys.

4. On each AOS computer, run the following PowerShell script.

```
.\RetailUpdateDatabase.ps1 -RetailSelfServicePackages 'C:\RetailSelfService\Packages'
```

## NOTE

The parameter **-RetailSelfServicePackages** is the full path location created in the beginning of this step (C:/selfservicepackages).

5. Download the appropriate binary update from LCS to have the Commerce installers. For instructions, see [Download updates from Lifecycle Services \(LCS\)](#).
6. Extract the zip file and copy all self-service installers into the folder **C:/selfservicepackages** defined and created in step 2 in each of the AOS machines. The six self-service installers include:
- AsyncServerConnectorServiceSetup.exe
  - RealtimeServiceAX63Setup.exe
  - HardwareStationSetup.exe
  - ModernPosSetup.exe
  - ModernPosSetupOffline.exe
  - StoreSystemSetup.exe

## NOTE

Cloud environments can synchronize self-service installers through Headquarters from what is available in LCS ([Synchronize self-service installers in Dynamics 365 Commerce](#)). On-premises environments cannot utilize this functionality, however, these environments can still download from LCS. The SDK is available in the deployable package zip file. The self-service installers are available from the **LCS Asset library**. You can utilize the upload and download mechanism from within LCS, but the Headquarters synchronization functionality will not work.

7. Navigate to the AD FS machine, then go to the InfrastructureScripts folder. This is the same file directory where the previously run PowerShell script was located (**RetailUpdateDatabase.ps1**). Find the PowerShell script **Create-ADFSServerApplicationForRetail.ps1**.
8. On the AD FS machine that you're currently using, run this script in a new PowerShell window using the

command `.\Create-ADFSServerApplicationForRetail -HostUrl 'https://ax.d365ffo.onprem.contoso.com'`, where the `HostUrl` value can be found in Service Fabric. To find the `HostUrl` value, go to **Service Fabric > Application fabric:/AXSF > Details > Aad\_AADValidAudience**.

9. Access the newly generated Server application from the **Application Groups** in AD FS Management.
10. Edit the newly generated Server application and select **Reset the Secret**.

#### NOTE

It is an important security measure to run this script for each Commerce Scale Unit. This maximizes security and minimizes the workload in case of a security breach.

It is critical to keep this secret safe. This secret should only be copied once and never stored on the system. The Client ID and Secret generated will be used during the Commerce Scale Unit installer, so it is required to be used at a later time. You can always reset the secret again, but it must then be updated on any Commerce Scale Unit that used the previous secret.

11. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Connector for Microsoft Dynamics AX**.
12. Select **Edit** on the Action pane.
13. In the **Profile** field, enter the value **Default**. If needed, enter a description in the **Description** field.

#### NOTE

It is possible for the following fields in steps 12 through 14 to already have values. If this occurs, skip those steps and continue from there. What is important is to have a selectable profile title (default in this case).

14. In the **Web application name** field, enter **RetailCDXRealTimeService**.
15. In the **Protocol** field, select **https**.
16. In the **Common name** field, enter **AXServiceUser@contoso.com**.
17. Select **Save** on the Action Pane.
18. In Headquarters, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
19. Select the **Security** tab.
20. Under the sub-heading **Transaction service legacy properties**, select the **Real-time Service profile** field, and then select the newly created **Default** value.
21. Select the **Identity providers** tab.
22. On the **Identity providers** FastTab, select **Add**.
23. In the new **Issuer** row, enter the new Identity provider value <https://sts.windows.net/> in the field.
24. Select **Save** on the Action Pane.
25. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**.
26. On the **General** tab, select the **Initialize** link to configure seed data for Commerce functionality.

#### NOTE

The installers will not download from their relevant pages the first time a download is attempted. This is because the installers have only just been placed into the download location and the associated database values do not yet exist. In Headquarters, when the **Download** functionality is attempted (for example, Commerce Scale Unit or Modern POS), an error will display and then an automated upload functionality will be initiated to allow the installers to be downloaded the second time that the download is attempted. (Wait one minute before attempting to download the installer again).

The Peripheral Simulator (downloaded on the Hardware profile page in headquarters) will not be available until at least one Hardware profile has been created and is functional. After that point has been achieved, the following script can be run.

```
.\RetailUpdateDatabase.ps1 -envName 'LBDenv1' -UpdateRetailHardwareProfileSelfServicePackage
```

27. Follow the installation steps for installing the Commerce Scale Unit. For instructions, see [Configure and install Commerce Scale Unit \(self-hosted\)](#). At multiple locations in this document there will be notes referencing changes to the instructions for an on-premises deployment. It is important to note each of these changes.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Buy Finance + Operations (on-premises)

2/18/2021 • 3 minutes to read • [Edit Online](#)

Microsoft Dynamics 365 Finance + Operations (on-premises) offers a variety of license types to best suit the needs of your organization. To better understand how Finance + Operations (on-premises) is licensed, please work with your partner, who can access the [Licensing guide](#) on PartnerSource. When you are ready to purchase licenses for your organization, work with your partner to follow the steps outlined in this topic.

## IMPORTANT

On-premises environments are not supported on any public cloud infrastructure, including Azure.

## Purchase client access licenses

To run on-premises environments, you must obtain the proper number of client access licenses (CALs) for your organization per the licensing guide. The CAL purchased for an individual user determines the functionality that the user has the rights to use. User CALs can be purchased from the [Microsoft Volume Licensing Service Center](#).

## Purchase server licenses

A server license is required for every server running Finance + Operations (on-premises). After purchasing your server licenses, work with your partner to download a license file from the [PartnerSource Business Center](#). Keep this license file handy, as the details it contains will be used when setting up your Lifecycle Services (LCS) project.

Partners can download a customer's Finance + Operations (on-premises) license file from the PartnerSource Business Center using these steps:

1. Log on to the [PartnerSource Business Center](#).
2. Enter the customer name or account number in the **Find A Customer** field, and then click **Search**.
3. Click the company name of the customer. This opens the **Customer Summary** page.
4. Under **Registered Products**, click **Registration Keys**.
5. Select **version 07** in the **Request and Display License Keys For Version** field.
6. Click **Display License Keys**.
7. On the **Request License Keys** page, select **Download Current License/Registration Key**.
8. Click **Save As** in the **File Download** dialog box, select the folder where you want to download the license file to in the **Save As** dialog box, and then click **Save**.

## NOTE

If you cannot see registration keys in PartnerSource Business Center, you will need to ensure that your PartnerSource Business Center Profile has **Can See Registration Keys** set to **Yes**.

## Get started with Lifecycle Services (LCS)

To purchase Finance + Operations (on-premises) you must have a Microsoft Online Services ID. The Microsoft Online Services ID is used to provision an LCS project that contains the necessary artifacts for on-premises environments. LCS is the service where on-premises environments will be provisioned, Business Process Model

created/uploaded, hotfixes are made available, support cases are entered and managed, license key serial number activation submitted, etc.

The Microsoft Online Services ID is required to provision and register Finance + Operations (on-premises) into entity-owned hardware and environment. See the Provisioning guide (linked to below) to complete the provisioning and registration process. If a Microsoft Online Services ID already exists, the process must be completed by the Global Administrator. If creating a Microsoft Online Services ID for the first time, the person initiating the process will be the Global Administrator.

If you have an existing Microsoft Online Services trial or paid subscription, you already have a Microsoft Online Services ID that was created at the time of sign-up. When you click a link below, choose to sign in with this account if you want to use this same Azure Active Directory (AAD) tenant for the on-premises environment.

Access to the Provisioning guide can be found here:

- [Provisioning guide on CustomerSource](#)
- [Provisioning guide on PartnerSource](#)

After you have logged into LCS, a project will be automatically provisioned for you. The LCS project will allow you to deploy an on-premises environment. For more details on getting started with your LCS project, see [Set up on-premises projects in Lifecycle Services \(LCS\)](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce Data Exchange implementation guidance

2/18/2021 • 16 minutes to read • [Edit Online](#)

This topic is intended for people who implement functionality that is related to data synchronization (Commerce Data Exchange, [CDX]) in a Microsoft Dynamics 365 Commerce environment. It gives an overview, implementation tips, and overall guidance that you should consider as you plan your implementation, in regard to pages, setup, configuration, best practices, and more.

## Overview

Proper configuration and synchronization of data is crucial to a correct implementation. Regardless of business requirements, IT infrastructure, and overall preparedness, if data isn't correctly synchronized, the whole environment is effectively useless. Therefore, a top priority is to understand what is required to configure, generate, synchronize, and verify data across the full implementation. This goes from Commerce headquarters through the Commerce Scale Unit to the brick-and-mortar stores that use Modern POS (With or without an offline database) and other in-store components. CDX is the Commerce functionality that replicates and synchronizes data across databases. However, CDX differs from typical data replication functionality because it also allows for filtering. Therefore, CDX helps minimize data sets by generating only data that is specific to the channels that were specified for selection, filtering specific tables from offline databases, and filtering expired records for data that is no longer used, such as expired discounts.

Before you go through this topic, it's important that you understand the concepts of a channel (store), registers and devices, and the Modern POS offline database. Therefore, we recommend that you review some of the resources at the end of this topic, such as the Device management implementation guide and the overview of the Commerce architecture.

### Important Commerce headquarters pages

- **Channel database** – Use this page to create, review, and edit the channel databases that are used in Commerce Scale Units (both Cloud and Self-hosted) and the offline databases that are used with Modern POS. Each database that you create here refers to a single, physical database (in other words, there is a one-to-one [1:1] mapping). A channel database or offline database must be associated with a channel database group. From this page, you can also create full synchronizations of a scheduler job for a selected channel database or offline database.
- **Channel database group** – Use this page to create, review, and edit channel database groups. Each group is associated with one or more channel or offline databases. The database group is responsible for gathering all the relevant data that is required by all the associated channel and offline databases, and that must be generated as part of CDX data synchronization.
- **Channel profile** – Use this page to create, review, and edit channel profiles. Each channel profile stores the URLs that are relevant to the network-based communication that is required within a channel. A channel profile typically has a Retail Server URL and a Cloud POS URL. Often, there is also a Media Server Base URL. This URL is the internet addressable location of images that are used by the POS, E-Commerce, and other Commerce channels. Although a channel profile is automatically generated for a Commerce Scale Unit (Cloud), it must be manually generated as part of the configuration and installation steps for a Commerce Scale Unit (Self-hosted).
- **Offline profile** – Use this page to create, review, and edit offline profiles. Each offline profile lets a user configure settings that are related to offline mode. For example, you can configure settings that let users manually switch to offline mode before they sign in, enable advanced offline switching, and pause offline

synchronization. These settings are discussed later in this topic and also in the related topics that are listed at the end of this topic.

- **Commerce channel schema** – Use this page to create, review, and edit channel schemas. By default, one schema that is named **AX7** is already created and available. The channel schema is required to define how the Commerce headquarters database should be read for Commerce data. It also includes a setting that lets you exclude customer-related data from data synchronization to offline databases. This setting is discussed later in this topic and also in the related topics that are listed at the end of this topic.
- **Distribution schedule** – Use this page create, review, and edit distribution schedule jobs. These schedule jobs determine which channel database groups run the associated scheduler job (see the next item in this list). A schedule job can be marked as active, and a single direction of data is associated with it. Typically, the direction is **Download**, so that data is sent down to the channels. By default, all Commerce-related jobs already exist and are ready to be used with any generated Commerce Scale unit (Cloud). From this page, you can also create delta synchronizations for a selected schedule job.
- **Scheduler job** – Use this page to create, review, and edit a selected job from the schedule job. This job has a series of associated subjobs. It's also associated with a channel schema (typically, the default **AX7** schema). A job can be excluded from synchronization to offline databases.
- **Scheduler subjob** – Use this page to create, review, and edit a subjob. A subjob is associated with one or more jobs, as shown on the **Scheduler job** page. A subjob is associated with a single table in the Commerce headquarters database. It shows the channel field mapping, which lists all the related fields that are used in the database table.
- **Download sessions and Upload sessions** – Use these pages to review and edit download or upload sessions that were created through the data packages that were generated via the previously described pages. These pages show how many rows of data must be synced, when the data was made available, and when it was synced. They also show the overall size of the data package. These pages let you manage and troubleshoot data packages to some extent. For example, you can view any errors that have occurred, and cancel or delete any jobs that are causing an issue. For more information, see [Commerce Data Exchange best practices](#).

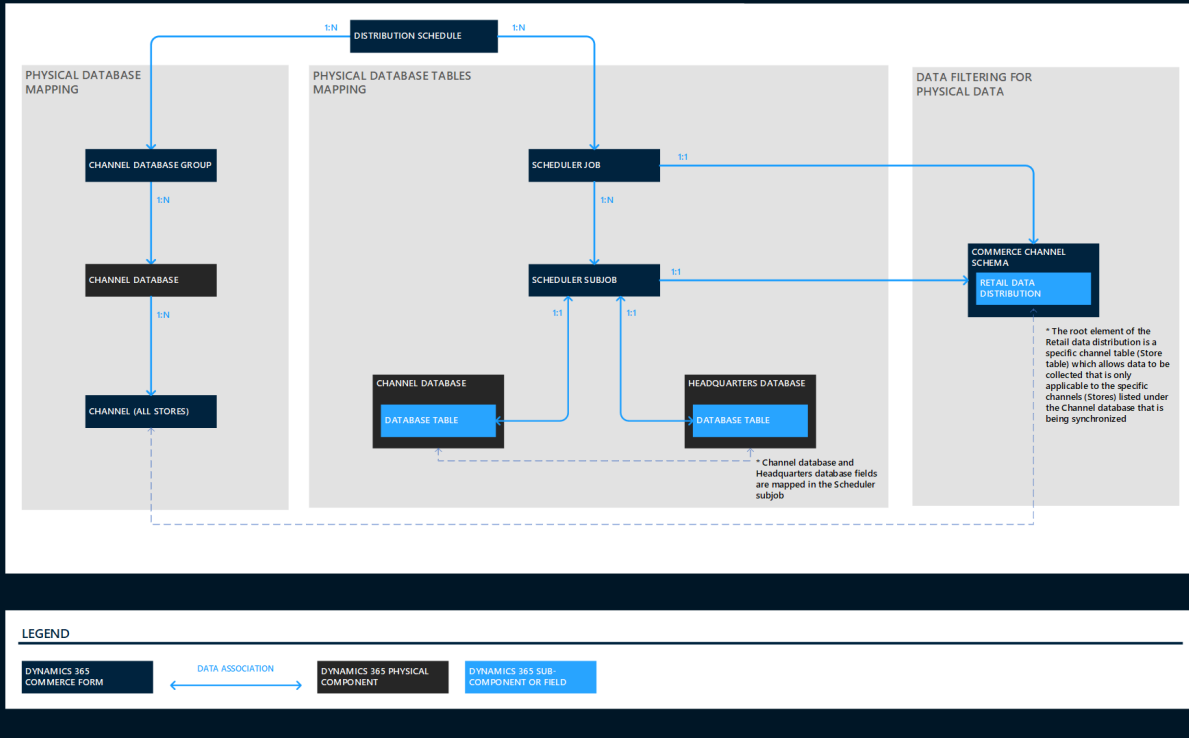
### **Data synchronization overview**

When a scheduler job is run, the channel database group selects, from the fields that are listed in the accumulated subjobs, the relevant data for all channel or offline databases that are associated with itself. The result of this data selection is a data package. A data package is a file or multiple files that are zipped together, and contains data that must be applied to one or more destination databases. This data is either all previously selected data or, typically, a selected delta of data. The destination databases can be either channel databases or offline databases.

Data is generated and flows in a specific direction (either download or upload). To understand how best to configure the timing and select data for synchronization, it's important that you understand how the various pages in Commerce headquarters are used and how data generation occurs. When data generation is done correctly, it helps increase performance and reduce Commerce headquarters utilization.

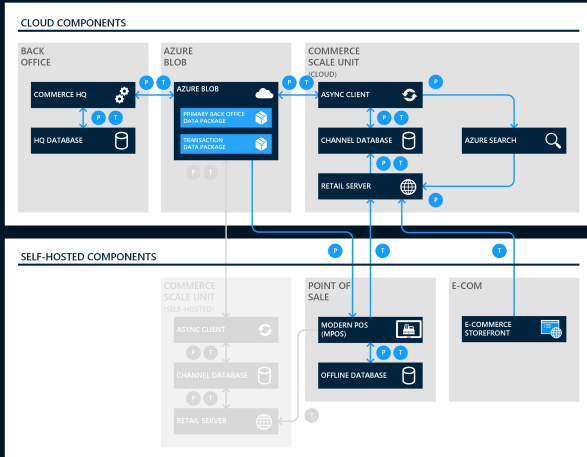
The following illustration shows the various pages in Commerce headquarters and how they are related to each another. (For descriptions of these pages, see the previous section.) CDX data generation can occur only if it's fully configured across all these pages. Data can be downloaded or uploaded. The data synchronization status is viewable on two different pages in Headquarters: Download sessions and Upload sessions. CDX data generation occurs through Headquarters and is synchronized down (download). Modern Point of Sale (POS) transactional data generated while offline requires the data to be synchronized up (upload).

### HEADQUARTERS FORMS AND COMPONENT ASSOCIATION

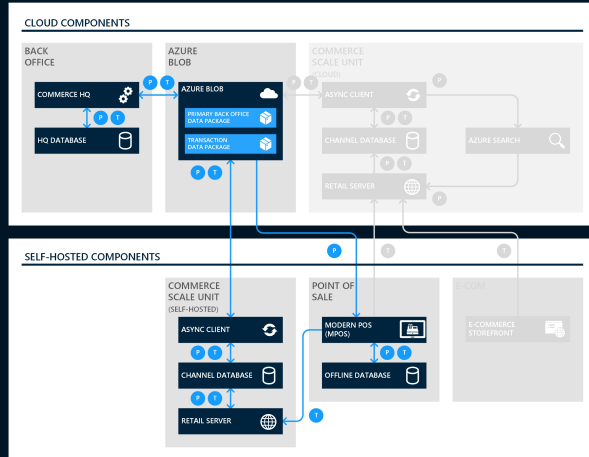


The following illustration shows the data flows for download and upload. Data packages that are generated through CDX flow downward. A generated data package can apply to the Commerce Scale Unit and to Modern POS offline databases, based on the channel database groups that are configured. Transactional data flows upward from the Modern POS offline databases to the Commerce Scale Unit channel database. All transactional data stores in the channel database is then uploaded to the Headquarters database.

### COMMERCE SCALE UNIT (CLOUD)



### COMMERCE SCALE UNIT (SELF-HOSTED)



### Overview of package management

As packages are created, they are processed and sent to Azure storage, where they are ready for download to a channel or offline database. During all stages from creation through full application, these packages can be viewed as a detailed list on the [Download sessions](#) page. In a similar way, as transactions are uploaded, there

is an **Upload sessions** page. On these pages, you can manage the packages to some extent during the various stages.

In general, packages automatically retry and take care of themselves. However, various scenarios can cause a package to become stuck in some way, so that it endlessly retries itself, but without success or outright failure. When you troubleshoot the package application process in these scenarios, the ability to delete or manually retry jobs can be valuable.

### Important CDX-related features

All these features are available in version 10.0.12 and later.

FEATURE NAME	DESCRIPTION
Advanced offline	This feature consists of a series of settings in the offline profile. These settings make additional offline switching scenarios available, give users the ability to switch to offline mode before they sign in to the POS, and allow for enhanced Commerce headquarters availability testing, so that you can switch to offline mode more often and more easily return to online status.
Offline data exclusion	This feature is also known as Data sizing improvements. It provides the ability to flag specific data that must be excluded from offline databases, and that must not be synced in the future. It's also used to fully remove all customer-related data from offline databases.
Forced Batch processing and inability to use recurrence with a Full sync	In version 10.0.11 and later, you can't perform the <b>Run now</b> command from the <b>Distribution schedule</b> page unless batch processing is used. This change was made because of performance issues that occurred if jobs were run during times when environments were most heavily used. In another change that was made as a part of this feature enhancement, recurrence can't be used when the <b>Full data sync</b> command (full job synchronization) is run from the <b>Channel database</b> page in Commerce headquarters. Only a single occurrence can be run.

#### Advanced offline

This feature can be configured in the offline profile. Three settings are related to it:

- **Allow manual switch to offline before sign in** – This setting lets Modern POS users switch to offline mode before they sign in to the POS. It's helpful in scenarios where time-outs might occur before sign-in is completed, or where atypical response codes from the Commerce Scale Unit (Cloud or Self-hosted) are occurring. When this setting is turned on, a Modern POS user who is using an offline database can access the **Settings** menu from the POS sign-in page. This menu includes a new option for switching to offline mode. By selecting this option, the user can sign in directly against the offline database instead of first having to sign in via a call to the Commerce Scale Unit.
- **Enable advanced offline switching** – This setting enables Modern POS to switch to offline mode more easily and more often. Typically, Modern POS tries to maintain its online status and switches to offline mode only when such a switch is required to continue functionality. When this setting is turned on, Modern POS can switch more often, especially in scenarios that involve sign-in and additional Commerce Scale Unit responses that might be considered a delay to POS operation. This setting is most valuable in scenarios where speed is a higher priority than maintaining availability of online-only features (for example, paying with a gift card, which requires connection to Headquarters).
- **System health check interval (mins)** – This setting works as a subfeature of the **Enable advanced offline switching** setting that was just described. Usually, when that setting is turned off, and Modern POS

is in offline mode, the POS waits a specific amount of time, based on configuration in the **Offline profile**, and then tries to reconnect to the Commerce Scale Unit during the next operation call that occurs. This advanced offline health check provides a more frequent, operation-independent method of checking online availability and switching more quickly as soon as online functionality is available again.

#### Offline data exclusion

This feature began to be released in version 10.0.11, and the full feature set was completed in version 10.0.12. This feature is intended to help reduce that amount of data that is synced to offline databases. On the **Scheduler job** and **Scheduler subjob** pages in Commerce headquarters, an option that is named **Exclude from offline databases** lets you exclude data (tables) when you sync data to the offline database.

- On the **Scheduler job** page, set the option to **Yes** to stop all generated data packages for the job from being synced to offline databases. If the **Full data sync** command is run for the excluded job from the **Channel database** page, the relevant tables in offline databases will be emptied (that is, all data that previously existed will be cleared).
- On the **Scheduler subjob**, set the option to **Yes** to stop the associated table for any job that contain the subjob from being synchronized to offline databases. For example, the channel database SQL table **DIRPARTYTABLE** is synchronized by three different scheduler jobs. Therefore, if you exclude the **DIRPARTYTABLE** table, you stop its data from being synced to offline databases by all three jobs. (We don't recommend that you exclude this example table (**DIRPARTYTABLE**) as it is critical for the staff related data it stores in the offline database).

This feature also represents the first step in row-level filtering. In Commerce headquarters, the **Commerce channel schema** page includes a new option that is named **Filter shared customer data tables**. (To open the **Commerce channel schema** page, go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database group**, and then, in the **Commerce channel schema** field, select a value. The default value is **AX7**.) By setting the option to **Yes**, you flag all customer data in shared tables. This setting works only for standard Microsoft-created tables (that is, tables that aren't custom-created tables). When you set this option to **Yes**, you receive a message that states, "This will remove customer data from the records in the channel data distribution only. All schedule jobs that contain customer data also need to be marked to skip offline synchronization." You can then select either **Yes** or **No**. This message is intended as a reminder that the **Exclude from offline databases** option must also be set to **Yes** for all customer data jobs. (By default, the only customer data job is the 1010 job.)

Here is an example of this filtering. The **DIRPARTYTABLE** table is used for both customers and employees. If you set the **Filter shared customer data tables** to **Yes**, all customer records at the row level are flagged to indicate that they should not be synced to offline databases. When the **Exclude from offline databases** option is also set to **Yes**, all customer data will be excluded from synchronization to offline databases.

## Implementation considerations

This section describes configurations that you should consider when you begin to plan your implementation. The features that are described here are related to data management and data configuration. Before you read the guidance that is provided here, we highly recommend that you read [Commerce Data Exchange best practices](#).

- **Create a Scheduler job calendar** – How often will each job occur? How many times per day will each job occur? Will large, non-critical jobs occur only during off-hours, when the overall environment isn't heavily used? By creating a calendar (either physical or virtual, as you prefer), you can learn the details about how jobs will intersect with other workloads that affect performance (for example, statement posting), hours of operation, batch processing for external data, and any customizations that push or pull data at specified times (or frequently throughout the day, just like a CDX job).
- **Pause offline synchronization** – As a retail organization expands, it should take advantage of this offline profile feature as fully as possible. Growth is good, but data generation should be managed to

help minimize the performance impact on the currently operating business. This feature enables the creation of channels, registers, and databases, but without requiring a massive, performance-affecting amount of data generation long before the registers are ever used.

- **Advanced offline** – The previously described advanced offline features can be helpful, but they should be used only if they suit the priorities and values of the retail organization. Although the advanced offline health check interval can help maximize online time, it will also be more forceful about pushing a register to offline mode if Commerce headquarters or the Commerce Scale Unit becomes unresponsive or unavailable for any reason. It can be valuable to maximize the performance of registers by quickly switching to offline mode instead of waiting for time-outs or repeated retry responses. However, this approach must be understood and managed against the standard seamless offline model that tries to stay online as long as possible, to allow for operations such as loyalty operations, additional payment methods, and customer orders.
- **Offline data exclusion** – In general, a small data set is typically faster than a large data set. It can be valuable to exclude data that isn't relevant to the functionality of the offline database when you want to reduce the overall database size (for example, SQL Express allows for databases of only 10 gigabytes [GB]), and also when you want to minimize the amount of data that the POS terminal queries as part of general operations while it's in offline mode.

This feature varies widely, depending on the business requirements of the retail organization. Therefore, it's crucial that you know what data is required for customizations to work, or even what data is required for standard day-to-day operations. For example, if a customer doesn't have to be attached to a transaction, customers can be excluded from the offline databases.

- **Channel database groups** – At a minimum, two channel database groups should exist: one for the initial Commerce Scale Unit (Cloud) that is used, and one for any or all offline databases that are used. Large retail enterprises might have multiple offline-focused channel database groups that are separated based on similarity of data in their associated channels (stores).

Additionally, it's helpful to have a "dummy" channel database group that can be used to configure new channel databases, registers that have offline support (in this case, offline databases will be created), and maybe even new but unused Commerce Scale Units (either Cloud or Self-hosted). Because this "dummy" group won't be associated with any distribution schedule jobs, no data generation will ever occur for anything that is associated with it. As time and the implementation progress, the associated entities (for example, channels [stores] and register offline databases) will be re-associated with the correct database group. A great alternative to this approach, and perhaps even an improvement, is to use the **Pause offline synchronization** feature that was described earlier.

## Resources

- [Commerce Data Exchange troubleshooting](#)
- [Commerce Data Exchange best practices](#)
- [Dynamics 365 Commerce architecture overview](#)
- [Select an in-store topology](#)
- [Device management implementation guidance](#)
- [Configure, install, and activate Modern POS \(MPOS\)](#)
- [Configure and install Commerce Scale Unit \(self-hosted\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Dynamics 365 Finance and Dynamics 365 Supply Chain Management - operated by 21Vianet in China

2/18/2021 • 4 minutes to read • [Edit Online](#)

## NOTE

Effective November 2020:

- Common Data Service has been renamed to Microsoft Dataverse. For more information, see [Power Automate Blog](#).
- Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. For more information, see [Terminology updates](#).

This topic will be updated soon to reflect the latest terminology.

Microsoft Dynamics 365 online services operated by 21Vianet is designed to comply with regulatory requirements in China. The services are a physically separated instance of cloud services operated and transacted by a local operator, Shanghai Blue Cloud Technology Co., Ltd (“21Vianet”). This is a wholly owned subsidiary of Beijing 21Vianet Broadband Data Center Co., Ltd. located in mainland China.

Microsoft strives to maintain functional parity between our commercially available service and Finance and Operations apps operated by 21Vianet in China. However, there are notable exceptions to this, which are affected by dependent service or partner-solution availability, market priorities, or compliance regulations.

## Provisioning

Customers in China have two options from which to select how they want to access Finance and Supply Chain Management apps.

- Services operated by 21Vianet in China - 21Vianet operates and offers Finance and Supply Chain Management services in China. This option provides a consistent application experience that is the same as global offerings. This option also meets the demands of customers who prefer to use online services provided by a local company that stores their data within China. These services are subject to Chinese laws.
- Services operated by Microsoft – This option is for Finance and Supply Chain Management customers that prefer to use services managed and delivered by Microsoft. For all new customers and existing customers, if the customer purchases Microsoft Azure, Dynamics 365, and Office using an Enterprise Agreement, Microsoft 365 and/or Dynamics 365 can co-exist on the tenant.

For information on provisioning environments, see [Create and manage environments in the Power Platform Admin center](#).

## Features not available

Due to certain technical dependencies, the following features listed will not be available for general availability of the Dynamics 365 Services operated by 21Vianet. For information about future feature availability, see [Business applications and platform release plans](#).

- **Development, build, and testing of customizations** will be unavailable in **Azure DevOps** in

**Mainland China.** However, use of Azure DevOps on-premises will be available in China in April 2019. Also, Azure DevOps can be used in other regions. For more information, see [Developer guide for Azure China 21Vianet](#).

- [Set up and maintain vendor collaboration](#) will be unavailable due to Azure Active Directory limitations.
- Certain **mobile apps** (e.g., [Install and configure the Warehousing app overview](#) and [Project time entry mobile workspace](#)) will be unavailable due to the Google Play Store not being available in China; however, alternatives are being considered.
- The **mobile platform** will not be available because certain App store dependencies are unavailable in China.
- The following **Microsoft Dynamics Lifecycle Services (LCS)** features will have a different experience or will be unavailable due to the dependencies that are not available:
  - **APQC Business process modeler (BPM) Library** will be unavailable. However, base Business process modeler (BPM) functionality will be available for custom models in April 2019. Search functionality in the BPM will be unavailable in China.
  - **Electronic reporting (ER) overview assets** will not be available automatically, but can be manually uploaded from the LCS global asset library.
  - **Code upgrade** will be unavailable for upgrades from Dynamics AX 2012.
  - **Service and Support requests** will be available through LCS but 21Vianet is the service operator. For more information, see [Support for Dynamics 365 Finance and Operations apps operated by 21Vianet in China](#).
  - **Extensibility requests** will be unavailable.
  - Hotfix requests will be unavailable.
  - [Dynamics 365 Translation Service overview](#) will not be available.
  - **Embedded Power Apps** and connectivity to Microsoft Power Apps and Microsoft Power Automate will be unavailable.
  - [Data integration using Dataverse overview](#) will be unavailable.

#### **NOTE**

The Lifecycle Services URL for implementations operated by 21Vianet in China is [lcs.dynamics.cn](https://lcs.dynamics.cn).

- The following features will not be available due to certain **current Azure Active Directory limitations** in China:
  - The **System administration > Setup > B2B Invitation configuration** page will not be available due to business-to-business (B2B) being unavailable in Azure Active Directory in China. For more information, see [What is guest user access in Azure Active Directory B2B](#).
- **Conditional access** is an Azure Active Directory feature that is available for the Azure Active Directory Premium 2 SKU. This is unavailable in China.
- The Microsoft Dynamics 365 Payment Connector for PayPal is not available in China.

## Additional resources

- [Dynamics 365 support site for 21Vianet \(Chinese\)](#)

- [Support for Dynamics 365 Finance and Operations apps operated by 21Vianet in China](#)
- [Model-driven apps in Dynamics 365 - operated by 21Vianet in China](#)
- [Dynamics 365 Privacy statement \(Dynamics 365 隐私声明\)](#)
- [Dynamics 365 Service Level agreement \(世纪互联在线服务的服务级别协议\)](#)
- [Dynamics 365 Legal information \(Dynamics 365 法律信息\)](#)
- [Service terms for Dynamics 365 Lifecycle Services](#)
- [OSPT of Dynamics 365 \(世纪互联在线服务的服务级别协议\)](#)
- [Azure Docs \(in Chinese\)](#)
- [Azure China 21Vianet](#)
- [Business applications availability in China – operated by 21Vianet in China](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Finance and Operations apps in France

2/18/2021 • 2 minutes to read • [Edit Online](#)

As of July 2019, Finance and Operations apps, Power Apps, and Power Automate are available in France's datacenters. This new deployment option serves customers in regulated industry and commercial organizations that do business with entities in France that require local data residency. Power BI has been available since March 2019.

The deployment of the Dynamics 365 services in France, are built upon the foundational principles of security, privacy, compliance, transparency, and reliability, offering French organizations a complete cloud infrastructure and platform, as well as familiar productivity and business application tools. All of this means that customer data stays resident within France.

Microsoft strives to maintain functional parity between our commercially available service and Dynamics 365 offerings in France. However, there are few exceptions affected by dependent service or partner-solution availability, market priorities, or compliance regulations. For more information about these exceptions or for questions about services in France, contact [Microsoft Dynamics Online support](#).

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Get support for Finance and Operations apps or Lifecycle Services (LCS)

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to get help with Finance and Operations apps or Microsoft Dynamics Lifecycle Services (LCS).

TASK	MORE INFORMATION
Ask the community.	Go to the <a href="#">Dynamics 365 Community</a> page to get help with your questions from the Microsoft Dynamics community.
Get help with questions about licensing.	Contact your partner or a Microsoft sales representative.
Use the <b>Issue search</b> tool.	In <b>LCS</b> , use the <b>Issue search</b> tool to quickly search for Microsoft Knowledge Base (KB) articles, hotfixes, and workarounds for reported issues. You can see which reported issues are in the process of being fixed for a specific functional area, and which issues have already been fixed. For more information, see <a href="#">Issue search (Lifecycle Services, LCS)</a> .
Get in-app support.	Select the <b>Help</b> button (?) in the upper-right corner of the app, and then select <b>Support</b> . Issues are reported on the <b>Active issues</b> tab in LCS. There, admins can determine whether they should provide in-house support or submit the issues to Microsoft.
Open a support ticket with the Microsoft Support team.	In <b>LCS</b> , the <b>Support</b> tile opens a tool that helps you manage support incidents. To submit issues directly to Microsoft, select the <b>Support</b> tile in your LCS project. You can then submit issues in two ways: <ul style="list-style-type: none"><li>• On the <b>Active issue</b> tab, select your issue, and then select <b>Submit to Microsoft</b>.</li><li>• On the <b>Submitted to Microsoft</b> tab, select <b>Submit an incident</b>, and then follow the on-screen instructions to submit the incident. After you submit an incident, you will receive an email message from the Microsoft Support engineer who is assigned to your case.</li></ul>
Request new features and functionality.	Visit <a href="#">Dynamics 365 Application Ideas</a> to view, search, or vote for existing ideas, or to add new ideas.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upgrade and N-1 support for Commerce

2/18/2021 • 2 minutes to read • [Edit Online](#)

Upgrade and N-1 support have been enabled in the July 2017 release of Microsoft Dynamics 365 Retail. N-1 support lets customers who have stores that run Microsoft Dynamics AX 2012 R3 Cumulative Update 10 (CU10) work with Headquarters after an upgrade. The main purpose of upgrade and N-1 support is to let AX 2012 R3 customers take advantage of the benefits of the cloud.

The following features let customers upgrade in a seamless manner:

- Users can now perform database upgrade from AX 2012 R3 to Headquarters through the upgrade process.
- Users can operate their stores by using the components of AX 2012 R3 CU10.
- Pre-upgrade and post-upgrade checklist validations are built into the upgrade process.
- The upgrade process has enhanced error handling and messaging, so that customers can quickly debug issues.
- Users can use tools to bring forward custom X++ code in their existing Headquarters to the upgraded version of the headquarters.

The upgrade procedure is largely the same as the procedure for upgrading Retail to the latest version. For details about upgrade in general, see [Upgrade overview: AX 2012 to Dynamics 365 Retail](#).

Planned downtime is required. Upgrade analysis is done first. The upgrade analysis runs against the Microsoft Dynamics AX 2012 database and is based on the Microsoft Dynamics Lifecycle Services (LCS) Diagnostic service. This step identifies tasks that can help make upgrade faster and less expensive. It also identifies the required SQL configuration, data headquarters cleanup, and deprecated features.

The actual data upgrade process then occurs. The AX 2012 database is moved to Microsoft Azure SQL Database, and then the data upgrade package is run as usual, through the runbook process. Upgrade validation is then done. A validation tool is run against the upgraded environment before it's used. This tool does an automated smoke test to verify that the service is running and accessible, row counts match, financials and inventory reconcile, and so on.

Most of the post-upgrade configuration for channels requires few manual steps. Customers can then use the pre-upgrade and post-upgrade checklists to learn about the tasks that must be completed. The post-upgrade tasks include initiating a full sync to the stores from the upgraded database, validating channels, registers, and devices against the upgraded database, validating transaction synchronization, and validating that N-1 support is in place.

For N-1 support, the customer must install the N-1 package in Commerce headquarters. No setup is required in the stores. This installation must be done during the upgrade window, after the N-1-related configuration has been completed.

After the Commerce headquarters upgrade and N-1 setup are completed, the N-1 store components can communicate with Commerce headquarters. No channel-side components must be installed for N-1 support. However, to enable the N-1 store to communicate with Commerce headquarters, cashiers must change their password the first time that they sign in.

For instructions for N-1 installation, see [Phased Rollout \(N-1\) installation, configuration, and cutover guide](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Phased Rollout (N-1) installation, configuration, and cutover guide

2/18/2021 • 29 minutes to read • [Edit Online](#)

This topic explains how to set up Phased Rollout (N-1) components so that your Microsoft Dynamics AX 2012 R3 channel components, such as Microsoft Dynamics AX for Retail Modern Point of Sale (MPOS) and Commerce Scale Unit, or Microsoft Dynamics AX for Retail Enterprise Point of Sale (EPOS), can work with Microsoft Dynamics 365 Commerce Headquarters.

## Key terms

TERM	DESCRIPTION
N-1 Async Server Connector Service	A component that is used to synchronize data packages between Headquarters and the AX 2012 R3 channel components.
N-1 Real-time Service	A component that supports real-time calls from the AX 2012 R3 channel components to Headquarters.

## Overview

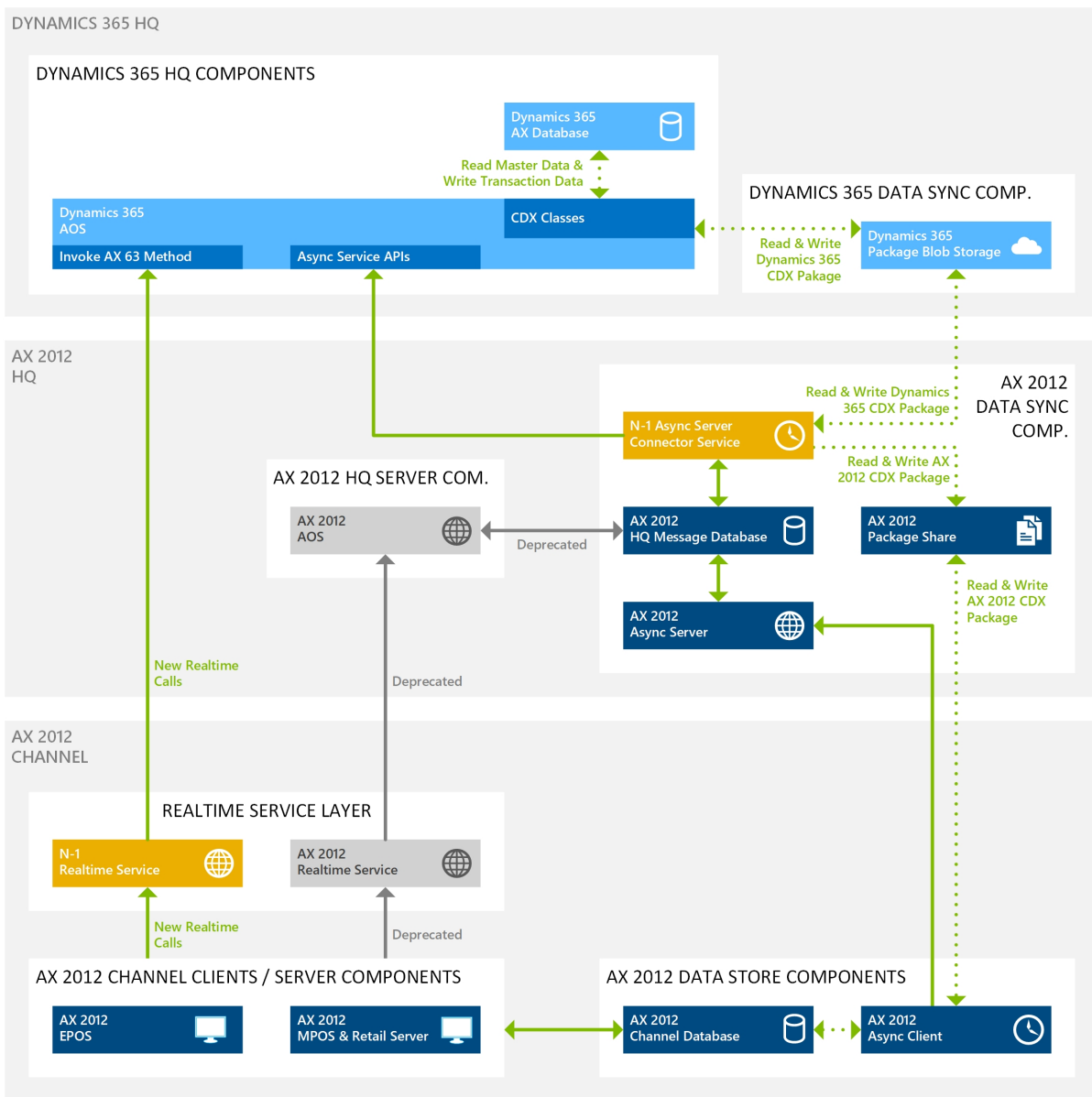
The sections in this topic describe the following steps, which you must complete to set up an environment with N-1 components. These steps assume that Headquarters is already deployed, and that an AX 2012 R3 environment is currently running.

- **Set up Azure AD accounts** – This section explains how to set up the Microsoft Azure Active Directory (Azure AD) accounts that the N-1 components use to connect to Headquarters.
- **Configure N-1 components** – This section explains how to configure the N-1 components in Headquarters.
- **Install N-1 components** – This section explains how to download and install N-1 components in the existing AX 2012 R3 environment.
- **Cutover steps to switch to N-1** – This section explains the how to use the new N-1 components to cut an existing AX 2012 R3 environment over from the AX 2012 R3 headquarters to the Dynamics 365 Headquarters.
- **Troubleshooting steps** – This section describes troubleshooting steps for typical issues.
- **Required KBs for N-1** – This section lists the Microsoft Knowledge Base articles (KBs) that are required in order to set up an N-1 environment.

### High-level architecture

The following illustration shows a high-level overview of the N-1 setup.





## Verify that the N-1 license key is turned on

Before you start to configure and install the N-1 components, make sure that the corresponding license key is turned on. This license key is automatically turned on during an upgrade from AX 2012 R3 to Microsoft Dynamics 365 Commerce. However, because the steps that follow require this key, you should verify that it's turned on before you continue.

1. Sign in to Commerce Headquarters, and go to **System administration > Setup > License configuration**.
2. On the **Configuration keys** tab, expand the **Commerce** key, expand the **Commerce scheduler** key, and verify that the check box for the **Retail Data Commerce Exchange backward compatibility** key is selected.

### NOTE

If the key isn't turned on, contact Microsoft Support for help turning it on.

## Set up Azure AD accounts

**IMPORTANT**

To help maintain a high level of security across the company, we strongly recommend that you create a new client ID and secret for this installation. This step requires a new Web App.

1. Generate an Azure Web App to create a client ID and secret for Connector for Microsoft Dynamics AX. For instructions, see the "Create an Azure Active Directory application" section in [Create an Azure Active Directory Application](#).
2. After you've finished creating the client ID and secret, the client ID must be accepted in Commerce. Go to **System administration > Setup > Azure Active Directory applications**. Enter the client ID in the **Client Id** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.

## Configure N-1 components

Follow the steps in this section to configure the N-1 components in Headquarters.

### Connector for Microsoft Dynamics AX

1. Sign in to Commerce Headquarters, and go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Connector for Microsoft Dynamics AX**.
2. On the Action Pane, select **New**, and set the following fields.

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
Header	Profile	Enter a unique name for the N-1 profile that you're setting up. You should create one profile for each environment/server where the AX 2012 R3 Real-time Service is currently installed.	AXConnect
Header	Description	Enter descriptive text to help identify the profile.	Connector for AX2012
Connection	Server	Enter the name of the server where the N-1 Real-time Service will be installed. (This server is the same server where the AX 2012 R3 Real-time Service is currently installed.)	sampleserver.local
Connection	Port	Enter the port that the N-1 Real-time Service will use.	8017
Connection	Web application name	Enter the name to use for the N-1 Real-time Service in Microsoft Internet Information Services (IIS).	Real-timeServiceAX63
Connection	Protocol	Select the protocol that the AX 2012 R3 Real-time Service currently uses.	https

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
Connection	Common name	Enter the friendly name of the certificate that the N-1 Real-time Service uses.	samplecertificate.local
Connection	Passphrase	Enter the passphrase to use for the N-1 Real-time Service.	
Connection	Language	Select the language that is associated with the stores that are mapped to the N-1 Real-time service.	EN-US

3. When you've finished, select **Save**.

### Commerce shared parameters

1. Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
2. On the **Security** tab, set the following fields.

FIELD	DESCRIPTION	SAMPLE VALUE
Real-time Service profile	Select the profile that you created on the <b>Connector for Microsoft Dynamics AX</b> page in the previous section.	AXConnect
TS password encryption name	Enter the algorithm that is used to connect to the transaction service. You should set this field to <b>SHA256</b> .	SHA256
Legacy device algorithm	Enter the AX 2012 R3 device algorithm.	AES

### Commerce scheduler parameters

1. Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Parameters > Commerce scheduler parameters**.
2. On the **HQ Message Database** FastTab, set the following fields.

FIELD	DESCRIPTION	SAMPLE VALUE
SQL Server instance name	Enter the name of the Microsoft SQL Server instance that hosts the existing AX 2012 R3 HQ Message Database.	sampleinstance
Server name	Enter the name of the server that hosts the existing AX 2012 R3 HQ Message Database.	sampleserver
Database name	Enter the name of the AX 2012 R3 HQ Message Database. In AX 2012 R3, the default name was <b>HQMessageDB</b> .	HQMessageDB

- When you've finished, select **Save**.

### Working folders

#### NOTE

The field values that are described here are the default values that are automatically set when environments are upgraded from AX 2012 R3. However, you should verify that they are correct. These values must be manually entered for environments that haven't been upgraded from AX 2012 R3.

- Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Working folders**.
- Make sure that the following field values are set in the grid.

FIELD	DESCRIPTION	SAMPLE VALUE
Name	The name of the working folder.	File storage for N-1
Description	A description of the working folder.	File storage for N-1
Download path	The network path where the AX 2012 R3 Commerce Data Exchange (CDX) download packages are stored.	\\sampleserver\Download
Upload path	The network path where the AX 2012 R3 CDX upload packages are stored.	\\sampleserver\Upload

- When you've finished, select **Save**.

### Channel database group

#### NOTE

The field values that are described here are the default values that are automatically set when environments are upgraded from AX 2012 R3. However, you should verify that they are correct. These values must be manually entered for environments that haven't been upgraded from AX 2012 R3.

- Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database group**.
- For each physical channel database in the AX 2012 R3 environment, on the Action Pane, select **New**, and set the following fields.

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
Header	Name	Enter the name of the channel database group that is used for the AX 2012 R3 environment.	Default_AX63
Header	Description	Enter a description of the channel database group that is used for the AX 2012 R3 channel environment.	Default group for AX63 channel database

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
<b>General</b> FastTab	Channel schema	Select the AX 2012 R3 schema. This field must be set to <b>AX2012R3</b> .	AX2012R3
<b>General</b> FastTab	Working folders	Select the reference to the working folders record that is used for the synchronization of CDX data packages. You created this working folders record in the previous section.	workingfolder

3. When you've finished, select **Save**.

### Channel database

#### NOTE

The field values that are described here are the default values that are automatically set when environments are upgraded from AX 2012 R3. However, you should verify that they are correct. These values must be manually entered for environments that haven't been upgraded from AX 2012 R3.

1. Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database**.
2. For each physical channel database in the AX 2012 R3 environment, on the Action Pane, select **New**, and set the following fields.

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
Header	Channel database ID	Enter a unique identifier for the physical channel database. This value should be derived from the AX 2012 R3 environment.	Default_AX63_Database
Header	Channel database group	Select the channel database group that the channel database is mapped to. This value should be derived from the AX 2012 R3 environment.	Default_AX63_DatabaseGroup
Header	Type	Select the type of the channel database record. This field must be set to <b>Channel database</b> .	Channel database
Header	Data sync interval	Select the interval at which to run the CDX data synchronization. <b>This field must be left blank.</b>	

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
Header	Username (Case Sensitive)	Enter the user name to use to connect to the AX 2012 R3 channel database.	channeluser
Header	Password	Enter the password to use to connect to the AX 2012 R3 channel database.	<i>passphrase</i>
Header	Database name	Enter the name of the AX 2012 R3 channel database.	SampleChannelDB
Header	Server name	Enter the name of the server that hosts the AX 2012 R3 channel database.	sampleserver
<b>Channel</b> FastTab	Channel	Select <b>Add</b> to add the list of channels for Phased Rollout (N-1) that are mapped to the channel database. The values should be derived from the AX 2012 R3 environment.	London

3. When you've finished, select **Save**.

### Channel profiles

#### IMPORTANT

This section applies only if the existing AX 2012 R3 environment uses Retail Server to interact with the channel database. If direct channel database access is enabled from the AX 2012 R3 MPOS, you can skip this step.

The field values that are described here are the default values that are automatically set when environments are upgraded from AX 2012 R3. However, you should verify that they are correct. These values must be manually entered for environments that haven't been upgraded from AX 2012 R3.

1. Sign in to Headquarters, and go to **Retail and Commerce > Channels > Stores > All stores**.
2. For each Retail Server that is hosted in the N-1 environment, on the Action Pane, select **New**, and set the following fields.

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
---------	-------	-------------	--------------

SECTION	FIELD	DESCRIPTION	SAMPLE VALUE
Header	Name	Enter a unique name for the N-1 channel profile that you're setting up. You should create one profile per environment/server where you currently have installed the AX 2012 R3 Retail Server. You should create one profile for each environment/server where the AX 2012 R3 Retail Server is currently installed.	Default_AX63_Profile
Profile properties	Property key	Enter the key for the Retail Server URL that is used in the AX 2012 R3 environment. This field must be set to <b>Retail Server URL</b> .	Retail Server URL
Profile properties	Property value	Enter the Retail Server URL that is used in the AX 2012 R3 environment.	http://localhost:35080/RetailServer/v
Profile properties	Property key	Enter the key for the Hardware Station URL that is used in the AX 2012 R3 environment.	Hardware Station URL
Profile properties	Property value	Enter the Hardware Station URL that is used in the AX 2012 R3 environment.	ipc://localhost

3. When you've finished, select **Save**.

### All stores

#### NOTE

The field values that are described will be set if the Headquarters environment was upgraded from the AX 2012 R3 headquarters.

1. Sign in to Headquarters, and go to **Retail and Commerce > Channels > Stores > All stores**.
2. For each store that is mapped to the N-1 environment, select the channel ID. Then, on the store details page, on the **General** FastTab, set the following fields.

FIELD	DESCRIPTION	SAMPLE VALUE
Live channel database	Select the channel database that you mapped the store to and set it up for earlier.	Default_AX63_Database
Channel profile	Select the channel profile that you mapped the store to and set it up for in the previous section.	Default_AX63_Profile

3. When you've finished, select **Save**.

### Initialize the AX 2012 scheduler

1. Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Initialize AX 2012 retail scheduler**.
2. Select **OK**.

### Distribution schedule

#### NOTE

The field values that are described will be set if the Headquarters environment was upgraded from the AX 2012 R3 headquarters.

1. Sign in to Headquarters, and go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. In the left pane, select each distribution schedule that has the suffix **AX63**. Then, on the **Channel database groups** FastTab, make sure that the entries that you created earlier are mapped to the distribution schedule.

### Workers

#### NOTE

The field values that are described will be set if the Headquarters environment was upgraded from the AX 2012 R3 headquarters.

1. Sign in to Headquarters, and go to **Retail and Commerce > Employees > Workers**.
2. Select the name of each worker who will sign in to the AX 2012 R3 environment. Then, on the worker details page, on the **Commerce** tab, on the **Commerce** FastTab, set the **Password** field.

## Install N-1 components

Before you run the Connector for Microsoft Dynamics AX installers, make sure that the following requirements are met:

- The installer requires that the Microsoft .NET Framework version 4.5.1 be installed on the system.
- The installers install the Connector for Microsoft Dynamics AX applications only on the following operating systems. Before you install any component, you must update the operating system with all service packs and updates that are available for it.
  - Windows 7 Professional, Enterprise, or Ultimate edition (both x86 and x64 architectures). Home edition and embedded edition aren't supported.
  - Windows 8.1 Update 1 Pro or Enterprise edition (both x86 and x64 architectures). Standard edition isn't supported.
  - Windows 10 Pro or Enterprise edition (both x86 and x64 architectures). Home edition isn't supported.
  - Microsoft Windows Server 2012 R2 or Microsoft Windows Server 2016.

### Async Server Connector Service

The installer validates that all prerequisites are met. These prerequisites include SQL prerequisites, such as a local installation of SQL Server, or the alternative prerequisites, such as SQL Command Line Utilities and other required SQL connectivity installations.

To meet prerequisites, the SQL Server that is connected to must have Full-text search and, at a minimum, support for Transport Layer Security (TLS) 1.2. For Microsoft SQL Server 2012, Service Pack 3 must be installed, at a minimum. For Microsoft SQL Server 2014, Service Pack 2 must be installed.



If a system restart is required, the installer shows this requirement. Although the restart is recommended, the installer can continue without it.

When you're ready, follow these steps to download and install the component.

1. Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Connector for Microsoft Dynamics AX**.
2. Select the connector that you created earlier.
3. On the Action Pane, on the **Download** tab, select **Async Server Connector service** to download the installer file, and then select **Configuration file** below **Async Server Connector service** to download the corresponding configuration file. Make sure that the configuration file is saved next to the installer file.
4. After the installer and configuration files are downloaded, double-click the installer file, and follow these steps:
  - a. Verify the Application Object Server (AOS) URL (the URL that is used to access Headquarters), and then select **Next**.
  - b. If a specific user is required, enter the user name and password that the service should run as. By default, the installer automatically generates a service account to use. This approach is more secure and is recommended, but it can't be used when the database is located on a separate computer. Select **Next** to continue.
  - c. Enter the application ID (client ID) and secret that are associated with this Connector for Microsoft Dynamics AX installation.
  - d. Select **Install**.

#### NOTE

- For information about how to correctly generate an Azure Web App to create a client ID and secret, see the "Basics of Registering an Application in Azure AD" section in [Create an Azure Active Directory Application](#).
- When you create the Web App, the initial URI and URL don't have to be any specific value. Only the application ID (client ID) and secret that are created are important.

5. After an application ID (client ID) and secret are created, the application ID must be accepted in Commerce. Go to **Retail and Commerce > Headquarters setup > Azure Active Directory applications**. Enter the application ID in the **Client Id** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.
6. Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
7. On the **Identity Providers** tab, select the provider that has an **Issuer** value that begins with `HTTPS://sts.windows.net/`. The values on the **Relying parties** FastTab are automatically set, based on your selection.
8. On the **Relying parties** FastTab, select **Add**. Enter the application ID (client ID) that was created for this installation. Set the **Type** field to **Public** and the **UserType** field to **Worker**. Then, on the Action Pane, select **Save**.
9. On the Action Pane, select **Save**.

#### N-1 Real-time service

1. Sign in to Headquarters, and go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Connector for Microsoft Dynamics AX**.
2. Select the connector that you created earlier.

3. On the Action Pane, on the **Download** tab, select **Real-time service for Dynamics AX 2012 R3** to download the installer file, and then select **Configuration file** below **Real-time service for Dynamics AX 2012 R3** to download the corresponding configuration file. Make sure that the configuration file is saved next to the setup file.
4. After the installer and configuration files are downloaded, double-click the installer file, and follow these steps:
  - a. Verify the AOS URL (the URL that is used to access Headquarters), and then select **Next**.
  - b. Select a valid SSL certificate to use for HTTPS communication, and then select **Next**.

The certificate must use private key storage, and server authentication must be listed in the enhanced key usage property. Additionally, the certificate must be trusted locally, and it can't be expired. It must be stored in the personal certificate store location on the local computer.

- c. If a specific user is required, enter the user name and password that the application pool should run as. By default, the installer automatically generates a service account to use. This approach is more secure and is recommended, but it can't be used when the database is located on a separate computer. Select **Next** to continue.
- d. Verify the HTTPS port that should be used, and verify that the host name of the computer is correct. Select **Next** to continue.

The HTTPS port is listed in the Store system profile. To access the Store system profile, on the **Store details** page, on the **Store systems** FastTab, select the profile ID of the selected Store system. The installer automatically enters the host name. If, for any reason, the host name must be changed for the installation, change it here. The host name must be the fully qualified domain name (FQDN) of the system, and it must match the value that you entered on the **Connector for Microsoft Dynamics AX** page earlier in this topic.

- e. Enter the application ID (client ID) and secret that are associated with this Connector for Microsoft Dynamics AX installation. Then select **Install**.

This application ID and secret can be the same application ID and secret that you used in the Async Server Connector service installation. For information about how to correctly generate an Azure Web App to create a client ID and secret, see the "Basics of Registering an Application in Azure AD" section in [Create an Azure Active Directory Application](#). When you create the Web App, the initial URI and URL don't have to be any specific value. Only the application ID (client ID) and secret that are created are important.

## Cutover steps to switch to N-1

This section gives the recommended step-by-step instructions to switch an existing AX 2012 R3 channel environment from the AX 2012 R3 headquarters to the Commerce Headquarters. Note that these instructions are generic. Different implementations will likely have to deviate from these steps to accommodate specific business or technical requirements.

### Prerequisites

Follow these steps to prepare your environment for the cutover.

STEP	DETAILS	TIMELINE
1. Deploy Headquarters.	Headquarters is up and running. Cloud POS (CPOS) can be used to validate functionality in the environment.	Weeks or months before the cutover

STEP	DETAILS	TIMELINE
2. Install the Commerce application (X+ +) KBs.	Install the KBs that are listed in the <a href="#">Required KBs for N-1</a> section to make sure all issues that are related to N-1 are addressed.	Weeks or months before the cutover
3. Set up Azure AD accounts.	Follow the instructions in the <a href="#">Set up Azure AD accounts</a> section to create the accounts that are required for the N-1 components to authenticate against Headquarters.	Weeks or months before the cutover
4. Configure Headquarters.	Follow the instructions in the <a href="#">Configure N-1 components</a> section to configure all the settings for the N-1 components before they are installed.	Weeks or months before the cutover
5. Configure Headquarters.	Go to <b>Retail and Commerce &gt; Headquarters setup &gt; Parameters &gt; Commerce shared parameters</b> , select the <b>Security</b> tab, and change the value in <b>TS password encryption name</b> to "SHA256".	
6. Install the N-1 components.	Follow the instructions in the <a href="#">Install N-1 components</a> section to install the N-1 components. Note that the N-1 Async Server Connector Service component should be installed but immediately turned off to ensure that AX 2012 R3 and Dynamics 365 CDX packages aren't mixed.	Weeks or months before the cutover

## Preparation

Follow these steps to prepare a few days before the cutover is scheduled.

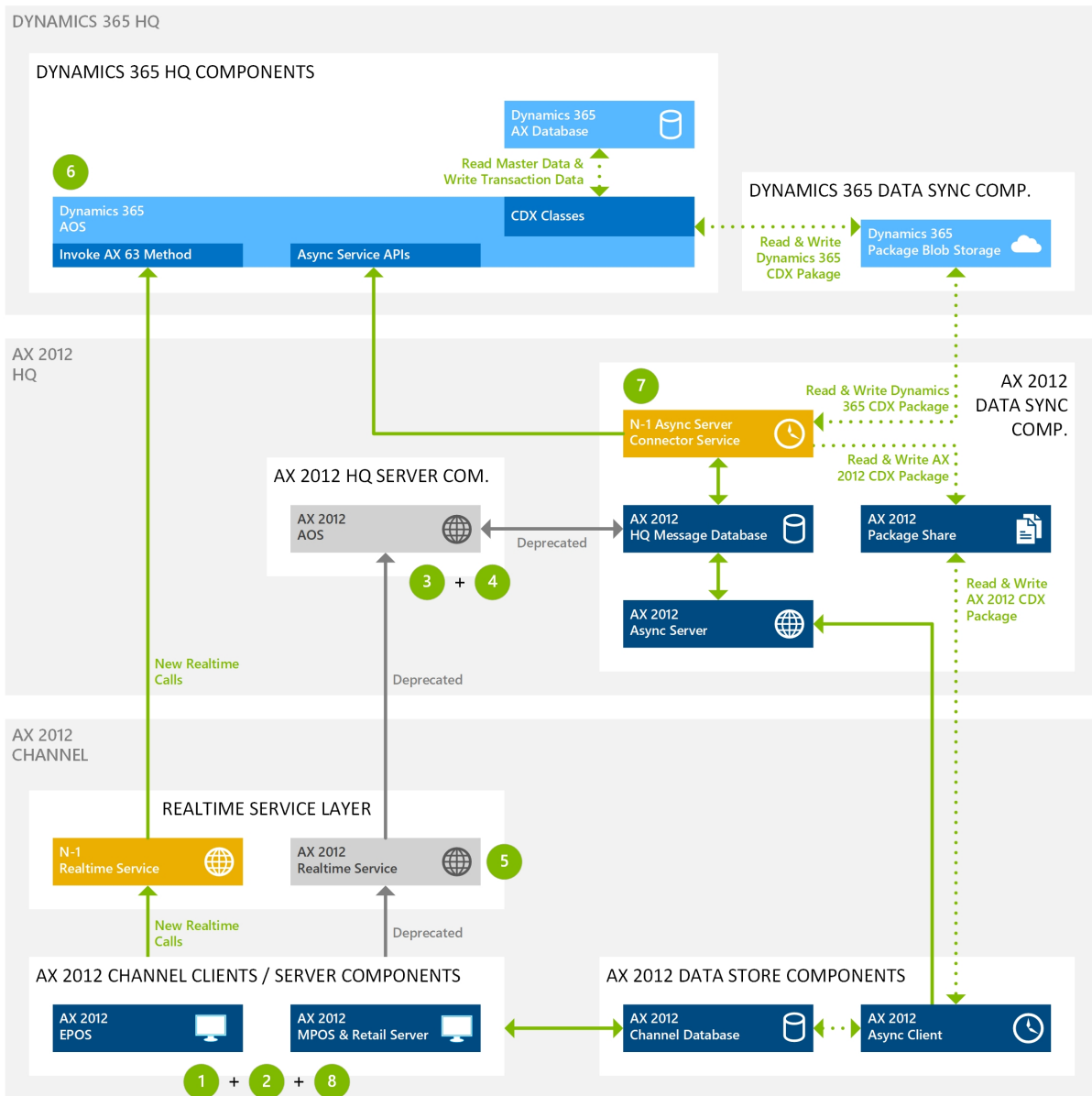
STEP	DETAILS	TIMELINE	HOW TO VALIDATE THAT THIS STEP IS DONE
1. Stop all AX 2012 R3 download jobs.	Make sure that all AX 2012 R3 download jobs are stopped in the AX 2012 R3 headquarters.	At least a couple days before the cutover	All packages in the AX 2012 R3 network share for upload jobs are processed, and no new packages appear.
2. Do a full synchronization of all AX63 CDX download jobs in Headquarters.	Run the CDX download jobs to make sure that the packages are generated and dropped in the Azure Blob storage blob, so that the N-1 Async Server Connector Service can consume them during cutover.	At least a couple days before the cutover	CDX download jobs are in the <b>Available</b> state on the <b>Download Sessions</b> page in Headquarters.

## Cutover steps

Follow these steps to do the actual cutover.

STEP	DETAILS	TIMELINE	HOW TO VALIDATE THAT THIS STEP IS DONE
1. Close all shifts.	Make sure that all shifts are closed.	After store closing	Manually validate with the stores that the shifts are closed.
2. Clean up any suspended transactions.	Make sure that all suspended transactions are cleaned up.	After the previous step	Manually validate with the stores that the shifts are closed.
3. Synchronize all transactional data.	Make sure that all transactional data is synchronized through the CDX upload jobs to the AX 2012 R3 headquarters.	After the previous step	All packages in the AX 2012 R3 network share for upload jobs are processed, and no new packages appear.
4. Disable AX 2012 R3 CDX upload jobs.	Make sure that all AX 2012 R3 upload jobs are disabled, so that packages are no longer being picked up.	After the previous step	Manually validate through the AX 2012 R3 headquarters UI that the CDX upload jobs are disabled.
5. Shut down the AX 2012 R3 Real-time Service.	Connect to the server that hosts the AX 2012 R3 Real-time Service, start IIS, right-click the AX 2012 R3 Real-time Service, and stop the service.	After the previous step	Manually validate in IIS that the service is stopped.
6. Run the <b>Reset metadata synchronization</b> command in Headquarters.	In Microsoft Dynamics AX, go to <b>Retail and Commerce &gt; Headquarters setup &gt; Commerce scheduler &gt; Connector for Microsoft Dynamics AX</b> , and select <b>Reset metadata synchronization</b> to reset the <b>HQMessageDB</b> database in the AX 2012 R3 environment for N-1 cutover.	After the previous step	Not applicable
7. Turn on the N-1 Async Server Connector Service.	Connect to the server that hosts the N-1 Async Server Connector Service, and enable the Microsoft Windows service.	After the previous step	The status of download jobs on the <b>Download Sessions</b> page in Headquarters changes from <b>Available</b> to <b>Applied</b> .

The following illustration is a visual representation of these steps.



## Troubleshooting steps

This section describes some typical issues that you might encounter, and the steps that you can follow to investigate or recover from them.

### Runtime

This section describes troubleshooting steps for errors that you might encounter during the runtime of the application.

#### Metadata synchronization fails

FIELD	VALUE
Event Log	Microsoft-Dynamics-Commerce-AsyncServerConnectorService/Operational

FIELD	VALUE
Sample Event Log Error Message	<p>Async server connector service encounters error in download timer tick. CorrelationId {4c9cd9a0-d4e3-43e5-80da-59ea2eb01acf}; Error details:  Microsoft.Dynamics.Retail.AsyncServerConnector.Service.Exceptions.SyncMetadataException: Failed synchronizing metadata. ---&gt;  Microsoft.Dynamics.Retail.AsyncServerConnector.Service.Exceptions.MessageDBOperationException: Failed updating metadata in HQ message DB. ---&gt;  System.Data.SqlClient.SqlException: A connection was successfully established with the server, but then an error occurred during the login process. (provider: SSL Provider, error: 0 - The certificate chain was issued by an authority that is not trusted.) ---&gt;  System.ComponentModel.Win32Exception: The certificate chain was issued by an authority that is not trusted</p>
Troubleshooting Steps	<p>This could have happened because the Synch service web.config connection string has the TrustServerCertificate set to false. To fix the issue browse the Sync service website and open the web.config. Find the connectionStrings section, update the TrustServerCertificate to True if it is false.</p>

#### Metadata synchronization fails

FIELD	VALUE
Event Log	Microsoft-Dynamics-Commerce-AsyncServerConnectorService/Operational
Sample Event Log Error Message	<p>Async server connector service encounters error in download timer tick. CorrelationId {73d9d0d3-4d12-42ca-ac65-3f1f947c7840}; Error details:  Microsoft.Dynamics.Retail.AsyncServerConnector.Service.Exceptions.SyncMetadataException: Failed synchronizing metadata. ---&gt;  Microsoft.Dynamics.Retail.AsyncServerConnector.Service.Exceptions.MessageDBOperationException: Failed updating metadata in HQ message DB. ---&gt;  System.Data.SqlClient.SqlException: Cannot open database "HQMessageDB" requested by the login. The login failed. Login failed for user 'localhost\RetailAsUser'.</p>
Troubleshooting Steps	<p>This could have happened because the user who runs the Async server connector service does not have access to the HQ message database. To fix the issue run services.msc, check the user who runs the Async server connector service. Provide this user access to HQ message database in SQL.</p>

#### Metadata synchronization fails, or CDX jobs are successfully downloaded but aren't applied

FIELD	VALUE
Event Log	Microsoft Dynamics AX Retail : Async Client SynchClientService

FIELD	VALUE
Sample Event Log Error Message	<p>Unable to communicate with server for download. Please check username/password, server and database connections. Error Details: System.ServiceModel.CommunicationException: An error occurred while making the HTTP request to <code>https://localhost:44300/SynchService/DownloadService.svc</code>. This could be due to the fact that the server certificate is not configured properly with HTTP.SYS in the HTTPS case. This could also be caused by a mismatch of the security binding between the client and the server. ---&gt; System.Net.WebException: The underlying connection was closed: An unexpected error occurred on a send. ---&gt; System.IO.IOException: Authentication failed because the remote party has closed the transport stream.</p>
Troubleshooting Steps	<p>This could have happened because of the following reasons.</p> <ol style="list-style-type: none"> <li>1) The user ID and password that we provide in async client does not match the channel database user ID and password that we provide in AX.</li> <li>2) The async service end point is not reachable. <code>https://localhost:44300/SynchService/DownloadService.svc</code>. It is critical to note that typically the user is in the form of <b>Contoso/Administrator</b> (domain/user). In the following steps, this has been known to cause errors (communication or inability to save) if the domain is utilized in the <b>AsyncClientConfigurationUtility</b> utility. We recommend that the user ID in this utility does not contain the domain prefix that is used in the server installation and configuration.</li> </ol> <p>To fix this issue:</p> <ol style="list-style-type: none"> <li>1) Launch the Async client configuration utility (AsyncClientConfigurationUtility.exe) from the async client install location. Under the <b>Async Server connection tab &gt; Authentication information (case sensitive) &gt; Update the Channel database ID, User name and Password</b> fields as per the values provided under N-1 channel database (<b>Retail and Commerce &gt; Headquarters setup &gt; Channel database</b>) in AX.</li> <li>2) Select <b>Test connection</b> in the utility. If the connection is successful, restart the Async client service. If the connection fails, go to N-1 channel database (<b>Retail and Commerce &gt; Headquarters setup &gt; Channel database</b>) in AX, update the Username and password, and then save. Go to <b>Retail scheduler parameters</b> in AX and select <b>Reset metadata synchronization</b>. This will populate the HQ message database with the new values. Repeat step 1 again.</li> <li>3) If the async server end point (<code>https://localhost:44300/SynchService/DownloadService.svc</code>) is not reachable, check the bindings of the service and make sure there is an associated certificate. If the issue still exists, check to be sure that the <b>Working folders</b> were specified for all the Channel database groups that were linked to <code>AX2012R3</code> retail channel schema.</li> </ol>

**CDX jobs are successfully downloaded but aren't applied**

FIELD	VALUE
Event Log	Microsoft Dynamics AX Retail : Async Client SynchClientService
Sample Event Log Error Message	Exception during DownloadAgent.Execute. Error Details: System.IO.FileNotFoundException: Could not load file or assembly 'Microsoft.Dynamics.Retail.StoreConnect.Request.Base, Version=6.3.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35' or one of its dependencies. The system cannot find the file specified. File name: 'Microsoft.Dynamics.Retail.StoreConnect.Request.Base, Version=6.3.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35' at Microsoft.Dynamics.Retail.SynchClient.Core.DownloadAgent.ApplySessionFileToClientDatabase(SessionManager sessionMgr, String fileName) at Microsoft.Dynamics.Retail.SynchClient.Core.DownloadAgent.ProcessDownloadSession(DownloadSession session) at Microsoft.Dynamics.Retail.SynchClient.Core.DownloadAgent.Execute()
Troubleshooting Steps	This could have happened because the AX63 store connect components were not installed on the environment. To fix the issue install the AX63 store connect components on the environment that is running the async client component.

#### N-1 Async Server Connector Service communication exception

FIELD	VALUE
Event Log	Microsoft Dynamics AX Retail : Async Client SynchClientService
Sample Event Log Error Message	Unable to communicate with server for download. Please check username/password, server and database connections. Error Details: System.ServiceModel.CommunicationException: An error occurred while making the HTTP request to <a href="https://localhost:44300/SynchService/DownloadService.svc">https://localhost:44300/SynchService/DownloadService.svc</a> . This could be due to the fact that the server certificate is not configured properly with HTTPSYS in the HTTPS case. This could also be caused by a mismatch of the security binding between the client and the server. ---> System.Net.WebException: The underlying connection was closed: An unexpected error occurred on a send. ---> System.IO.IOException: Authentication failed because the remote party has closed the transport stream.
Troubleshooting Steps	Make sure the HQ Message DB has data. Use the asyncClient configuration tool to validate the connection is successful.

#### AX 2012 R3 MPOS activation fails with an error on step 1

FIELD	VALUE
Event Log	Microsoft Dynamics AX Retail : Retail Server RetailServer



FIELD	VALUE
Sample Event Log Error Message	Microsoft.Dynamics.Commerce.Runtime.UserAuthenticationException: An error occurred during logon. ---> Microsoft.Dynamics.Commerce.Runtime.ConfigurationException: The published channel cannot be found in local database. Please make sure at least 1 retail channel is published to this DB through AX.
MPOS Error on Activation Screen	DA1002: A server side error occurred that prevents user from logging on. Please check the server log for detailed information or contact your IT support.
Troubleshooting Steps	Make sure the CDX download jobs ran successfully and the channel database has data.

#### AX 2012 R3 MPOS activation fails with an error on step 2

FIELD	VALUE
Event Log	Microsoft Dynamics Retail Modern POS
Sample Event Log Error Message	Dynamics-Error: LoginViewModel ActivateDevice Logon failed. ErrorMessage: ; ErrorCode: Microsoft_Dynamics_Commerce_Runtime_HeadquarterTransactionServiceMethodCallFailure;
MPOS Error on Activation Screen	DA2001
Troubleshooting Steps	Device you are trying to activate is already active. Please deactivate the device and try the activation.

#### AX 2012 R3 MPOS activation fails with an error on step 2

FIELD	VALUE
Sample Event Log Error Message	Exception occurred: [04/19/2018 19:26:51] Microsoft.Dynamics.Commerce.Runtime.UserAuthenticationException: An error occurred during logon. ---> Microsoft.Dynamics.Commerce.Runtime.StorageException: Failed to read from the database. See inner exception for details. DatabaseErrorCode: 0 ---> Microsoft.Dynamics.Commerce.Runtime.Data.DatabaseException: Invalid object name 'crt.EMPLOYEEPERMISSIONSVIEW'. ---> System.Data.SqlClient.SqlException: Invalid object name 'crt.EMPLOYEEPERMISSIONSVIEW'. at System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) at System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose)
MPOS Error on Activation Screen	DZ1001
Troubleshooting Steps	Device you are trying to activate is already active. Please deactivate the device and try the activation.

#### AX 2012 R3 MPOS activation fails with an error on step 9

FIELD	VALUE
Event Log	Microsoft Dynamics Retail Modern POS
Sample Event Log Error Message	Dynamics-Error: LoginViewModel ActivateDevice Logon failed. ErrorMessage: Sorry, something went wrong with the encryption on your device. Please contact your system administrator.; ErrorCode: MICROSOFT_DYNAMICS_POS_DATAENCRYPTIONERROR;
MPOS Error on Activation Screen	DA3122: Sorry, something went wrong with the encryption on your device. Please contact your system administrator.
Troubleshooting Steps	Check the Real-time Service profile filed in the CDX Backward compatibility section of the store. This must be set to the RTS profile that you created for N-1.

## Required KBs for N-1

The following list describes all the KBs that are required for N-1 to work correctly.

### Dynamics 365 Retail – Microsoft Dynamics 365 7.2 headquarters

KB NUMBER	TITLE
4095190	Expose RetailSharedParameter's TransactionServiceProfileID in the RetailSharedParameters form as it is required to enable N-1 functionality when customer did not follow the official upgrade process to move data from 6.3 to Dynamics 365.
4095209	The Real-timeServiceAX63 N-1 component's webconfig contains an invalid authentication type name. This causes issue when the component tries to authenticate with the transaction service in Dynamics 365
4095189	RetailTransactionServiceProfile table's Protocol column is not being synced to the old version channel databases from the new version Dynamics 365 AX database.
4095191	The user is not allowed to set the retail server URL to be http based URL as only secured URLs are allowed. This prevents the user from connecting to old (6.3) version retail servers while running in backward compatibility mode.
4132456	[Upgrade & N-1][Designer] Number pad height should be extensible
4095926	Upgrade & N-1: Return order fails from 63MPOS as the transaction was not found on a N-1 non upgrade environment.
4095664	Allow Microsoft Dynamics AX 2012 clients connecting to AX7.2 HQ to create new customers.
4132454	N-1 version of 1070 fails due to duplicate record exception caused by the CompanyImage_AX63 subjob.

KB NUMBER	TITLE
4131243	When running Dynamics 365 Retail in N-1 mode, the user is not able to set the HardwareStationURL using the AX client hence may not be able to configure the 6.3 hardware station properly.
4338120	Async service connector installer should use provided HQ message db name instead of always using a default name regardless of the HQ message database name provided by the user.
4337834	When running Dynamics 365 in N-1 backward compatibility the N-1 (6.3)version MPOS may fails while adding or updating customer from MPOS due to missing methods in RetailTransactionService.
4337864	When running Dynamics 365 in N-1 backward compatibility the N-1 (6.3)version MPOS may fails during login or payment when attempting to load or use the hardware profile payment merchant property.
4132453	Running the N-1 version of the Product download job(1040_AX63) fails when moving data to the InventDim table in the 6.3 version channel database
4206905	POS reports throw errors when they are run in non-upgrade N-1 environment.
4133289	Running the N-1 version of the Product - Refunds from Journal or existing transactions failing
4161086	CDX table distribution ignores 'FieldValue' link types added on root table nodes and hence does not use that information for filtering the table using the provided condition (x++ fix)
4161099	CDX table distribution ignores 'FieldValue' link types added on root table nodes and hence does not use that information for filtering the table using the provided condition (Binary changes)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upgrade the Retail channel extension to the latest Retail SDK

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic provides information about how to upgrade to the latest update of the Retail SDK from earlier releases. The overall process and the supported scenario information are included, but this topic doesn't provide detailed instructions of every step in the process. This topic is applicable for Dynamics 365 Commerce and Dynamics 365 Finance.

The following sections will walk through how to manually move your extension to the new Retail SDK, however you can do this using any source control system like Azure DevOps or Git.

## Update the Retail SDK

If you deploy a new environment, the new Retail SDK is located in the services volume of the virtual machine (VM) or in the C drive of the downloadable VHD. When you update the Retail SDK by applying a new binary hotfix from Lifecycle Services (LCS), a new **Update** folder is created inside the existing RetailSDK folder and a copy of the new updated SDK is created inside the Update folder with name `{{Guid.RetailSDKUpdate.Date}}`. We recommend that you rename this folder using a shorter name otherwise when copying this folder you may get an error saying the path or file name is too long.

### Retail SDK components

The Retail SDK updates consist mainly of the following components:

- Assets: Configuration file changes related to extensions.
- Build tools: Customization package and versioning settings.
- Database: Extension DB scripts.
- Documents: Instructions to run the samples.
- Package: Deployment package.
- Payment externals: Payment packaging folders.
- Payment: Payment sample code.
- POS: Modern and Cloud POS App code and samples.
- References: All binary reference and commerce analyzer proxy tool.
- Sample extensions: Extension sample projects.

## Upgrade scenarios

Upgrade can be completed for one of the following scenarios:

- Update to a specific binary hotfix.
- Upgrade your custom code.
- Upgrade to the latest application release.

The process for SDK upgrade varies between versions. With version 7.3 and higher, we sealed all of the components and customizations must be completed only by using the extension points. This should result in an easier code upgrade experience. However, if you are upgrading from 7.0, 7.1, or 7.2 and if you have made inline changes, you must move the inline changes to extensions. This will require additional work.

#### NOTE

When upgrading to newer version, do not remove any of the existing Commerce Scale Unit, Commerce Runtime, Proxy, Hardware station, CDX, or Database extension code/APIs. Your POS client may depend on this code and removing it will cause runtime exception errors. If you want to remove it then during code upgrade, make sure that the client code is also updated to support this change, otherwise runtime failure will occur. As a best practice, extension code must be written in such a way that it is always backward compatible.

The following tables provide some high-level information about which version the code is sealed. If you are upgrading from an unsealed version to a sealed version, you should identify all of your customizations and move any inline customizations to extensions. To move the inline customizations, verify that you have all of the necessary extension points to do this. If you don't, submit an extensibility request.

#### NOTE

You might have to rewrite some of the POS inline changes that were completed in 7.1 when you upgrade to version 7.3 or higher.

APPLICATION VERSION	CRT SEALED	HWS SEALED	POS SEALED	DB SEALED	PROXY SEALED
Application release 8.1	Yes	Yes	Yes	Yes	Yes
Application release 8.0	Yes	Yes	Yes	Yes	Yes
Application 7.3	Yes	Yes	Yes	Yes	Yes
July 2017 release (Application 7.2)	Yes	Yes	Yes	Yes	No
Release 1611 (Application 7.1)	Yes	No	No	No	No
February 2016 release (Application 7.0)	No	No	No	No	No

## Upgrade the channel extension from 7.3 to a higher version

If you are completing a code upgrade, the following Retail SDK components must be upgraded. Each of these components are folders inside the Retail SDK.

#### NOTE

Code upgrade can be completed using a source control/merge tool. We recommend using a source control tool for this process so that you can track the changes and revert if required. If you are not using any source control, before you upgrade the code, be sure to make a back up of the old and new Retail SDK folder.

- **Assets:** If you have modified any of the following extension config files to include your custom assemblies, you must move those changes to the same config files in the new **Asset folder**.

- CommerceRuntime.Ext.config - CRT extensions.
- CommerceRuntime.MPOSOffline.Ext.config - CRT offline extensions.
- HardwareStation.Extension.config - Hardware station and payment extensions.
- RetailProxy.MPOSOffline.ext.config - Proxy extensions.
- **Build tools:** The **Build tools** folder contains the files related to packaging, build settings and pfx, and snk files. Merge or update the following files if you have made any changes in the older versions:
  - Customization.settings
  - Microsoft.Dynamics.RetailSdk.Build.props
  - Any custom pfx or snk files that you have created.
- **Database:** Copy and drop all your custom SQL scripts: ...\\RetailSDK\\Database\\Upgrade\\Custom
- **Online Store:** If you have an online store web project or online extension code, include it in this folder.

#### NOTE

Deployable package will not include online store code for packaging.

- **Payment externals:** Copy and paste all of your payment extension assemblies to the following **Payment assemblies** folders:
  - IPaymentDeviceAssemblies
  - IPaymentProcessorAssemblies
  - PaymentWebFiles
- **Payments:** Merge all of your payment extension projects into this folder.

#### NOTE

If you created a new payment extension project and did not modify anything in this folder, you can include it as part of the build and packaging process. Update **dirs.proj** in the **RetailSDK** folder to build your custom project. Next, update the **Customization.settings** file with the payment extension drop location so that during package creation your custom project is built. Make sure to include the extension as part of the packaging.

- **POS:** If you have any POS extensions, including the generated proxy, copy your POS extensions from ... \\RetailSDK\\POS\\Extensions to the new Retail SDK POS extension folder, *RetailSDK\\POS\\Extensions*. After you copy the extensions, update and merge the extension.json file inside the *POS\\Extensions* folder with the new POS extensions folder you copied.

For example, if you copied the **CustomExtension** folder, then you would update extension.json as shown below.

```
{
  "extensionPackages": [
    {
      "baseUr1": "SampleExtensions"
    },
    {
      "baseUr1": "SampleExtensions2"
    },
    {
      "baseUr1": "CustomExtension"
    }
  ]
}
```

You should also update `tsconfig.json` with the custom extension package so that the POS can compile the changes during build.

- **Reference:** Copy all of your extension output assemblies, such as **Commerce runtime**, **Hardware station**, **proxy**, and any external assemblies, to the reference folder. Include any assemblies that you want included as part of your deployment and packaging.
- **Commerce runtime (CRT) and Commerce Scale Unit extensions:** Copy all of your CRT and extension projects under the **Retail SDK** folder. Make sure to include the CRT and RS extension solution file details in the `dirs.proj` file under the **RetailSDK** folder so that during `msbuild`, all of the extension project is built and the output path for the project assemblies is set to the `RetailSDK\Reference` folder.
- **Hardware station (HWS) and payment extensions:** Copy all of your hardware station (HWS) and payment extension projects under the **Retail SDK** folder. Make sure to include the HWS and payment extension solution file details in the `dirs.proj` file under the **RetailSDK** folder so that during `msbuild`, all of the extension projects are built and the output path for the project assemblies is set to the `RetailSDK\Reference` folder.
- **Retail proxy extensions:** Copy all of your proxy extension projects under the **Retail SDK** folder. Make sure to include the proxy solution file details in the `dirs.proj` file under the **RetailSDK** folder so that during `msbuild`, all of the extension projects are built and the output path for the project assemblies is set to the `RetailSDK\Reference` folder.

#### NOTE

Run `msbuild` on the root of the Retail SDK folder. Use the Microsoft Visual Studio developer command-line tool to generate the deployable packages.

After you have upgraded all of the components, deploy the Commerce deployable packages, validate the deployment, and add the Retail SDK files to the repository.

## Upgrade the channel extension from 7.2 to a higher version

The steps mentioned in the previous section, **Upgrade the channel extension from 7.3 to higher versions**, will remain same for all the components except the Commerce proxy. In 7.2, you must have completed inline changes in the proxy project if you have CRT with RS extension and the typescript proxy was auto-generated based on the `customization.settings` file.

To upgrade your proxy to 7.3, complete the steps in the topic, [Typescript and C# proxies for Retail point of sale \(POS\)](#) and then move the proxy to the Retail SDK folder and then update the config file, `RetailProxy.MPOSOffline.ext.config`.

## Upgrade the channel extension from 7.1 to a higher version

In 7.1, you should have completed most of the POS and Proxy customizations inline. To upgrade to higher a application release, you should move all of your inline changes to extensions. If it is a binary hotfix upgrade, then you must perform a code merge with the new Retail SDK and then regenerate the package.

## Upgrade the channel extension from 7.0 to a higher version

In 7.0, you should have completed most of the customizations inline. To upgrade to higher a application release, you should move all of your inline changes to extensions. If it is binary hotfix upgrade, you must perform a code merge with the new Retail SDK and regenerate the package.

# Generate a deployable package for validation

Complete the steps in the topic, [Create deployable packages](#), to generate the deployable package for validation.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# What's new or changed in Finance and Operations apps home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

## Application releases

To see what's new or changed in each release of a Finance and Operations app, see the following topics:

- **Finance:** [What's new or changed in Dynamics 365 Finance](#)
- **Supply Chain Management:** [What's new or changed in Dynamics 365 Supply Chain Management](#)
- **Commerce:** [What's new or changed in Dynamics 365 Commerce](#)
- **Human Resources:** [What's new or changed in Dynamics 365 Human Resources](#)

## Platform updates

To see what's new or changed in the Platform updates for Finance and Operations apps, see the following topic:

- [What's new or changed in Platform updates](#)

## Lifecycle Services releases

To see what's new or changed in Lifecycle Services, see the following topic:

- [What's new or changed in Lifecycle Services \(LCS\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# What's new or changed in Dynamics 365 Commerce

2/18/2021 • 3 minutes to read • [Edit Online](#)

## Releases of Dynamics 365 Commerce

Dynamics 365 Commerce released to public preview in October 2019. To see what's new or changed in each release of Commerce, see the following topics.

VERSION	BUILD NUMBER	AUTO-UPDATE AVAILABILITY	LEARN MORE
10.0.16	10.0.689	February 2021	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.16</a>
10.0.15	10.0.644	January 2021	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.15</a>
10.0.14	10.0.605	November 2020	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.14</a>
10.0.13	10.0.569	October 2020	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.13</a>
10.0.12	10.0.507	August 2020	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.12</a>
10.0.11	10.0.464	July 2020	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.11</a>
10.0.10	10.0.420	May 2020	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.10</a>
10.0.9	10.0.383	April 2020	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.9</a>
10.0.8	10.0.319	February 2020	<a href="#">What's new or changed in Dynamics 365 Commerce version 10.0.8</a>

## Releases before February 2020

To see what's new or changed in releases before February 2020, see the following topics.

RELEASE	VERSION	BUILD NUMBER	AVAILABILITY	LEARN MORE
Microsoft Dynamics 365 Retail	10.0.7	10.0.283	January 2020	<a href="#">What's new or changed in Dynamics 365 Retail version 10.0.7</a>
Microsoft Dynamics 365 Retail	10.0.6	10.0.234	November 2019	<a href="#">What's new or changed in Dynamics 365 Retail 10.0.6</a>
Microsoft Dynamics 365 for Finance and Operations	10.0.5	10.0.197	October 2019	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 10.0.5 (October 2019)</a>
Microsoft Dynamics 365 for Finance and Operations	10.0.4	10.0.136	July 2019	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 10.0.4 (July 2019)</a>
Microsoft Dynamics 365 for Finance and Operations	10.0.3	10.0.107	June 2019	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 10.0.3 (June 2019)</a>
Microsoft Dynamics 365 for Finance and Operations	10.0.2	10.0.80	May 2019	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 10.0.2 (May 2019)</a>
Microsoft Dynamics 365 for Finance and Operations	10.0.1	10.0.51	April 2019	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 10.0.1 (April 2019)</a>
Microsoft Dynamics 365 for Finance and Operations	10.0	10.0.8	April 2019	<a href="#">What's new or changed in Finance and Operations version 10.0 (April 2019)</a>
Microsoft Dynamics 365 for Finance and Operations	8.1.3	8.1.227	January 2019	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 8.1.3 (January 2019)</a>

RELEASE	VERSION	BUILD NUMBER	AVAILABILITY	LEARN MORE
Microsoft Dynamics 365 for Finance and Operations	8.1.2	8.1.195	December 2018	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 8.1.2 (December 2018)</a>
Microsoft Dynamics 365 for Finance and Operations	8.1.1	8.1.170	October 2018	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 8.1.1 (October 2018)</a>
Microsoft Dynamics 365 for Finance and Operations	8.1	8.1.136	October 2018	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 8.1 (October 2018)</a>
Microsoft Dynamics 365 for Finance and Operations	8.0	8.0.30, 8.0.35	April 2018	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations version 8.0 (April 2018)</a>
Microsoft Dynamics 365 for Finance and Operations, Enterprise edition	7.3	7.3.11971.56116	December 2017	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations, Enterprise edition 7.3</a>
Microsoft Dynamics 365 for Finance and Operations, Enterprise edition	July 2017	7.2.11792.56024	June 2017	<a href="#">What's new or changed in Dynamics 365 for Finance and Operations, Enterprise edition (July 2017)</a>
Microsoft Dynamics 365 for Operations	1611	7.1.1541.3036	November 2016	<a href="#">What's new or changed in Dynamics 365 for Operations version 1611 (November 2016)</a>
Microsoft Dynamics AX	7.0.1	7.0.1265.23014	May 2016	<a href="#">What's new or changed in Dynamics AX application version 7.0.1 (May 2016)</a>
Microsoft Dynamics AX	7.0	7.0.1265.3015	February 2016	<a href="#">What's new or changed in Dynamics AX 7.0 (February 2016)</a>

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Preview features in Dynamics 365 Commerce 10.0.17 (April 2021)

2/18/2021 • 3 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic lists features that are either new or changed in Microsoft Dynamics 365 Commerce 10.0.17. This version has a build number of 10.0.761 and is available on the following schedule:

- **Preview of release:** February 2021
- **General availability of release (self-update):** March 2021
- **General availability of release (auto-update):** April 2021

## Features included in this release

The following features are included in this release. The feature titles link to additional information on the [Release plans](#) site. Additional links point to additional documentation that is currently available for that feature. Most of these features must be enabled using [Feature management](#) before you can use them.

- [B2B e-commerce](#)
  - For more information, see [Set up a B2B e-commerce site](#).
- [Edit partially fulfilled customer orders in POS](#)
  - For more information, see [Customer orders in Point of Sale \(POS\)](#).
- [Improved POS ordering and fulfillment experiences for serialized items](#)
  - For more information, see [Work with serialized items in the POS](#).
- [Shop similar descriptions](#)
  - For more information, see [Enable "shop similar descriptions" recommendations](#).
- [Configure module properties to be shown based on context](#)
  - For more information, see [Configure module properties to be shown based on context](#).
- [GeoLookup connector](#)
  - For more information, see [Configure and enable connectors](#).
- [Dynamic e-commerce pages based on URL parameters](#)
  - For more information, see [Create dynamic e-commerce pages based on URL parameters](#).
- [Quick view module](#)
  - For more information, see [Quick view module](#).
- [Clienteling improvements](#)
  - For more information, see [Clienteling overview](#).
- [Omnichannel support for incremental payment capture](#)
  - For more information, see [Incremental capture for back-office invoicing](#).
- [Refactored payment processing in storefront checkout](#)
- [Email receipt improvements and new features](#)
- [Exclusive threshold discounts compete with exclusive non-threshold periodic discounts](#)
- [Improved user experience for pickup order processing in point of sale](#)

- For more information, see [Process customer order pickups in POS](#).
- [Independent deployment and installation packages for MPOS, CPOS, HWS, and CSU extensions](#)
- [Simplified Commerce SDK update and developer experience](#)
- [Commerce localization for Brazil](#)
  - For more information, see [Commerce localization for Brazil](#).
- [Digital signing of retail transactions for France based on the fiscal integration framework](#)
- [Dynamics 365 Commerce and Microsoft Teams integration – add org structure, stores, workers](#)
- [Synergize task management between Dynamics 365 Commerce and Microsoft Teams](#)

## Additional resources

### Platform updates for Finance and Operations apps

Dynamics 365 Commerce 10.0.17 includes platform updates. To learn more, see [Platform updates for version 10.0.17 of Finance and Operations apps](#).

### Bug fixes

For information about the bug fixes that are included in this update, sign in to Lifecycle Services (LCS) and view the [KB article](#).

### Dynamics 365: 2021 release wave 1 plan

Wondering about upcoming and recently released capabilities in any of our business apps or platform?

Check out the [Dynamics 365: 2021 release wave 1 plan](#). We've captured all the details, end to end, top to bottom, in a single document that you can use for planning.

### Removed and deprecated features

The [Removed or deprecated features in Dynamics 365 Commerce](#) topic describes features that have been removed or deprecated for Commerce.

- A *removed* feature is no longer available in the product.
- A *deprecated* feature is not in active development and may be removed in a future update.

Before any feature is removed from the product, the deprecation notice will be announced in the [Removed or deprecated features in Dynamics 365 Commerce](#) topic 12 months prior to the removal.

For breaking changes that only affect compilation time, but are binary compatible with sandbox and production environments, the deprecation time will be less than 12 months. Typically, these are functional updates that need to be made to the compiler.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# What's new and changed in Dynamics 365 Commerce 10.0.16 (February 2021)

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic lists features that are either new or changed in Microsoft Dynamics 365 Commerce 10.0.16. This version has a build number of 10.0.689 and is available on the following schedule:

- **Preview of release:** November 2020
- **General availability of release (self-update):** January 2021
- **General availability of release (auto-update):** February 2021

## Features included in this release

The following features are included in this release. The feature titles link to additional information on the [Release plans](#) site. Additional links point to additional documentation that is currently available for that feature. Most of these features must be enabled using [Feature management](#) before you can use them.

- [Leverage the Dynamics 365 Commerce pricing engine to create a sales quote in Dynamics 365 Sales](#)
  - For more information, see [Use the Dynamics 365 Commerce pricing engine with Dynamics 365 Sales](#)
- [Customize transactional emails by mode of delivery](#)
  - For more information, see [Customize transactional emails by mode of delivery](#)
- [Upload and serve static files](#)
  - For more information, see [Upload and serve static files](#)
- [Clear system discounts from a sales line or sales transaction](#)
  - For more information, see [Online and offline point of sale \(POS\) operations](#)
- [Enable multiple pickup delivery modes for customer orders](#)
  - For more information, see [Enable multiple pickup delivery modes for customer orders](#)
- [Restrict affiliation and loyalty discounts in stores](#)
- [Gift card purchase in e-commerce](#)
- [Improved point of sale ordering and fulfillment experiences for serialized items](#)
  - For more information, see [Work with serialized items in the POS](#)

## Additional resources

### Platform updates for Finance and Operations apps

Dynamics 365 Commerce 10.0.16 includes platform updates. To learn more, see [Platform updates for version 10.0.16 of Finance and Operations apps](#).

### Bug fixes

For information about the bug fixes that are included in this update, sign in to Lifecycle Services (LCS) and view the [KB article](#).

### Dynamics 365: 2020 release wave 2 plan

Wondering about upcoming and recently released capabilities in any of our business apps or platform?

Check out the [Dynamics 365: 2020 release wave 2 plan](#). We've captured all the details, end to end, top to bottom, in a single document that you can use for planning.

### Removed and deprecated features



The [Removed or deprecated features in Dynamics 365 Commerce](#) topic describes features that have been removed or deprecated for Commerce.

- A *removed* feature is no longer available in the product.
- A *deprecated* feature is not in active development and may be removed in a future update.

Before any feature is removed from the product, the deprecation notice will be announced in the [Removed or deprecated features in Dynamics 365 Commerce](#) topic 12 months prior to the removal.

For breaking changes that only affect compilation time, but are binary compatible with sandbox and production environments, the deprecation time will be less than 12 months. Typically, these are functional updates that need to be made to the compiler.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# What's new or changed in Dynamics 365 Commerce 10.0.15 (January 2021)

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic lists features that are either new or changed in Microsoft Dynamics 365 Commerce 10.0.15. This version has a build number of 10.0.644 and is available on the following schedule:

- **Preview release:** October 2020
- **General availability (self-update):** November 2020
- **Auto-update:** January 2021

## Features included in this release

The following features are included in this release. The feature titles link to additional information on the [Release plans](#) site. Additional links point to additional documentation that is currently available for that feature. Most of these features must be enabled using [Feature management](#) before you can use them.

- [Curbside pickup](#)
  - For more information, see [Process credit cards without a hardware station](#).
- [Support for closing lines in purchase orders during receiving in POS](#)
- [Experimentation in Dynamics 365 Commerce](#)
  - For more information, see [Experimentation in Dynamics 365 Commerce](#).
- [Commerce localization for Brazil](#)
  - For more information, see [Commerce localization for Brazil](#).
- [Improved inbound and outbound inventory operations in store](#)
  - For more information, see [Inbound inventory operation in POS](#).
- [Improvements to the recall order operation in point of sale \(POS\)](#)
  - For more information, see [Recall order operation in POS](#).
- [Ability to skip "Change due" dialog in POS when no change is due](#)
- [CSP support for inline scripts and styles, and enabling nonce](#)
  - For more information, see [Manage Content Security Policy \(CSP\)](#).

## Additional resources

### Platform updates for Finance and Operations apps

Dynamics 365 Commerce 10.0.15 includes platform updates. To learn more, see [Platform updates for version 10.0.15 of Finance and Operations apps](#).

### Bug fixes

For information about the bug fixes that are included in this update, sign in to Lifecycle Services (LCS) and view the [KB article](#).

### Dynamics 365: 2020 release wave 2 plan

Wondering about upcoming and recently released capabilities in any of our business apps or platform?

Check out the [Dynamics 365: 2020 release wave 2 plan](#). We've captured all the details, end to end, top to bottom, in a single document that you can use for planning.

### Removed and deprecated features

The [Removed or deprecated features in Dynamics 365 Commerce](#) topic describes features that have been removed or deprecated for Commerce.

- A *removed* feature is no longer available in the product.
- A *deprecated* feature is not in active development and may be removed in a future update.

Before any feature is removed from the product, the deprecation notice will be announced in the [Removed or deprecated features in Dynamics 365 Commerce](#) topic 12 months prior to the removal.

For breaking changes that only affect compilation time, but are binary compatible with sandbox and production environments, the deprecation time will be less than 12 months. Typically, these are functional updates that need to be made to the compiler.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Removed or deprecated features in previous releases

2/18/2021 • 51 minutes to read • [Edit Online](#)

## NOTE

Effective November 2020:

- Common Data Service has been renamed to Microsoft Dataverse. For more information, see [Power Automate Blog](#).
- Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. For more information, see [Terminology updates](#).

This topic will be updated soon to reflect the latest terminology.

## IMPORTANT

This topic is no longer updated. To see a current list of features that have been removed or deprecated from Finance and Operations apps, search for "**Removed or deprecated features**" content that relates to the app you're using.

This topic describes features that have been removed or deprecated from Dynamics 365 for Finance and Operations and previous releases of that product.

- A *removed* feature is no longer available in the product.
- A *deprecated* feature is not in active development and may be removed in a future update.

This list is intended to help you consider these removals and deprecations for your own planning.

Detailed information about objects in Finance and Operations apps can be found in the [Technical reference reports](#). You can compare the different versions of these reports to learn about objects that have changed or been removed in each version of Finance and Operations apps.

## Finance 10.0.7 with Platform update 31

### Chinese voucher types without Account groups selection

Reason for deprecation/removal	Changed to the feature with account groups selection.
Replaced by another feature?	Yes
Product areas affected	Application
Deployment option	All
Status	Deprecated: By December 1, 2020, we plan to no longer support Chinese voucher types setup without Account groups selection. Find more details about new feature design in What's new in 10.0.7

# Finance and Operations 10.0.6 with Platform update 30

## DimensionHash.getHash(str\_message)

Reason for deprecation/removal	Windows is deprecating the use of SHA1, as documented in <a href="#">Windows Enforcement of SHA1 Certificates</a> .
Replaced by another feature?	Yes
Product areas affected	Application
Deployment option	All
Status	Deprecated: By April 1, 2020, developers must use the platform APIs found in the class <b>HasFunction</b> .

## Hash.ComputeSHA1Hash(string message)

Reason for deprecation/removal	Windows is deprecating the use of SHA1, as documented in <a href="#">Windows Enforcement of SHA1 Certificates</a> .
Replaced by another feature?	Yes
Product areas affected	Platform
Deployment option	All
Status	Deprecated: By April 1, 2020, developers must use the platform APIs found in the class <b>HasFunction</b> .

## FormDateTimeControl.setUtcString()

Reason for deprecation/removal	We are retiring the <b>setUtcString()</b> method, because a better replacement method is available.
Replaced by another feature?	Yes
Product areas affected	Platform
Deployment option	All
Status	Deprecated: By October 1, 2020, we plan to no longer support the <b>setUtcString()</b> method. Developers should be using the <b>setUtcDateTime()</b> method instead.

Reason for deprecation/removal	Not legally required.
Replaced by another feature?	No
Product areas affected	Italian localization
Deployment option	All
Status	Deprecated: By October 1, 2020, we plan to no longer support the <b>Blacklist report (IT) – Feature reference IT-00001</b> .

### Domestic tax report – Feature reference IT-00003

Reason for deprecation/removal	Not legally required.
Replaced by another feature?	No
Product areas affected	Italian localization
Deployment option	All
Status	Deprecated: By October 1, 2020, we plan to no longer support the <b>Domestic tax report – Feature reference IT-00003</b> .

## Finance and Operations 10.0.5 with Platform update 29

### US Payroll tax updates

Reason for deprecation/removal	We are retiring tax updates for the US Payroll functionality due to low usage and enhanced functionality that is now offered via strategic integrations.
Replaced by another feature?	Yes
Product areas affected	Payroll
Deployment option	All
Status	Deprecated: By July 31, 2024, we plan to no longer provide tax updates to US Payroll customers. The functionality will remain in the product, but enhancements will no longer keep the functionality up to date, and any product defects will be evaluated on a case-by-case basis.

**NOTE**

This represents a change from the original discontinuation date of October 1, 2021. For more information, see [Tax updates being retired for US Payroll feature in Microsoft Dynamics 365 for Finance and Operations](#).

**Data management staging clean up**

Reason for deprecation/removal	Does not meet the core requirements that are needed for scheduling periodic cleanup.
Replaced by another feature?	Yes, the Job history cleanup feature is being added to meet the scenarios holistically.
Product areas affected	Data management
Deployment option	All
Status	Deprecated: Target timeframe for the functionality to be removed is December 2020.

**Finance and Operations 10.0.4 with Platform update 28****France: FEC Accounting data export in XML**

Reason for deprecation/removal	Replaced by TXT format, <b>French FEC audit file</b> is available through <b>General ledger &gt; Periodic tasks &gt; Data export</b> .
Replaced by another feature?	Yes
Product areas affected	General ledger
Deployment option	All
Status	Deprecated. Target timeframe for the functionality to be removed is July 2020.

**Legacy navigation bar**

Reason for deprecation/removal	Header alignment with other Dynamics and Office products. For more details, see <a href="#">Updated navigation bar that aligns with the Office header</a> .
Replaced by another feature?	Starting in Platform update 24, a restyled navigation bar that features search was introduced.
Product areas affected	Web client

Deployment option	All
Status	Deprecated: Starting in April 2020, the legacy navigation bar will no longer be available. Until that point, customers can revert to the legacy navigation bar through the <b>Client performance options</b> page.

## Finance and Operations 10.0.2 with Platform update 26

### Legacy default action behavior

Reason for deprecation/removal	The legacy behavior for default actions in grids results in an unexpected column having the default action link after grid columns have been reordered via personalization. The new sticky default action feature corrects this. For more details, see <a href="#">Sticky default actions in grids</a> .
Replaced by another feature?	Starting in Platform update 21, a feature for "sticky default actions" was introduced. This feature can be enabled on the <b>Client performance options</b> page.
Product areas affected	Grids in the web client
Deployment option	All
Status	Deprecated: Starting in April 2020, sticky default actions will be the default behavior, without a mechanism to revert to the legacy behavior.

### Legacy "is one of" filtering experience

Reason for deprecation/removal	The "is one of" filtering experience went through a redesign in Platform update 22, with the plan for this to eventually be the only "is one of" filtering experience.
Replaced by another feature?	Starting in Platform update 22, an improved "is one of" filtering experience became available on the <b>Client performance options</b> page. For more information, see <a href="#">Optimized is one of filtering experience</a> .
Product areas affected	Web client
Deployment option	All
Status	Deprecated: Starting in April 2020, the improved "is one of" experience will be the default behavior, without a mechanism to revert to the legacy behavior.

### Parameter to enable sales orders with multiple project contract funding sources

Support for creating project-based sales orders where the project contract has multiple funding sources is



enabled with the **Project management parameters** setting **Allow sales orders for project with multiple funding sources**. By default, this parameter is not enabled.

<b>Reason for deprecation/removal</b>	The functionality will always be enabled after the parameter is removed.
<b>Replaced by another feature?</b>	No. The functionality to support project-based sales orders with multiple funding sources will always be enabled.
<b>Product areas affected</b>	The <b>Allow sales orders for projects with multiple funding sources</b> parameter will be removed. The following methods will be modified when the parameter is removed: <b>ctrlSalesOrderTable</b> method in <b>ProjStatusType</b> class, <b>validate</b> method for <b>ProjId</b> field, and <b>run</b> method in <b>SalescreateOrder</b> form. The following methods will be deprecated when the parameter is removed: <b>IsSalesOrderAllowedForMultipleFundingSources</b> in <b>ProjTable</b> table file, <b>IsAllowSalesOrdersForMultipleFundingSourcesParam Enabled</b> method in <b>ProjTable</b> table file, <b>AllowSalesOrdersForMultipleFundingSources</b> data field in <b>ProjParameters</b> form and <b>ProjParameterEntity</b> files, <b>IsAssociatedToMultipleFundingSourcesContract</b> private method in <b>ProjTable</b> table file.
<b>Deployment option</b>	All
<b>Status</b>	Deprecation is planned for the April 2020 release wave.

#### Legacy workflow reports for tracking and instance status

<b>Reason for deprecation/removal</b>	The legacy workflow reports for tracking and instance status are being deprecated because they are no longer referenced from the navigation. The report names are <b>WorkflowWorkflowInstanceByStatusReport</b> and <b>WorkflowWorkflowTrackingReport</b> .
<b>Replaced by another feature?</b>	The workflow history form can be used instead.
<b>Product areas affected</b>	Web client
<b>Deployment option</b>	All
<b>Status</b>	Deprecated: Target timeframe for the functionality to be removed is April 2020.

## Finance and Operations 10.0.1 with Platform update 25

### Deprecated APIs and potential breaking changes

Deriving from internal classes is deprecated

<b>Reason for deprecation/removal</b>	Before Platform update 25, it was possible to create a class or table that derives from an internal class/table that is defined in another package/module. This is not a safe coding practice. As of Platform update 25, the compiler will display a warning.
<b>Replaced by another feature?</b>	The compiler warning will be replaced by an error in Platform update 26. This change is backward compatible at runtime, which means that Platform update 25 or newer can be deployed on any sandbox or production environment without the need to modify custom code. This change only affects development and compile time.
<b>Product areas affected</b>	Visual Studio development tools
<b>Deployment option</b>	All
<b>Status</b>	Deprecated: The warning will become a compilation error in Platform update 26.

#### Overriding internal methods is deprecated

<b>Reason for deprecation/removal</b>	Before Platform update 25, it was possible to override an internal method in a derived class that is defined in another package/module. This is not a safe coding practice. As of Platform update 25, the compiler will display a warning.
<b>Replaced by another feature?</b>	This warning will be replaced by a compile error in Platform update 26. This change is backward compatible at runtime, which means that Platform update 25 or newer can be deployed on any sandbox or production environment without the need to modify custom code. This change only affects development and compile time.
<b>Product areas affected</b>	Visual Studio development tools
<b>Deployment option</b>	All
<b>Status</b>	Deprecated: The warning will become a compilation error in Platform update 26.

## Finance and Operations 10.0.0 with Platform update 24

### Renaming released products

<b>Reason for deprecation/removal</b>	When you use the <b>Rename primary key</b> function to change the ItemId of a released product, only direct foreign key references are updated. Any other references to the released product, such as from production orders, will retain the old ItemId. As a result, there could be inconsistent data that will eventually block business processes.

Replaced by another feature?	No.
Product areas affected	Product information management
Deployment option	All
Status	Removed as of Finance and Operations 10.0.0 with Platform update 24.

## Finance and Operations 8.1.3 with Platform update 23

### SQL Server Reporting Services ReportViewer Control

Customers can use the **Export** action provided by the embedded SQL Server Reporting Services (SSRS) ReportViewer control to download documents produced by Finance and Operations applications. This HTML-based presentation of the report offers users a non-paginated preview of the document.

Reason for deprecation/removal	The non-paginated nature of the HTML-based preview experience does <b>not</b> deliver fidelity with the physical documents ultimately produced by Finance and Operations. By fully embracing PDF as the standard format for business documents, users are able to take advantage of a modern viewing experience with improved performance when producing application reports.
Replaced by another feature?	Going forward, PDF documents will be the default format for reports rendered by Finance and Operations.
Product areas affected	This change does <b>not</b> impact customer scenarios where reports are distributed electronically or sent directly to printers.
Deployment option	All
Status	Deprecated: A removal date has not been set for this feature. The functionality to automatically preview application reports using an embedded PDF viewer is planned for the May 2019 Platform update.

### Client KPI controls

Embedded key performance indicators (KPIs) could be modeled in Visual Studio by a developer and further customized by the end user.

Reason for deprecation/removal	The native client controls used to define KPIs have low customer uptake and rely on a developer to add trackable metrics.

Replaced by another feature?	PowerBI.com service delivers world-class tooling for defining and managing KPIs based on data from external sources. In an upcoming release, we plan to enable you to embed solutions hosted on PowerBI.com in application workspaces.
Product areas affected	This update will prevent developers from introducing new KPI controls in Visual Studio designer.
Deployment option	All
Status	Deprecated: A removal date has not been set for this feature.

## Deprecated APIs and future breaking changes

### Field groups containing invalid field references

Reason for deprecation/removal	<p>It is possible for table metadata definitions to have field groups containing invalid field references. If deployed, this can cause runtime failures in Financial Reporting and SQL Server Reporting Services (SSRS). This issue is currently categorized as a <i>compiler warning</i> rather than an <i>error</i>, meaning that the deployable package creation and deployment can proceed without fixing the issue. To fix this issue:</p> <ol style="list-style-type: none"> <li>1. Remove the invalid field reference from the table field group definition.</li> <li>2. Recompile.</li> <li>3. Ensure any warnings or errors are addressed.</li> </ol>
Replaced by another feature?	This warning will be replaced by a compile error in the future.
Product areas affected	Visual Studio development tools
Deployment option	All
Status	Deprecated: The warning is a compile-time error with platform updates for version 10.0.11 of Finance and Operations apps.

#### Complete list

To access the full list of APIs that are being deprecated, see [Deprecation of methods and metadata elements](#).

## Finance and Operations 8.1 with Platform update 20

### Batch transfer rules for subledger journal account entries

The Synchronous transfer mode is being deprecated in the General ledger parameters. This mode is replaced by Asynchronous and scheduled batch only, which already exist as options for transfer. For additional information, see the [General Ledger Parameters – Batch transfer rules](#) blog.

<b>Reason for deprecation/removal</b>	We are removing the synchronous option due to performance impact to the system.
<b>Replaced by another feature?</b>	Asynchronous and scheduled batch are options to use in place of Synchronous.
<b>Product areas affected</b>	General Ledger, Accounts payable, Accounts Receivable, Procurement, Expense
<b>Deployment option</b>	All
<b>Status</b>	Deprecated: Target timeframe for the functionality to be removed is the 10.0 version.

### **Electronic reporting for Russia**

Feature for configuring .txt and .xml file formats of declarations.

<b>Reason for deprecation/removal</b>	Replaced with Electronic reporting.
<b>Replaced by another feature?</b>	Yes.
<b>Product areas affected</b>	General Ledger
<b>Deployment option</b>	All
<b>Status</b>	Removed as of Finance and Operations 8.1 with Platform update 20.

### **Financial reports generator for Russia**

A tool for setting up data collection for accounting and tax reports, and to export data to XLS and DOC report templates. Functional parts: Export data to XLS and DOC report templates, queries, fixed requisites are removed.

<b>Reason for deprecation/removal</b>	Removed parts are replaced with Electronic reporting.
<b>Replaced by another feature?</b>	Yes. Financial reports setup user interface should be used for setting up data collection rules by GL accounts or tax registers. Export data to various file types, fixed requisites and query-like data collection rules should be configured in Electronic reporting.
<b>Product areas affected</b>	General ledger.
<b>Deployment option</b>	All
<b>Status</b>	Removed as of Finance and Operations 8.1 with Platform update 20.

### **Integration with external providers for sending electronic reporting through communication channels for**

## Russia

Feature exporting generated electronic files of declarations to folder for further sending to official providers of electronic reporting as well as importing state back.

Reason for deprecation/removal	Replaced with electronic messages configurable feature.
Replaced by another feature?	Yes.
Product areas affected	General Ledger, Tax
Deployment option	All
Status	Removed as of Finance and Operations 8.1 with Platform update 20.

### Profit tax register wizard

Feature for creating templates for new profit tax registers. This feature creates X++ objects for new registers, which are then created as templates with the appropriate calculation logic added in.

Reason for deprecation/removal	Feature is not compatible with the Finance and Operations extensibility model.
Replaced by another feature?	No
Product areas affected	Tax
Deployment option	All
Status	Removed as of Finance and Operations 8.1 with Platform update 20.

## Finance and Operations 8.0 with Platform update 15

No features have been removed or deprecated with this release. Platform update 15 is cumulative and contains new or changed features from Platform update 13, Platform update 14, and Platform update 15.

## Finance and Operations, Enterprise edition 7.3 with Platform update 12

### Personalized product recommendations

Starting February 15, 2018, retailers will no longer be able to display personalized product recommendations on a point of sale (POS) device. For more information, see [Product recommendations overview](#).

Reason for deprecation/removal	We are removing the current version of the product recommendation service as we redesign this feature with a better algorithm and newer retail-oriented capabilities.

Replaced by another feature?	No. However, after Spring 2018, we plan to bring back this feature to leverage a new recommendation service.
Product areas affected	Personalized product recommendations in POS.
Deployment option	All
Status	Removed as of February 15, 2018. This affects customers running Dynamics 365 for Operations 1611 and later.

### Extension of the list of Electronic reporting (ER) functions

The possibility to introduce custom functions to be used in the ER expression builder (for more information, see [Extend the list of Electronic reporting \(ER\) functions](#)) is not supported any more. Due to changes of the ER APIs, the API to call built-in functions from the ER expression builder became internal and can't be extended any longer.

Reason for deprecation/removal	Code sealing initiative
Replaced by another feature?	<p>None. Whenever a new built-in function is needed, a new extension request must be addressed to the ER framework team.</p> <p>As a temporary work around while the requested function is under development by the ER team, the required logic can be programmed as a method of a custom application class. This method can be accessed in an ER expression as a property of the added ER data source of the <b>Application\Class</b> type that refers to that custom application class.</p>
Product areas affected	Electronic reporting framework
Deployment option	All
Status	Removed as of Finance and Operations, Enterprise edition 7.3.

### Inventory by item group and Inventory by inventory dimension aging reports

These two reports are no longer supported in Finance and Operations. Instead, the **Inventory aging** report can be used to improve the user experience.

Reason for deprecation	Duplicate functionality
Replaced by another feature?	Yes. The two reports have been replaced by the <b>Inventory aging</b> report.
Product areas affected	Inventory management, Cost management

Deployment option	All
Status	Deprecated: The menu items for the two reports have been removed in version 7.3. However, the code for the reports remains in the product. The plan is to remove the code in a future release.

### Power BI content packs available on AppSource

The **Cost management**, **Financial performance**, and **Retail channel performance** content packs, available on the [Microsoft AppSource](#) site, are deprecated as a consequence of product updates in Microsoft Power BI. System administration forms used to deploy these content packs to PowerBI.com are also being deprecated in Finance and Operations.

Reason for deprecation/removal	Product updates in Microsoft Power BI.
Replaced by another feature?	The <b>Cost management</b> , <b>Financial performance</b> , and <b>Retail channel performance</b> content packs, available on the <a href="#">AppSource</a> site, are being replaced by analytical applications which allow for solution integrations at the database level. For more information about analytical applications, see <a href="#">Embedded Power BI in workspaces</a> .
Product areas affected	Cost management, Finance, and Retail
Deployment option	Cloud only (Integration with PowerBI.com is not supported in on-premises deployments.)
Status	Deprecated: Target timeframe for the functionality removal is Q2 2018.

### Standard UI in data management workspace

The standard UI in data management is the legacy UI, which is the default UI presented to the users when they visit the data management workspace.

Reason for deprecation/removal	We are investing in providing new user experiences in the new UI.
Replaced by another feature?	The new UI called <i>Enhanced views</i> is replacing the old UI.
Product areas affected	Data management workspace
Deployment option	All
Status	Deprecated: Target timeframe for the functionality to be removed is Q2 2018.

### Excise, Sales Tax, Service Tax for India

These taxes have been subsumed into Indian GST.



Reason for removal or deprecation	These taxes have been subsumed into Indian GST.
Replaced by another feature?	Indian GST
Product areas affected	Tax
Deployment option	All modules
Status	Deprecated: A removal date has not been set for this feature.

#### **File Validation Utility (FVU) for India**

Reason for removal or deprecation	Lack of customer usage
Replaced by another feature?	No
Product areas affected	Indian withholding tax
Deployment option	All modules
Status	Deprecated: A removal date has not been set for this feature.

#### **TDS/TCS certificate for India**

Users can download this from the government portal.

Reason for removal or deprecation	Lack of customer usage
Replaced by another feature?	No
Product areas affected	Indian withholding tax
Deployment option	All modules
Status	Deprecated: A removal date has not been set for this feature.

#### **Export/import (EXIM) incentive scheme for India**

Reason for removal or deprecation	Lack of customer usage
Replaced by another feature?	No
Product areas affected	Import and export

Deployment option	All modules
Status	Deprecated: A removal date has not been set for this feature.

## Dynamics 365 for Retail 7.2

### Personalized product recommendations

Starting February 15, 2018, retailers will no longer be able to display personalized product recommendations on a point of sale (POS) device. For more information, see [Product recommendations overview](#).

Reason for deprecation/removal	We are removing the current version of the product recommendation service as we redesign this feature with a better algorithm and newer retail-oriented capabilities.
Replaced by another feature?	No. However, after Spring 2018, we plan to bring back this feature to leverage a new recommendation service.
Product areas affected	Personalized product recommendations in POS.
Deployment option	All
Status	Removed as of February 15, 2018. This affects customers running Dynamics 365 for Retail 7.2 and later.

## Finance and Operations, Enterprise edition July 2017 with Platform update 8

### Currency conversion for accounting and reporting currencies

Currency conversion for accounting and reporting currencies was introduced when the euro was introduced.

Reason for deprecation/removal	Limited usage and addition of the Copy legal entity functionality as a replacement.
Replaced by another feature?	No, but the Copy legal entity and Configurations features were added to make it easier to move to a company that has changing core requirements.
Product areas affected	Financial management
Status	Deprecated: A removal date has not been set for this feature.

### Warehouse mobile devices portal

Warehouse mobile devices portal (WMDP) was a standalone component that was intended for on-premises self-deployment. This component is no longer supported in Finance and Operations. A native app that improves the user experience has replaced the functionality of WMDP.

Reason for deprecation/removal	Duplicate functionality.
Replaced by another feature?	Yes. This feature has been replaced by Finance and Operations - Warehousing. For more information about setup and prerequisites, see <a href="#">Install and configure the Warehousing app overview</a> .
Product areas affected	Warehouse management, Transportation management
Deployment option	Warehouse mobile devices portal (WMDP) was a standalone component that was intended for on-premises self-deployment.
Status	Deprecated: Target timeframe for the functionality to be removed is Q4 2019.

### Advanced bank reconciliation matching rule for manual matching

A matching rule was used to select and mark a bank document when documents were manually matched in the reconciliation worksheet.

Reason for deprecation/removal	Limited usage.
Replaced by another feature?	No. Column filtering capabilities should be used to find documents for reconciliation.
Product areas affected	Cash and bank management
Deployment option	All
Status	Removed as of July 2017.

## Dynamics 365 for Operations 1611 with Platform update 3

### AEB payment formats for Spain

The Consejo Superior Bancario payment formats were used to send remittance files to the bank for customer payments and vendor payments. The content of these formats was determined by the Asociación Española de Banca. It covers Cuaderno 19, 32, 58, 34.

Reason for deprecation/removal	The payment formats are no longer used.
Replaced by another feature?	Yes, ISO20022 Credit transfer and Direct debit payment formats for Spain
Product areas affected	Accounts payable, Accounts receivable

<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Bank payments transfer for Lithuania

Bank payment transfers were generated and printed by using the Payment transfer (LT) export format for Lithuania. The Lithuanian market began to use LITAS, the unified electronic banking system, in 2005.

<b>Reason for deprecation/removal</b>	The payment formats are no longer used.
<b>Replaced by another feature?</b>	Yes, ISO20022 Credit transfer payment format for Lithuania
<b>Product areas affected</b>	Accounts payable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### BBS Direkte Remitting payment formats for Norway

BBS Direkte Remitting payment formats include customer payment collection export (direct debit) and return message import.

<b>Reason for deprecation/removal</b>	The payment formats are no longer used.
<b>Replaced by another feature?</b>	The AvtaleGiro customer payment format for Norway can be used to generate direct debit messages. Return message import will be implemented in future releases.
<b>Product areas affected</b>	Accounts payable, Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Chart of Accounts tool for Spain

This tool is used when a chart of accounts in Spain requires major changes. Users can import a new chart of accounts in Microsoft Excel or text format, and can also import financial statements.

<b>Reason for deprecation/removal</b>	Limited usage
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	General ledger
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Dom80 payment format for Belgium

Legacy Belgian payment format for payment collection (direct debit).

Reason for deprecation/removal	The payment format is no longer used.
Replaced by another feature?	Yes, ISO 20022 Direct debit payment format for Belgium
Product areas affected	Accounts receivable
Status	Deprecated: A removal date has not been set for this feature.

#### DTA/EZAG payment formats for Switzerland

DTA/EZAG formats are integrated into the ESR system, because they can carry on the reference number. Because the reference number isn't mandatory, these formats can be used to process any vendor payments. These formats are used by companies that have a bank account in a location other than "Postfinance."

Reason for deprecation/removal	The payment formats are no longer used.
Replaced by another feature?	Yes, ISO20022 Credit transfer payment format for Switzerland
Product areas affected	Accounts payable
Status	Deprecated: A removal date has not been set for this feature.

#### EDIFACT-DIRDEB payment format for Austria

EDIFACT-DIRDEB payment format for payment collection (direct debit).

Reason for deprecation/removal	The payment format is no longer used.
Replaced by another feature?	Yes, ISO 20022 Direct debit payment format for Austria
Product areas affected	Accounts receivable
Status	Deprecated: A removal date has not been set for this feature.

#### EDIVAT for Belgium

EDIVAT is an obsolete Belgian standard for electronic declaration via secure mail. Dynamics AX 2012 retains the read-only solution to enable access to the historical data.

Reason for deprecation/removal	The functionality is no longer used.
Replaced by another feature?	No

<b>Product areas affected</b>	General ledger
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### **eGiro EDIFACT CREMUL payment import format for Norway**

eGiro is based on the international UN EDIFACT CREMUL (Multiple Credit Advice Message) standard that is used for automatic posting of customer payments. In Dynamics AX, eGiro is implemented as a customer payment import format.

<b>Reason for deprecation/removal</b>	The payment format is no longer used.
<b>Replaced by another feature?</b>	Yes, the ISO20022 Camt.054 notification import.
<b>Product areas affected</b>	Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### **External inventory for Poland**

Evidence of goods that are taken from a vendor for sales without purchase. Goods that are handled in external inventory don't affect standard inventory, and can be sold and then purchased automatically. This process creates real inventory movements.

<b>Reason for deprecation/removal</b>	Replaced by another feature
<b>Replaced by another feature?</b>	Yes, the core Inbound consignment functionality
<b>Product areas affected</b>	Accounts payable, Inventory management
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### **Financial reports generator for Eastern Europe**

A tool is used to set up data collection for accounting and tax reports, and to export data to XLS and DOC report templates.

<b>Reason for deprecation/removal</b>	Limited usage
<b>Replaced by another feature?</b>	No. The tool will be replaced by Electronic reporting configurations in future releases.
<b>Product areas affected</b>	General Ledger

<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Import of customer payment transactions for Finland

You can select an import format for Finnish payments to import customer payment transactions from an external file that the bank provides.

<b>Reason for deprecation/removal</b>	The payment format is no longer used.
<b>Replaced by another feature?</b>	Yes, the ISO20022 Camt.054 notification import.
<b>Product areas affected</b>	Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Import of payment transactions into a general ledger journal for Finland

A format that is specific to Finland is used to import accounting transactions into the general ledger.

<b>Reason for deprecation/removal</b>	The payment format is no longer used.
<b>Replaced by another feature?</b>	Yes, the ISO20022 Camt.053 bank statement import using Advanced Bank Reconciliation.
<b>Product areas affected</b>	Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Integration with Isabel synchronized (CIS) for Belgium

Isabel is the framework for electronic banking in Europe and is a de-facto standard in Belgium.

<b>Reason for deprecation/removal</b>	Integration with Isabel client has been discontinued.
<b>Replaced by another feature?</b>	No. The payment formats that are no longer used are replaced by ISO20022 Credit transfer payment format for Belgium.
<b>Product areas affected</b>	Accounts payable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Modifications in the chart of accounts and accounting rules for Spain

This feature is used for changes in the chart of accounts and accounting rules in Spain. It maps accounts to help

transform the old chart of accounts into the new chart of accounts, and compares the previous fiscal year with the new fiscal year, even if they were posted to different account numbers.

Reason for deprecation/removal	Limited usage
Replaced by another feature?	No
Product areas affected	General ledger
Status	Deprecated: A removal date has not been set for this feature.

#### **Pagamento Fornitori vendor payment format**

Legacy Italian payment format for credit transfers.

Reason for deprecation/removal	The payment format is no longer used.
Replaced by another feature?	Yes, ISO20022 Credit transfer payment format for Italy
Product areas affected	Accounts payable
Status	Deprecated: A removal date has not been set for this feature.

#### **Payment export formats for Estonia**

The Telehansa and Teleservice formats are used for bank payment export.

Reason for deprecation/removal	The payment formats are no longer used.
Replaced by another feature?	Yes, ISO20022 Credit transfer payment format for Estonia
Product areas affected	Accounts payable
Status	Deprecated: A removal date has not been set for this feature.

#### **Payment file archive for Norway**

When payment files are generated, the file archive automatically archives all files that are created, even files that were previously written or read.

Reason for deprecation/removal	Replaced by another feature
Replaced by another feature?	Yes, Electronic reporting archived jobs



<b>Product areas affected</b>	Accounts payable, Accounts receivable, Organization administration
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Payment import formats for Estonia

The Telehansa and TeleTeenus formats are used for bank payment import.

<b>Reason for deprecation/removal</b>	The payment formats are no longer used.
<b>Replaced by another feature?</b>	Yes, the ISO20022 Camt.054 bank notification import.
<b>Product areas affected</b>	Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Payroll information in Human Resources

Human Resources Payroll information

<b>Reason for deprecation/removal</b>	This functionality has been replaced by core Payroll and Human Resources pages.
<b>Replaced by another feature?</b>	<b>Benefits, Earnings</b> , and other related pages that were previously in US Payroll have been reconfigured, and are now part of the core Human Resources configuration to help support external payroll processing. This functionality is accessed by using the <b>Human Resources 1 &gt; Payroll</b> configuration key.
<b>Product areas affected</b>	Human Resources, Payroll
<b>Status</b>	Removed as of Dynamics 365 for Operations version 1611.

### Performance management goal workflow

Performance management includes goal management and integration with performance reviews.

<b>Reason for deprecation/removal</b>	Performance management was redesigned, and the number of goal pages was reduced to simplify the process.
<b>Replaced by another feature?</b>	No. Goals are visible to managers through the Manager Self Service portal, and can be changed and viewed by the manager.
<b>Product areas affected</b>	Human capital management

<b>Status</b>	Removed as of Dynamics 365 for Operations version 1611.

### Postgirot and Postgirot Utland payment formats for Sweden

Postgirot and Postgirot Utland payment formats for Sweden.

<b>Reason for deprecation/removal</b>	The payment formats are no longer used.
<b>Replaced by another feature?</b>	Yes, ISO20022 Credit transfer payment format for Sweden
<b>Product areas affected</b>	Accounts payable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Radio frequency identifier

Radio Frequency Identification (RFID) is a data-collection technology that uses electronic tags to store identification data and a no-line-of-sight requirement reader to capture the identification data.

<b>Reason for deprecation/removal</b>	Low customer usage and a limited feature set.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Inventory management
<b>Status</b>	Removed as of Dynamics 365 for Operations 1611.

### Report about state invoices numbering for Latvia

Latvian legislation provides specific rules about the numbering of sales invoices. The functionality lets you assign specific numbers to sales invoices, based on the user or user group. You can then generate a report or an XML file. You can also print a report about invoice numbers that are used.

<b>Reason for deprecation/removal</b>	The state invoice numbering no longer has to be maintained. The report about used invoice numbers is no longer required.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Set up the names of the manager and general accountant of a company for Lithuania

The names of the manager and the general accountant of a company can be specified in the company

information and used in different local report printouts.

<b>Reason for deprecation/removal</b>	Replaced by another feature
<b>Replaced by another feature?</b>	Yes, the setup of officials can be used for the same purpose.
<b>Product areas affected</b>	Accounts payable, Accounts receivable, Cash and bank management
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Shipping carrier interface

<b>Reason for deprecation/removal</b>	Duplicate functionality
<b>Replaced by another feature?</b>	Partially replaced by Transportation management
<b>Product areas affected</b>	Sales and marketing, Inventory management
<b>Status</b>	Removed as of Dynamics 365 for Operations version 1611.

### Telepay payment formats for Norway

Telepay payment formats include vendor payment export (credit transfer) and customer payment collection (direct debit).

<b>Reason for deprecation/removal</b>	The payment formats are no longer used.
<b>Replaced by another feature?</b>	Yes, ISO20022 Credit transfer payment format and AvtaleGiro customer payment format for Norway, as well as pain.002 and camt.054 bank notification return files import.
<b>Product areas affected</b>	Accounts payable, Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Vendor payment export formats for Finland

Two formats for exporting payments are available for Finland. LM02 (FI) is used for domestic payments, and LUM2 (FI) is used for foreign payments.

<b>Reason for deprecation/removal</b>	The payment formats are no longer used.
<b>Replaced by another feature?</b>	Yes, ISO20022 Credit transfer payment format for Finland

<b>Product areas affected</b>	Accounts payable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Warehouse management II

<b>Reason for deprecation/removal</b>	The Warehouse management II solution (WMS II) that was available in the <b>Inventory management</b> module duplicates functionality that is in the <b>Warehouse management</b> module that was released in Dynamics AX 2012 R3.
<b>Replaced by another feature?</b>	The <b>Warehouse management</b> module that was released in AX 2012 R3, Dynamics AX 2012 R3 CU8, and Dynamics AX 2012 R3 CU9 replaces the Warehouse management II features. The new module has more advanced features and more flexible warehouse management processes than Warehouse management II.
<b>Product areas affected</b>	Inventory management, Sales and marketing, Procurement and sourcing
<b>Status</b>	Removed as of Dynamics 365 for Operations version 1611.

### Worker reminders in Human Resources

Human Resources Payroll information

<b>Reason for deprecation/removal</b>	Low usage
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Human resources
<b>Status</b>	Removed as of Dynamics 365 for Operations version 1611

### Workflow for creating goals

A workflow for managing the creation of employee goals is one of several workflows that were available to help coordinate the performance management process.

<b>Reason for deprecation/removal</b>	Performance management has been completely redesigned in Finance and Operations.

Replaced by another feature?	The redesigned Performance management feature gives more control over the content of the goals, the measurements that are used to track progress, and the attachment of supporting documentation. Goals can be stored as templates and then reused. This feature can help you set up additional goals for your employees more quickly.
Product areas affected	Human capital management
Status	Removed as of Dynamics 365 for Operations version 1611.

## Dynamics AX 7.0

### Ability to cancel changes to a vendor invoice

Reason for deprecation/removal	Performance enhancement
Replaced by another feature?	No
Product areas affected	Accounts payable
Status	Removed as of Dynamics AX 7.0.

### AIF, AxD, and AxBC integrations

In Application Integration Framework (AIF), data can be exchanged with external systems through business logic that is exposed as services. Dynamics AX includes services that are based on documents and .NET Business Connector (AxBC). A document is created by using XML. The XML includes header information that is added to create a *message* that can be transferred into or out of Dynamics AX. Examples of documents include sales orders and purchase orders. However, almost any entity, such as a customer, can be represented by a document. Services that are based on documents use the `Axd <Document>` classes.

Reason for deprecation/removal	The architecture of AIF and AxDs could not be scaled to a cloud service. There were performance issues around bulk import.
Replaced by another feature?	This feature is replaced by the Data Import/Export framework, which supports recurring bulk import/export. For AxBC, we recommend that you use the actual tables.
Product areas affected	AxDs, AxBCs, and AIF
Status	Removed as of Dynamics AX 7.0.

### Billing code rate scripts

Billing scripts were used to calculate billing rates for billing codes. This scripts required custom development in the C Sharp or Visual Basic programming language. In the current version of Dynamics AX, the **billing code rate scripts** are not supported.

<b>Reason for deprecation/removal</b>	The support for the custom C Sharp or Visual Basic scripts was not added in Dynamics AX 7.0.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Public sector, Accounts receivable
<b>Status</b>	Removed as of Dynamics AX 7.0.

### **BOMs without BOM versions**

When the **BOM versions** configuration key was disabled, bill of materials (BOM) versions were hidden in all forms, and the system forced a 1:1 relationship between released products and BOMs. In the current version of Dynamics AX, the **BOM versions** configuration key can't be disabled.

<b>Reason for deprecation/removal</b>	Using a configuration key to control BOM versions doesn't scale in a cloud environment.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Product information management, Inventory management
<b>Status</b>	Removed as of Dynamics AX 7.0.

### **Brazilian Bordero**

Specific method of payment for Brazilian companies

<b>Reason for deprecation/removal</b>	Support for the Brazilian Bordero method of payment has been discontinued from Brazilian localization
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Accounts payable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### **Brazilian Sintegra statement**

Federal tax statement for ICMS tax

<b>Reason for deprecation/removal</b>	This statement is no longer applicable in some Brazilian states.
<b>Replaced by another feature?</b>	No. Users can use Generic Electronic reporting tool to configure the statement if required under specific situations.

<b>Product areas affected</b>	Fiscal books
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### **Brazilian SCAN contingency mode for NF-e**

(SCAN) contingency environment is used to generate, export, and import the status of a Nota Fiscal eletrônica (NF-e) when the environment of Secretaria da Fazenda (SEFAZ) is not available.

<b>Reason for deprecation/removal</b>	This method of contingency is no longer applicable in all Brazilian states
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Accounts receivable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### **Business Analyzer**

This mobile application let users review key business metrics.

<b>Reason for deprecation/removal</b>	This functionality has been replaced by another feature.
<b>Replaced by another feature?</b>	The Monitor financial performance content pack for Microsoft Power BI will include key financial metrics that were previously available in Business Analyzer.
<b>Product areas affected</b>	General ledger
<b>Status</b>	Deprecated: The use of Business Analyzer has been deprecated.

### **Business statistics**

The setup of business statistics inquiries that can help you analyze the performance of the organization

<b>Reason for deprecation/removal</b>	Legacy approach to business intelligence (BI), low customer usage, and a limited feature set
<b>Replaced by another feature?</b>	New BI solutions for the current version of Dynamics AX
<b>Product areas affected</b>	Procurement and sourcing, Accounts payable, Sales and marketing, Accounts receivable
<b>Status</b>	Removed as of Dynamics AX 7.0.

### Change document date function in Invoice approval journal

Reason for deprecation/removal	Low usage
Replaced by another feature?	Yes. The document date on the posted vendor transaction can be changed.
Product areas affected	Accounts payable
Status	Removed as of Dynamics AX 7.0.

### ClieOp03 payment format for the Netherlands

Reason for deprecation/removal	The format is no longer applicable in the Netherlands, because it has been replaced by SEPA functionality.
Replaced by another feature?	SEPA payments export
Product areas affected	All modules
Status	Deprecated: A removal date has not been set for this feature.

### Compliance Center

The Compliance Center was an Enterprise Portal site for managing the documentation requirements for compliance initiatives that are related to the Sarbanes-Oxley law.

Reason for deprecation/removal	Lack of customer usage. Microsoft SharePoint includes the same capability that was available in the Compliance Center.
Replaced by another feature?	No
Product areas affected	Compliance and internal controls
Status	Removed as of Dynamics AX 7.0.

### Connector for Microsoft Dynamics

This tool was used to integrate key data from Microsoft Dynamics CRM to Dynamics ERP applications.

Reason for deprecation/removal	This functionality has been replaced by another feature.
Replaced by another feature?	Dataverse
Product areas affected	Connector for Dynamics



Status	Removed as of Dynamics AX 7.0.

### Container unit and multi dimension on-hand

Reason for deprecation/removal	Duplicate functionality
Replaced by another feature?	Yes. Since AX 2012, this functionality has been replaced by the consolidated batch orders feature set. This feature set includes the consolidated on-hand view.
Product areas affected	Product information management, Production control, Inventory management, Sales and marketing
Status	Removed as of Dynamics AX 7.0.

### Cue group metadata

Reason for deprecation/removal	Cue groups were used to display one or more Cues in the FactBox area. There was limited uptake, and there were also performance concerns, because a record change in a parent form caused one query per Cue in the Cue group.
Replaced by another feature?	No
Product areas affected	All modules
Status	Removed as of Dynamics AX 7.0.

### Cue metadata

Reason for deprecation/removal	Cue metadata was limited to count or sum information.
Replaced by another feature?	Tile metadata was introduced to provide more flexibility for modeling. For example, you can model current counts, navigation, and key performance indicators (KPIs). Count tile metadata is the direct replacement of the Cue metadata.
Product areas affected	All modules
Status	Removed as of Dynamics AX 7.0

### Danish check format

--	--

<b>Reason for deprecation/removal</b>	Support for the Danish check format layout has been discontinued, and the report has been removed from DK localization.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	All modules
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Data partitions

Data partitions provide a logical separation of data in the Dynamics AX database.

<b>Reason for deprecation/removal</b>	Data partitions were introduced in Dynamics AX 2012 R2 to enable data isolation. In a common scenario, a company has subsidiaries, and the data from one subsidiary should not be visible to another subsidiary, even though both subsidiaries are managed by the same IT department. However, extra scripts and management overhead throughout the program were required in order to create new partitions and populate them with data, and to back up partition data. In the cloud, where we have access to platform as a service (PaaS) database services (Microsoft Azure SQL Database), it's much more efficient to use a database as the isolation container than to do isolation in the program. Regardless of whether data partitioning is required for subsidiaries, for multiple tenants, or just for scale, we believe that the scenarios can be handled better through multiple instances of Finance and Operations.
<b>Replaced by another feature?</b>	Customers using data partitions must use multiple instances of Finance and Operations if database level separation is a critical issue.
<b>Product areas affected</b>	All modules
<b>Status</b>	Removed as of Dynamics AX 7.0.

### Database and file share storage for attachments

Dynamics AX 2012 allowed storage of attachments in the database and in file shares. Both of those options are no longer supported.

--	--

<b>Reason for deprecation/removal</b>	Files share storage is no longer supported because cloud-hosted environments cannot communicate with local file shares. Database storage has been deprecated in favor of Azure Blob storage. Azure Blob storage is equivalent to storage in the database, as documents can only be accessed through Finance and Operations client forms. This provides the added benefit of providing storage that doesn't negatively affect the performance of the database. Blob storage is the default storage mechanism for Document Management and works immediately.
<b>Replaced by another feature?</b>	Database storage has been deprecated in favor of Azure Blob storage.
<b>Product areas affected</b>	All modules
<b>Status</b>	Removed as of Dynamics AX 7.0.

#### **Delimitation**

<b>Reason for deprecation/removal</b>	No use of the functionality was found.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Time and attendance
<b>Status</b>	Removed as of Dynamics AX 7.0.

#### **Desktop client**

<b>Reason for deprecation/removal</b>	The Dynamics AX client experience has been redesigned to improve usability across multiple platforms and devices.
<b>Replaced by another feature?</b>	The new web client is based on the desktop Form metadata and programming model that have been modified to provide a rich web platform.
<b>Product areas affected</b>	All modules
<b>Status</b>	Removed as of Dynamics AX 7.0.

#### **Direct database connection**

In Dynamics AX 2012 R3, Retail Modern POS could connect directly to the Channel DB in similar fashion to Enterprise POS. This was in addition to the standard communication method of Retail Modern POS communicating through Retail Server.

--	--

<b>Reason for deprecation/removal</b>	Direct database connectivity required lower security protocols and was primarily used to achieve the highest levels of performance. Due to the performance and security enhancements that have occurred in Finance and Operations, this functionality now causes more issues than it solves.
<b>Replaced by another feature?</b>	No. Only standard Retail Server communication is now supported.
<b>Product areas affected</b>	Channel DB/Retail Modern POS
<b>Status</b>	Removed as of Dynamics AX 7.0.

#### **Dutch SWIFT MT940**

<b>Reason for deprecation/removal</b>	Generic functionality is now used instead of localized functionality.
<b>Replaced by another feature?</b>	Yes, this functionality has been replaced by Advanced bank reconciliation functionality.
<b>Product areas affected</b>	All modules
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

#### **eBilanz (XBRL for Germany)**

This functionality provided eXtensible Business Reporting Language (XBRL) output that is intended specifically for the German eBilanz taxonomy.

<b>Reason for deprecation/removal</b>	Lack of customer usage
<b>Replaced by another feature?</b>	This feature hasn't been replaced by another feature, but multiple specialized XBRL packages that provide rich XBRL functionality are available for the German market.
<b>Product areas affected</b>	Management Reporter
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

#### **Enterprise Portal client**

<b>Reason for deprecation/removal</b>	A single client platform has been provided.

Replaced by another feature?	The new web client is based on the desktop form metadata and programming model that have been modified to provide a rich web platform.
Product areas affected	All modules
Status	Removed as of Dynamics AX 7.0.

### Environmental sustainability

Reason for deprecation/removal	Low customer usage and a limited feature set
Replaced by another feature?	No
Product areas affected	Compliance and internal controls, Accounts payable
Status	Removed as of Dynamics AX 7.0.

### Form ActiveX and Managed Host controls

Reason for deprecation/removal	The ActiveX and Managed Host controls are based on the deprecated desktop client.
Replaced by another feature?	The extensible control framework supports building new controls that are based on HTML, CSS, and JavaScript, and is a first-class control in the Microsoft Visual Studio Tooling environment.
Product areas affected	All modules
Status	Removed as of Dynamics AX 7.0.

### Generate prenotes by using a batch

Prenote generation can't be done by using a batch, but it can still be done by a user.

Reason for deprecation/removal	No form exists to persist and display the resulting prenote file when it's generated by using a batch.
Replaced by another feature?	Prenotes can still be generated, and the user has control over the location where the file is saved.
Product areas affected	Accounts payable, Accounts receivable, Cash and bank management
Status	Removed as of AX 7.0.

### German DTAUS payment export and account statement import (totals and transactions)

<b>Reason for deprecation/removal</b>	The format is no longer applicable in Germany, because it has been replaced by Single Euro Payments Area (SEPA) functionality.
<b>Replaced by another feature?</b>	Yes, this functionality has been replaced by SEPA payment export and advanced bank reconciliation functionality for importing account statements.
<b>Product areas affected</b>	All modules
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

#### **German DTAZV payment format in domestic Currency**

<b>Reason for deprecation/removal</b>	The format is no longer applicable in Germany, because it has been replaced by SEPA functionality.
<b>Replaced by another feature?</b>	SEPA payments export
<b>Product areas affected</b>	Accounts payable
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

#### **German MT940 import**

<b>Reason for deprecation/removal</b>	Generic functionality is now used instead of localized functionality.
<b>Replaced by another feature?</b>	Yes, this functionality has been replaced by Advanced bank reconciliation functionality.
<b>Product areas affected</b>	All modules
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

#### **German XML EU Sales list**

<b>Reason for deprecation/removal</b>	The XML format for German EU Sales List reporting is no longer supported. Only the ELMA5 text file format can be used to submit the EU Sales List report to the German Tax Office.
<b>Replaced by another feature?</b>	No

Product areas affected	Tax
Status	Deprecated: A removal date has not been set for this feature.

### GL SSRS reports

Reports that include the following menu items have been removed: **Summary trial balance**, **Detailed trial balance**, **Chart of accounts**, **Audit trail**, **Balances**, and **Balance list**.

Reason for deprecation/removal	Financial Microsoft SQL Server Reporting Services (SSRS) reports have been replaced by Management Reporter capabilities and default reports.
Replaced by another feature?	Management Reporter (labeled <b>Financial reporting</b> in the current version of Dynamics AX)
Product areas affected	General ledger
Status	Removed as of Dynamics AX 7.0.

### InfoPart and FormPart metadata

Reason for deprecation/removal	InfoPart and FormPart metadata enabled the creation of FactBoxes for two different clients.
Replaced by another feature?	InfoPart metadata, which was a simplified form definition, is converted into a Form by upgrade tooling. FormPart metadata, which referenced a Form, is replaced by a more direct reference that is created by upgrade tooling.
Product areas affected	All modules
Status	Removed as of Dynamics AX 7.0.

### Main account list page

A list of accounts for the legal entity and related balance information

Reason for deprecation/removal	Balance information is available on the <b>Trial balance</b> list page by account and dimension.
Replaced by another feature?	<b>Main accounts</b> contains the same list of accounts that the <b>Main account</b> list page contained. The grid view in <b>Main accounts</b> also shows an even smaller, grid-like view.
Product areas affected	General ledger

<b>Status</b>	Removed as of Dynamics AX 7.0.

### Malaysia and Singapore bank cash flow report

This feature let the user print a cash flow report that shows transactions and details of the cash inflows and outflows for a specific date range for selected bank accounts.

<b>Reason for deprecation/removal</b>	The same information can be obtained from the Inquiry bank transaction.
<b>Replaced by another feature?</b>	The Inquiry bank transaction
<b>Product areas affected</b>	Cash and bank management
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Mexican CFD electronic invoice

This feature enabled the generation of Mexican electronic invoices by using the Comprobante Fiscal Digital (CFD) method, where the company signs the invoice by requesting the related authorization from the government. This feature also provides a monthly report that includes all electronics invoices that were issued in the period.

<b>Reason for deprecation/removal</b>	The method is no longer applicable. The generation of electronic invoices by using the CFD method was deprecated by the tax authorities and replaced by the Comprobante Fiscal Digital a través de Internet (CFDI) method, where the signing is delegated to the third-party provider (PAC). The monthly report has been removed, and an inquiry option lets users inquire about historical transactions.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Account receivables, Project
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Mexico realized and unrealized VAT

Dynamics AX 2012 managed unrealized value-added tax (VAT) by using Mexico-specific functionality for unrealized tax.

<b>Reason for deprecation/removal</b>	Duplicate functionality
<b>Replaced by another feature?</b>	Yes, this functionality has been replaced by standard conditional sales tax functionality that is provided by Core.



<b>Product areas affected</b>	Tax
<b>Status</b>	Deprecated: A removal date has not been set for this feature.

### Microsoft Outlook integration

<b>Reason for deprecation/removal</b>	This functionality has been replaced by Microsoft Exchange Server integration.
<b>Replaced by another feature?</b>	Yes
<b>Product areas affected</b>	Sales and marketing
<b>Status</b>	Removed as of Dynamics AX 7.0.

### Private blocking of inventory and warehouse management journals

The inventory and warehouse journals no longer support the ability to mark a journal as private for a selected user. Only the process of blocking journals as private for user groups and blocking during editing is supported.

<b>Reason for deprecation/removal</b>	No use of the functionality was found.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Inventory management
<b>Status</b>	Removed as of Dynamics AX 7.0.

### Product builder

Product builder was used to dynamically configure items from a sales order, purchase order, production order, sales quotation, project quotation, or item requirement. Based on a product model that had modeling variables, the user could select values to meet the customer requirements and get a unique product variant that had a BOM and route.

<b>Reason for deprecation/removal</b>	Product builder exposed X++ code to end users and isn't supported in the current version of Dynamics AX. It has been removed to avoid duplicate maintenance efforts on overlapping, sizeable codebases.
<b>Replaced by another feature?</b>	Yes. The constraint-based configuration was introduced in Dynamics AX 2012 where the deprecation of Product builder in future versions was already announced. The constraint-based configuration technology is selected on the product masters to enable the configuration. To learn more, see <a href="#">Product configuration overview</a> .

<b>Product areas affected</b>	Product information management, Sales and marketing
<b>Status</b>	Removed as of Dynamics AX 7.0.

### Production Floor app

This is the app for tablet devices running Windows 8.1 RT and Windows 8.1 Pro.

<b>Reason for deprecation/removal</b>	With the change to a web-based client, it is possible to deliver similar functionality through the native Dynamics AX 7.0 client. The Job Card Device provides a production floor user interface that is optimized for touch and tablet form factors.
<b>Replaced by another feature?</b>	Yes. The Job Card Device, which is a native part of Dynamics AX 7.0.
<b>Product areas affected</b>	Production control
<b>Status</b>	Deprecated: A removal date from the Microsoft store has not yet been set for this feature.

### Rename product dimension

This feature let you change the name of one of the three standard product dimensions (size, color, or style) to a name that better suited your business requirements. Renaming included all the labels where the product dimension name was used.

<b>Reason for deprecation/removal</b>	The current version of Dynamics AX doesn't support label changes at run time.
<b>Replaced by another feature?</b>	No
<b>Product areas affected</b>	Product information management
<b>Status</b>	Removed as of Dynamics AX 7.0.

### Retail Server connectivity using HTTP

In Dynamics AX 2012 R3, the Retail Server could function using HTTP communication (non-secured). This was in addition to the standard communication using HTTPS.

<b>Reason for deprecation/removal</b>	Due to new security requirements, only secured communication using TLS 1.2 (or above, as available) is now supported. The self-service installer will automatically configure the computer for this communication.
<b>Replaced by another feature?</b>	No. Only standard HTTPS communication is now supported.

Product areas affected	Retail Server
Status	Removed as of Dynamics AX 7.0.

### Role Center pages

Reason for deprecation/removal	Role Center pages were built on the deprecated Enterprise Portal platform, which has been replaced by the new web client platform in the current version of Dynamics AX.
Replaced by another feature?	The new Workspace form pattern provides users with a process-centered design that provides easy access to commonly used tasks within that process.
Product areas affected	All modules
Status	Removed as of Dynamics AX 7.0

### Sales tax jurisdictions

Reason for deprecation/removal	Low customer usage and a limited feature set
Replaced by another feature?	No
Product areas affected	US sales tax
Status	Removed as of Dynamics AX 7.0.

### Sites Services

Sites Services let you build websites that extend your business processes to the Internet without IT support.

Reason for deprecation/removal	The Microsoft Azure infrastructure that is used by Dynamics AX has new capabilities that can be used instead (for example, Azure sites).
Replaced by another feature?	No
Product areas affected	HR recruiting, Case management, Request for quotes, Vendor registration, Collaborative workspaces for opportunities and campaigns
Status	Removed as of Dynamics AX 7.0.

### SSAS demand forecasting strategy

<b>Reason for deprecation/removal</b>	The design of the feature cannot be supported in the new cloud architecture.
<b>Replaced by another feature?</b>	Azure Machine Learning demand forecasting strategy
<b>Product areas affected</b>	Master planning
<b>Status</b>	Removed as of Dynamics AX 7.0.

#### Vendor invoice pool excluding posting details

<b>Reason for deprecation/removal</b>	Low usage. This functionality has been replaced by the Invoice journal that has workflow functionality.
<b>Replaced by another feature?</b>	Workflow capabilities of the Invoice journal.
<b>Product areas affected</b>	Accounts payable
<b>Status</b>	Removed as of Dynamics AX 7.0.

#### Virtual company accounts

The virtual companies feature is no longer supported in Dynamics AX. The virtual companies feature let users set up tables that could be shared by a set of companies. For a description of the feature, see [Company accounts and Virtual company accounts](#). The feature works by grouping tables into collections that are assigned to virtual companies, which are groups of existing "real" companies. Queries are created so that all the companies in the virtual company can access the data in the tables of the associated table collections.

<b>Reason for deprecation/removal</b>	<ul style="list-style-type: none"> <li>- Virtual companies must be set up before data is stored in the tables. Retrofitting virtual companies onto an existing implementation is very difficult.</li> <li>- Because there has been so much data normalization in the current version of Dynamics AX, it has become difficult to know what to add to the table collections. For example, it's difficult to know which tables to share. All the tables referenced from tables that are in a virtual company must also added. Because of table normalization, even simple master data that is spread across multiple tables must be part of the virtual company. Any mistake that is made here will cause functional issues.</li> <li>- When a table is part of a virtual company, it loses information about the origin of the data, and only the virtual company is recorded.</li> </ul>
<b>Replaced by another feature?</b>	Global tables can be used to make tables accessible from all companies. Currently, there is no replacement.
<b>Product areas affected</b>	All modules

Status	Removed as of Dynamics AX 7.0.

### Windows 8 tablet app

The Windows 8 tablet app provided functionality for expense entry and approval.

Reason for deprecation/removal	Finance and Operations is compatible with tablets. The tablet app is no longer required.
Replaced by another feature?	No.
Product areas affected	Expense management
Status	Removed: This functionality is only available for Dynamics AX 2012 R3.

### Workplanner

Reason for deprecation/removal	Low usage
Replaced by another feature?	No, but the <b>Profile relation</b> page, which is opened from the <b>Profile groups</b> page, supports the same business scenario as the deprecated <b>Workplanner</b> page.
Product areas affected	Time and attendance
Status	The code has not been removed. However, the form, JmgWorkPlanner, was not migrated.

### X++ financial statements

Reason for deprecation/removal	This functionality has been replaced by another feature.
Replaced by another feature?	Management Reporter (labeled <b>Financial reporting</b> in the current version of Dynamics AX)
Product areas affected	General ledger
Status	Removed as of Dynamics AX 2012

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Removed or deprecated features in Dynamics 365 Commerce

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes features that have been removed, or that are planned for removal from Dynamics 365 Commerce.

- A *removed* feature is no longer available in the product.
- A *deprecated* feature is not in active development and may be removed in a future update.

This list is intended to help you consider these removals and deprecations for your own planning.

## NOTE

Detailed information about objects in Finance and Operations apps can be found in the [Technical reference reports](#). You can compare the different versions of these reports to learn about objects that have changed or been removed in each version of Finance and Operations apps.

## Features removed or deprecated in the Commerce 10.0.17 release

### IMPORTANT

Version 10.0.17 is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

### Full dataset generation interval is deprecated

Reason for deprecation/removal	Beginning in this release, in the <b>Commerce scheduler parameters</b> form in Dynamics 365 headquarters, the <b>Full dataset generation interval in days</b> field will be deprecated. Also starting in this release, the field will be visually removed so that the value cannot be edited. This will stay as the value <b>0</b> .
Replaced by another feature?	No
Product areas affected	Dynamics 365 Commerce
Deployment option	All
Status	Deprecated. Do not use this field or change the value in it.

## Features removed or deprecated in the Commerce 10.0.15 release

### Internet Explorer 11 support for Dynamics 365 is deprecated

Reason for deprecation/removal	<p>Effective December 2020, Microsoft Internet Explorer 11 support for all Dynamics 365 products is deprecated, and Internet Explorer 11 won't be supported after August 2021.</p> <p>This will impact customers who use Dynamics 365 products that are designed to be used through an Internet Explorer 11 interface. After August 2021, Internet Explorer 11 won't be supported for such Dynamics 365 products.</p>
Replaced by another feature?	We recommend that customers transition to Microsoft Edge.
Product areas affected	All Dynamics 365 products
Deployment option	All
Status	Deprecated. Internet Explorer 11 won't be supported after August 2021.

## Features removed or deprecated in the Commerce 10.0.11 release

### Data action hooks

Reason for deprecation/removal	The data action hooks feature has been deprecated due to performance issues.
Replaced by another feature?	We recommend using <a href="#">data action overrides</a> to modify business logic in the data action layer.
Product areas affected	e-Commerce extensibility data actions
Deployment option	All
Status	Deprecated: As of release 10.0.11

### Retail SDK support for Visual Studio 2015, msbuild 14.0, and Retail SDK\Reference libraries and tools

Reason for deprecation/removal	Retail SDK support for Visual Studio 2015 has been deprecated and updated to support VS 2017, msbuild 15.0 and all the reference libraries and commerce proxy generator tools in the RetailSDK\References folder moved to NuGet packages to simplify the extension model and SDK upgrade process.
Replaced by another feature?	We recommend that you follow the information in <a href="#">Migrate the Retail SDK from Visual Studio 2015 to Visual Studio 2017</a> to update your system.
Product areas affected	Retail SDK extensions
Deployment option	All

Status	Deprecated: As of release 10.0.11
--------	-----------------------------------

### Retail Server Extension using IEdmModelExtender and CommerceController

Reason for deprecation/removal	Retail server extension using IEdmModelExtender and CommerceController has been deprecated to provide simplified extension model. The new implementation will have only the controller class without any additional IEdmModelExtender class implementation. This also avoids the dependency with a particular OData version (if the OData version is updated it may break extensions.)
Replaced by another feature?	We recommend that you use the IController class extension model by importing the NuGet (Microsoft.Dynamics.Commerce.Hosting.Contracts) package.
Product areas affected	Retail server extensions
Deployment option	All
Status	Deprecated: As of release 10.0.11

### Hardware station Extension using IHardwareStationController

Reason for deprecation/removal	Hardware station extension using IHardwareStationController has been deprecated to provide simplified extension model. The new implementation will have only the IController class without any additional class implementation and to avoid the dependency with core hardware station libraries, previously extension need to refer multiple libraries.)
Replaced by another feature?	It is recommended to use the IController class extension model by importing the NuGet (Microsoft.Dynamics.Commerce.Hosting.Contracts) package.
Product areas affected	Hardware station extensions
Deployment option	All
Status	Deprecated: As of release 10.0.11

## Features removed or deprecated in the Commerce 10.0.10 release

### POS operation 803 - Picking and receiving

Reason for deprecation/removal	Picking and receiving operations is being deprecated due to new operation redesign.
--------------------------------	---



Replaced by another feature?	Yes. It is replaced by two new POS operations: inbound operation (804) and outbound operation (805).
Product areas affected	Point of sale (POS) application
Deployment option	All
Status	Deprecated: As of release 10.0.10, the picking and receiving operation will no longer receive any new feature updates. Only critical bug fixes will be done for this operation in future releases. All customers are encouraged to move to the new <a href="#">Inbound operations</a> and <a href="#">Outbound operations</a> , which will continue to be part of our long-term product roadmap.

## Features removed or deprecated in the Commerce 10.0.7 release

### Commerce GetProductAvailabilities and GetAvailableInventoryNearby API's

Reason for deprecation/removal	New optimized APIs have been created to replace the GetProductAvailabilities and GetAvailableInventoryNearby APIs.
Replaced by another feature?	Yes: It is replaced by GetEstimatedAvailability and GetEstimatedProductWarehouseAvailability APIs.
Product areas affected	e-Commerce application SDK
Deployment option	All
Status	Deprecated: As of release 10.0.7, there will no longer be engineering investments made for GetProductAvailabilities and GetAvailableInventoryNearby. Organizations that use these APIs in their e-Commerce deployments should convert to the new GetEstimatedAvailability and GetEstimatedProductWarehouseAvailability APIs and enable the <a href="#">Optimized product availability calculation feature</a> .

## Previous announcements about removed or deprecated features

To learn more about features that have been removed or deprecated in previous releases, see [Removed or deprecated features in previous releases](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# AX 2012 features that were postponed

2/18/2021 • 11 minutes to read • [Edit Online](#)

This topic lists features of Microsoft Dynamics AX 2012 that were postponed. These features weren't implemented in Microsoft Dynamics AX 7.0. In the following table, the **Current status** column indicates whether the feature has been implemented since the AX 7.0 release.

For a detailed list of the release date for each version, see [Software lifecycle policy and cloud releases](#).

AX 2012 FEATURE THAT WAS POSTPONED	DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)
Absence management in Human resources	Functionality for entering absence transactions isn't included. Additionally, functionality for approving absence transactions as a manager isn't included. Setup capabilities that are required for integration with other modules are available through the <b>Human Resources 2</b> configuration key.	Implemented in Dynamics 365 Human Resources
Alerts	Alerts help users keep track of data changes in the system.	Implemented in Platform update 15
Bank payment order for Latvia and Lithuania	You can print a payment order for Latvia and Lithuania. This feature will be available in a future update.	Not implemented
Bankgirot AP return format for Sweden	The Bankgirot return format is used to import bank return messages. This feature will be available in a future update.	Not implemented
Client drag-and-drop	The web client controls have application programming interfaces (APIs) for drag-and-drop operations, but these APIs are based on the deprecated desktop client technology and they require a redesign so that they work on the new web client platform. APIs that support drag-and-drop operations will be reviewed for inclusion in a future update.	Not implemented
Client right-to-left (RTL) layout	RTL layout is now supported.	Implemented in Platform update 2

AX 2012 FEATURE THAT WAS POSTPONED	DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)
Cost accounting	The <b>Cost accounting</b> module is designed to meet the requirements of internal costs and profitability reports at multiple organizational levels. To define the cost object level, the module depends on a correct mapping of financial dimensions. The module lets you perform advanced allocations of cost origin from expenditures that are registered in the general ledger or budget. It also lets you compare realized costs and budgeted costs.	Implemented in version 1611
Customer self-service (CSS)	CSS lets you create approved customer records. It also allows users to view selected product catalogs, order items, and view the status of invoices. Additionally, CSS lets you create and follow return orders.	Not implemented
Customizable help topics	The ability to create customized help topics has not yet been implemented. Custom task guides and custom field help are available. This feature will be available in a future update.	Not implemented
Employee self-service (ESS)	ESS shows employees several tiles that have task-related and career-related information on a single page. Employees can view pending work items and click links that open pages where they can take action on their tasks. ESS pages also show employees the status of their certifications, when their next performance reviews are scheduled, skills, goals, and compensation information, and other information, such as balances for vacation and sick time. Employees can also access a company directory from their ESS page.	Implemented in version 1611
External questionnaire and recruiting functionality	Functionality for externally posting questionnaires and open jobs will be added to Human Resources in a future update.	External questionnaire functionality hasn't been implemented. Recruiting functionality is available in Microsoft Dynamics 365 Talent: Attract.
Fiscal printers for Poland	Integration with Polish fiscal printers enables the required information to be sent to the fiscal printer in the correct format during invoice posting. Examples of Polish fiscal printers include the Posnet Thermal and Elzab Omega printer types. This feature will be available in a future update.	Not implemented

AX 2012 FEATURE THAT WAS POSTPONED	DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)
General budget reservations	The General budget reservations document is sometimes referred to as a commitment. Public sector entities often use this document to set aside or earmark budgeted funds so that they aren't available for other purposes.	Implemented in version 8.1
<b>Graphics</b> tab on the <b>Fixed asset value model</b> and <b>Depreciation book profile</b> pages	The chart shows the depreciation, accumulated depreciation, and net book value over time. Users can click the <b>Data</b> tab to view more detailed information than the chart shows. This chart will be redesigned in a future update.	Not implemented
Intelligent Data Management Framework (IDMF)	IDMF is an add-on tool that lets system administrators optimize performance. IDMF assesses the health of the application, analyzes current usage patterns, and helps reduce database size.	Not implemented
Microsoft Project client integration	The Microsoft Project client is integrated with projects.	Implemented in version 7.2 (July 2017 update)
Procurement site	In previous versions, the Employee self-service procurement site lets you enter requisitions for employees, view the status of an order (created, received, or receipt confirmed), and request onboarding of a new vendor. You could configure different procurement catalogs to show on the site depending on policy. You could also design procurement catalogs by adding new nodes. In the current version, procurement catalog capabilities are reduced and are used only to limit the products that can be ordered for an organization. The structure is always based on the Procurement categories hierarchy. Additionally, on the procurement site the employee could approve a vendor invoice and confirm receipts in relation to the requisitions and derived purchase orders.	Not implemented
Secure global address book	The ability to help secure the global address book by legal entity and address book is not available. This feature will be available in a future update.	Not implemented
Specifications for Electronic reporting (ER) payment formats	Currently, you must enter the payment format specifications manually. In a future update, you will be able to select payment format specifications in	Not implemented

AX 2012 FEATURE THAT WAS POSTPONED

a list. The following payment specifications are currently supported per payment format.

CURRENT STATUS (AS OF FEBRUARY 2019)

[!NOTE] Values for these supported payment specifications are used as payment specification parameters on the **Payment specification** page for a selected method of payment.

#### BTL91 for the Netherlands

PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION
ChqBen	Cheque, Begunstigde
ChqOff	Cheque, Kantoor opdrachtgever
ChqPri	Cheque, Opdrachtgever
TrfBenBen	Overboeking Begunstigde/Begunstigde
TrfBenBenUrg	Overboeking Begunstigde/Begunstigde Spoed
TrfEurBen	Overboeking Euro/Begunstigde
TrfEurBenUrg	Overboeking Euro/Begunstigde Spoed
TrfEurEur	Overboeking Euro/Euro
TrfEurEurUrg	Overboeking Euro/Euro Spoed
TrfForBen	Overboeking VV-rekening/Begunstigde
TrfForBenUrg	Overboeking VV-rekening/Begunstigde Spoed
TrfForFor	Overboeking VV-rekening/VV-rekening

AX 2012 FEATURE THAT WAS POSTPONED

PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION
------------------------------------	---------------------------

CURRENT STATUS (AS OF FEBRUARY 2019)

TrfForForUrg	Overboeking VV-rekening/VV-rekening Speed
--------------	---

**Betalingservice for Denmark**

PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION
B0112	BS-B 0112: Lang tekst & adresse
B0113	BS-B 0113: Erstat. bet. lang tekst
T0112	BS-T 0112: Lang tekst & adresse
T0117	BS-T 0117:FK;kort frist;lang tekst&adr.

**Nordea vendor for Denmark**

PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION
56	Currency account transfer between Nordea accounts in Denmark
47	Domestic check
45	Domestic transfer
50	Express transfer
55	Intercompany transfer (domestic)
51	Intercompany transfer to a foreign bank
54	International check
52	Nordea intercompany payment

AX 2012 FEATURE THAT WAS POSTPONED

PAYMENT SPECIFICATION (USED IN ER)

EXPORT FORMAT DESCRIPTION

CURRENT STATUS (AS OF FEBRUARY 2019)

43

Request for transfer

46

Transfer form/giro payment

**ISO20022 Credit transfer (CH)**

PAYMENT SPECIFICATION (USED IN ER)

EXPORT FORMAT DESCRIPTION

Tp1.ESROPS

Type 1 - ESR orange payment slip

Tp21.ISR1SPS

Type 2.1 - IS red 1 stage payment slip

Tp22.ISR2SPS

Type 2.2 - IS red 2 stage payment slip

Tp7.Dmstc

Type 7 - Domestic postal order

TpE1.PSWR

Type E1 - Payment slip with reference

TpE2.PSWN

Type E2 - Payment slip with notifications

**AvtaleGiro (NO)**

PAYMENT SPECIFICATION (USED IN ER)

EXPORT FORMAT DESCRIPTION

Varsling

AvtaleGiro-trans with notification

**AutoGiro (NO)**

PAYMENT SPECIFICATION (USED IN ER)

EXPORT FORMAT DESCRIPTION

Melding

Autogiro-trans with notification

**eFaktura (NO)**

AX 2012 FEATURE THAT WAS POSTPONED	PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)		
	<table border="1"> <tr> <td data-bbox="590 170 790 280">Reklame</td> <td data-bbox="790 170 1005 280">Include advertising flag</td> </tr> </table>	Reklame	Include advertising flag		
Reklame	Include advertising flag				
	<b>ISO20022 Credit transfer (DK)</b>				
	<table border="1"> <tr> <td data-bbox="590 347 790 459">PAYMENT SPECIFICATION (USED IN ER)</td> <td data-bbox="790 347 1005 459">EXPORT FORMAT DESCRIPTION</td> </tr> </table>	PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION		
PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION				
	<table border="1"> <tr> <td data-bbox="590 459 790 548">EasyAccountTransfer</td> <td data-bbox="790 459 1005 548">Easy-account with CVR (NKV)</td> </tr> </table>	EasyAccountTransfer	Easy-account with CVR (NKV)		
EasyAccountTransfer	Easy-account with CVR (NKV)				
	<table border="1"> <tr> <td data-bbox="590 548 790 660">Paym_slip</td> <td data-bbox="790 548 1005 660">Transfer forms (OCR)</td> </tr> </table>	Paym_slip	Transfer forms (OCR)		
Paym_slip	Transfer forms (OCR)				
	<b>ISPAG-CNAB240 format (BR)</b>				
	<table border="1"> <tr> <td data-bbox="590 728 790 840">PAYMENT SPECIFICATION (USED IN ER)</td> <td data-bbox="790 728 1005 840">EXPORT FORMAT DESCRIPTION</td> </tr> </table>	PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION		
PAYMENT SPECIFICATION (USED IN ER)	EXPORT FORMAT DESCRIPTION				
	<table border="1"> <tr> <td data-bbox="590 840 790 1075">A</td> <td data-bbox="790 840 1005 1075">OP (payment order), DOC (wire transfer), TED (other type of wire transfer), and direct credit in the account</td> </tr> </table>	A	OP (payment order), DOC (wire transfer), TED (other type of wire transfer), and direct credit in the account		
A	OP (payment order), DOC (wire transfer), TED (other type of wire transfer), and direct credit in the account				
	<table border="1"> <tr> <td data-bbox="590 1075 790 1310">J</td> <td data-bbox="790 1075 1005 1310">Bar code payments (invoice with bar code or other type of documents with bar code)</td> </tr> </table>	J	Bar code payments (invoice with bar code or other type of documents with bar code)		
J	Bar code payments (invoice with bar code or other type of documents with bar code)				
	<table border="1"> <tr> <td data-bbox="590 1310 790 1467">O</td> <td data-bbox="790 1310 1005 1467">Tax payments or other public services payments</td> </tr> </table>	O	Tax payments or other public services payments		
O	Tax payments or other public services payments				
US Payroll	US Payroll provides gross-to-net processing for employees in the United States. In Payroll, you can set up, enter, and maintain all payroll records and transactions.		Implemented in version 1611		



AX 2012 FEATURE THAT WAS POSTPONED	DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)
Vendor collaboration (Vendor Portal)	<p>Dynamics AX 2012 provided vendor portal capabilities via Enterprise Portal. Financial and Operations also provides these capabilities. In version 7.1 (also known as Dynamics 365 for Operations 1611), a vendor could view and respond to purchase orders.</p> <p>In version 7.3, the vendor can view and respond to RFQs. Vendors can also view and edit selected information from the vendor record such as addresses, contact information, and contact persons, and they can upload documents in relation to their certifications.</p>	Implemented in version 7.3
Vendor requests - external request to become a new vendor	<p>Dynamics AX 2012 provided the ability for an anonymous user to sign up to be a vendor in the system, which could lead to a vendor request for adding a new vendor to the vendor master. In version 7.3, the anonymous request from a prospective vendor can be imported via an entity (Data Management/OData), which can lead to inviting the vendor - or the vendor's contact person - to register more details about the prospective vendor. The information provided is included in a new vendor request that can be reviewed and approved via a workflow process. An approval of the vendor request leads to creation of a new vendor account.</p>	Implemented in version 7.3
Vendor requests in general	<p>Dynamics AX 2012 had a concept of vendor requests that served various purposes related to updating vendor-related information, such as requesting new procurement categories for the vendor, internal employees requesting new vendors, or requesting to add a vendor to another company. Only the vendor's request of being added as a vendor has been implemented in version 7.3.</p>	Not implemented

AX 2012 FEATURE THAT WAS POSTPONED	DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)
[Russia] Tax registers	<p>Legal entities can use registers to disclose their revenues and expenses. The registers are used to track revenue and expense data from the time that primary documents, such as sales invoices and delivery notes, are first entered by using the calculation of cost prices for production. The data from the registers is used to confirm the declared profit of the legal entity. This functionality includes the following features:</p> <ul style="list-style-type: none"> <li>• Current period incomes</li> <li>• Tax expenses</li> <li>• Other expenses of current period</li> <li>• Unrealized expenses of current period</li> <li>• Other unrealized expenses</li> <li>• Accounts receivable debt – inventory</li> <li>• Bad debts reserve calculation</li> <li>• Bad debts reserve movement</li> <li>• Accounts receivable movement</li> <li>• Procedure for writing-off AR bad debts</li> <li>• Accounts payable debt - inventory</li> <li>• Accounts payable debt movement</li> <li>• Procedure for writing-off AP bad debts</li> <li>• Goods cost calculation</li> <li>• FA object information</li> <li>• IA object information</li> <li>• FA depreciation</li> <li>• IA depreciation</li> <li>• FA/IA sale</li> <li>• Depreciation bonus recovery</li> </ul>	Implemented in version 8.1.3
[Russia] Electronic export/import format for Client-Bank interface and reconciliation procedure	Electronic formats for export of outgoing payments, and import of incoming payments.	Implemented in version 8.1.3
[Russia] VAT declaration	Electronic format of VAT declaration.	Implemented in version 10.0.1
[Russia] Cash Flow Management	The functionality which obtains a cash flow forecast and performs an analysis, manages payments on a daily basis using payment schedule journals, controls the company's cash position, and maintains the company's cash flows with centralized control,	Implemented in version 10.0.1

AX 2012 FEATURE THAT WAS POSTPONED	DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)
[Russia] Accounting reporting format	Electronic format of the following accounting reports: BalanceSheet, IncomeStatement, CashFlow, EquityStatement, TargetUsageMoney	Implemented in version 10.0.1
[Russia] Assessed tax reporting	Assessed tax declaration.	Implemented in version 10.0.1
[Russia] Land tax reporting	Land tax declaration. Creation of Land tax declaration by separate divisions.	Implemented in version 10.0.1
[Russia] Transport tax reporting	Transport tax declaration.	Implemented in version 10.0.1
[Russia] Indirect tax return (VAT and Excise) on import of goods	Indirect (withholding) tax return (VAT and Excise) on import of goods from state members of Customs union.	Implemented in version 10.0.1
[Russia] Journal of Alcohol sales in Retail	Daily Alcohol journal.	Implemented in version 10.0.1
[Russia] Optional posting of transfer orders to General ledger	Option to post/not post transactions to General ledger when posting a transfer order.	Implemented in version 8.1.2
[Russia] Inventory owner	Inventory dimension used to track owner of inventory (consignment stock, bailment, tolling, etc.).	Implemented in version 10.0.1
[Russia] AP/AR - Third-party miscellaneous charges	Registration of third-party miscellaneous charges and allocation by the following regimes: Inclusion into cost of purchased goods (allocation to invoices lines from other vendors), and redrawing to other parties re-allocation to other expense accounts.	Implemented in version 8.1.1
[Russia] Goods in transit from vendor	Registering goods in transit from vendor by special posting profile with Item type "purchased items en route". Creating Act of inventory holdings en route. (INV-6)	Implemented in version 8.1.2
[Russia] Goods in transit - sales to customer with postponed passing of property	Post sales invoice with postponed property transfer: no customer debts posted, all outgoing taxes are posted, items are transferred to transit warehouse. Register passing of property with posting debts and items sale from transit warehouse.	Implemented in version 8.1.2
[Russia] Bailment - accounting at bailee side	Accounting of inventory receipt for bailment as required by the Law and generation of primary form MX-1. Accounting of inventory return from bailment and generation of primary form MX-3. Bailment costs calculation from bailee side.	Implemented in version 8.1.2

AX 2012 FEATURE THAT WAS POSTPONED	DESCRIPTION	CURRENT STATUS (AS OF FEBRUARY 2019)
[Russia] Bailment - accounting at owner side	Accounting of inventory transfer to bailment and inventory return from bailment on goods owner side under bailment service contract.	Implemented in version 8.1.2
[Russia] Localization of Process Industries solution	Basic localization in two areas: correspondence of accounts for all new general ledger postings, and functional coexistence of Process Industries features and Russian country context (no issues when both Process Industries and Russian country context are enabled).	Implemented in version 10.0.1
[Russia] Alcohol sales declarations: Application 6, 7, 8 for wholesale. Applications 11, 12 for retail	Keeping track of alcoholic beverages types including producers, unit of measures, licenses for retail and wholesale trade. Preparing data for alcoholic beverages activities, including printing declarations and exporting them in XML format through e-reporting.	Implemented in version 10.0.1

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# End of mainstream support for Microsoft Dynamics AX 2009, Dynamics AX 2012, and Dynamics AX 2012 R2

2/18/2021 • 4 minutes to read • [Edit Online](#)

Mainstream support for Dynamics AX 2009 Service Pack 1 (SP1), Dynamics AX 2012, and Dynamics AX 2012 R2 ends on October 9, 2018. After that date, only security hotfixes will continue to be provided for these three versions through the extended support period that continues until October 12, 2021. For more information, see [support.microsoft.com](https://support.microsoft.com).

Mainstream support for Dynamics AX 2012 R3 continues through October 12, 2021. Microsoft will continue making security hotfixes, non-security hotfixes, and regulatory updates for Dynamics AX 2012 R3 throughout that mainstream support period. The source code for these non-binary, non-security hotfixes and regulatory updates will continue to be available for customers, and their partners, active on the Enhancement Plan or Software Assurance.

Dynamics AX 2009 SP1, Dynamics AX 2012, and Dynamics AX 2012 R2 customers can selectively integrate those changes as required. Customers and partners can get the source code from packages attached to relevant Dynamics AX 2012 R3 KB articles published on Lifecycle Services (LCS) and discoverable through Issue Search.

Customers are advised to upgrade to the latest version of Finance and Operations apps, such as Dynamics 365 Finance, Supply Chain Management, Retail, and Human Resources:

- Dynamics AX 2009 Service Pack 1 customers should use the [migration tool](#) that is available.
- Dynamics AX 2012 and Dynamics AX 2012 R2 customers should upgrade to Finance and Operations apps through Dynamics AX 2012 R3 using the upgrade tool that is available. Additional upgrade information is available in the [Upgrade from AX 2012 to Finance and Operations apps](#) topic.

## Frequently asked questions

### **When does the mainstream support for Dynamics AX 2009 Service Pack 1, Dynamics AX 2012, and Dynamics AX 2012 R2 end?**

Mainstream support ends on October 9, 2018.

### **Was the information of the end date of the mainstream support for Dynamics AX 2009 Service Pack 1, Dynamics AX 2012, and Dynamics AX 2012 R2 available before?**

Yes, it was always publicly available on the Microsoft Support Lifecycle site at [support.microsoft.com](https://support.microsoft.com).

### **Can customers on Premier Extended Hotfix Support or on Unified Support Advanced and Performance Levels get a non-security hotfix or regulatory update?**

No. Neither non-security hotfixes nor regulatory updates will be available for the Dynamics AX products during the Extended Support phase of the product lifecycle (Dynamics AX 2009 SP1, Dynamics AX 2012, or Dynamics AX 2012 R2).

While the ability to request a non-security hotfix for select products is included with Unified Support Advanced and Performance Levels, Microsoft has determined that non-security hotfixes cannot be provided with a *commercially reasonable* effort for these products. As a result, no requests for non-security hotfixes or regulatory updates will be accepted. However, Microsoft will continue making security hotfixes, non-security hotfixes, and regulatory updates for Dynamics AX 2012 R3 throughout that mainstream support period. The source code for these non-binary, non-security hotfixes and regulatory updates will continue to be available for

customers, and their partners, active on the Enhancement Plan or Software Assurance. Dynamics AX 2009 SP1, Dynamics AX 2012, and Dynamics AX 2012 R2 customers can selectively integrate those changes as required. Customers and partners can get the source code from packages attached to relevant Dynamics AX 2012 R3 KB articles published on LCS and discoverable through LCS Issue Search.

**I knew about the regulatory change before October 9, 2018, but it has the law enforcement date after October 9, 2018. Will I still get a regulatory update for Dynamics AX 2009 Service Pack 1, Dynamics AX 2012, and Dynamics AX 2012 R2?**

No, we will only provide regulatory updates for Dynamics AX 2009 Service Pack 1, Dynamics AX 2012, and Dynamics AX 2012 R2 for regulatory changes with the law enforcement dates on or earlier than October 9, 2018.

**A customer or partner can already download a fix through LCS and inspect the code by installing it into a test Dynamics AX 2012 R3 environment. Is there any difference with the approach that you have proposed?**

No, there is no difference.

**What happens if a new bug is found by a customer in Dynamics AX 2009 Service Pack 1, Dynamics AX 2012, or Dynamics AX 2012 R2?**

The bug must be reproducible in Dynamics AX 2012 R3. If it is reproducible and accepted, then a hotfix will be provided for Dynamics AX 2012 R3 and the customers can elect to integrate this hotfix in their version themselves, or work with their partners to integrate the changes.

**How are binary hotfixes handled for Dynamics AX 2009 Service Pack 1, Dynamics AX 2012, and Dynamics AX 2012 R2?**

If a hotfix is needed for a part of the system where Microsoft does not provide the source code and it is not a security bug, the hotfix will not be provided.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Help system

2/18/2021 • 6 minutes to read • [Edit Online](#)

Users of the following apps can access context-sensitive Help and other content that is based on the same Help system:

- Microsoft Dynamics 365 Commerce
- Dynamics 365 Finance
- Dynamics 365 Human Resources
- Dynamics 365 Supply Chain Management

In all these apps, you can access product-specific Help from the **Help** pane.

The screenshot displays the Microsoft Dynamics 365 interface. The main window shows a sales order for 'Contoso Retail San Diego' with a table of sales order lines. The Help pane is open on the right side, displaying a search bar and a list of help topics.

T...	Variant number	Item number	Product name	Sales category	CW quantity	CW unit	Quantity	Unit	Delivery type
		D0001	MidRangeSpeaker	Speakers			20.00	ea	Stock
		L0001	MidRangeSpeaker2	Speakers			29.00	ea	Stock
		P0001	AcousticFoamPanel				123.00	ea	Stock
		D0003	StandardSpeaker	Speakers			22.00	ea	Stock
		T0001	SpeakerCable / Speaker cable 10	Accessories			61.00	ea	Stock
		D0004	HighEndSpeaker / High End Spe...	Speakers			20.00	ea	Stock
		T0004	TelevisionM12037 / Television ...	Television			61.00	ea	Stock
		T0002	ProjectorTelevision	Television			37.00	ea	Stock
		T0005	TelevisionHDTVX59052 / Televisi...	Television			25.00	ea	Stock
		T0003	SurroundSoundReceive	Receivers			37.00	ea	Stock

The Help pane on the right contains a search bar and a list of help topics, including:

- Help
- Task Guides
- Showing help content for the current page
- Create sales orders (Publisher: Microsoft)
- Prospect to cash (Publisher: Microsoft)
- Generate and process customer rebates (Publisher: Microsoft)
- Credit card setup, authorization, and capture (Publisher: Microsoft)
- Confirm sales orders (Publisher: Microsoft)
- Working with serialized items (Publisher: Microsoft)
- Create a purchase order from a sales order (Publisher: Microsoft)
- Manage order holds (Publisher: Microsoft)
- RESOURCES
- Help
- Ask the community
- CustomerSource

## Help on docs.microsoft.com

The docs.microsoft.com site ([docs.microsoft.com/dynamics365](https://docs.microsoft.com/dynamics365)) is the default source for product documentation for the previously listed apps. This site offers the following features:

- **Access to the most up-to-date content** – The site gives Microsoft a faster and more flexible way to create, deliver, and update product documentation. Therefore, you have easy access to the latest technical information.
- **Content that is written by experts** – Content on the site is open to contributions by community members both inside and outside Microsoft.

You can find content on docs.microsoft.com by using any search engine. For the best results, we recommend that you use a site search, such as `site:docs.microsoft.com dynamics 365 "search term"`.

## Get notified about changes through an RSS feed

To subscribe to a Really Simple Syndication (RSS) feed of all updates that are to the content on docs.microsoft.com, use the following link:

[RSS feed](#)

## Leave us feedback



If you have feedback or questions about a topic, leave us a comment at the bottom of the page.

1. Select **Feedback** to get to the comments at the bottom of the page. Then, select either **Product feedback** or **Sign in to give documentation feedback**.
2. Start typing your comments, and then select **Submit feedback**.

## Feedback

We'd love to hear your thoughts. Choose the type you'd like to provide:

Our new feedback system is built on GitHub Issues. Read about this change in [our blog post](#).

Title

Leave a comment

### NOTE

If you want to submit documentation feedback, you must sign in by using a GitHub account. For more information, see [Setting up and managing your GitHub profile](#).

## Contribute to the documentation

You can contribute and make edits to the documentation. To get started, select the **Edit** button (pencil symbol) on a topic. The following video shows how you can contribute to our documentation.

The [How to contribute to the Microsoft Dynamics 365 documentation](#) video (shown above) is included in the Microsoft Dynamics 365 channel on YouTube.

For more information, see the [Docs contributor guide](#), which is published by the team that built the docs.microsoft.com site.

### NOTE

We only accept contributions to our English content at this time.

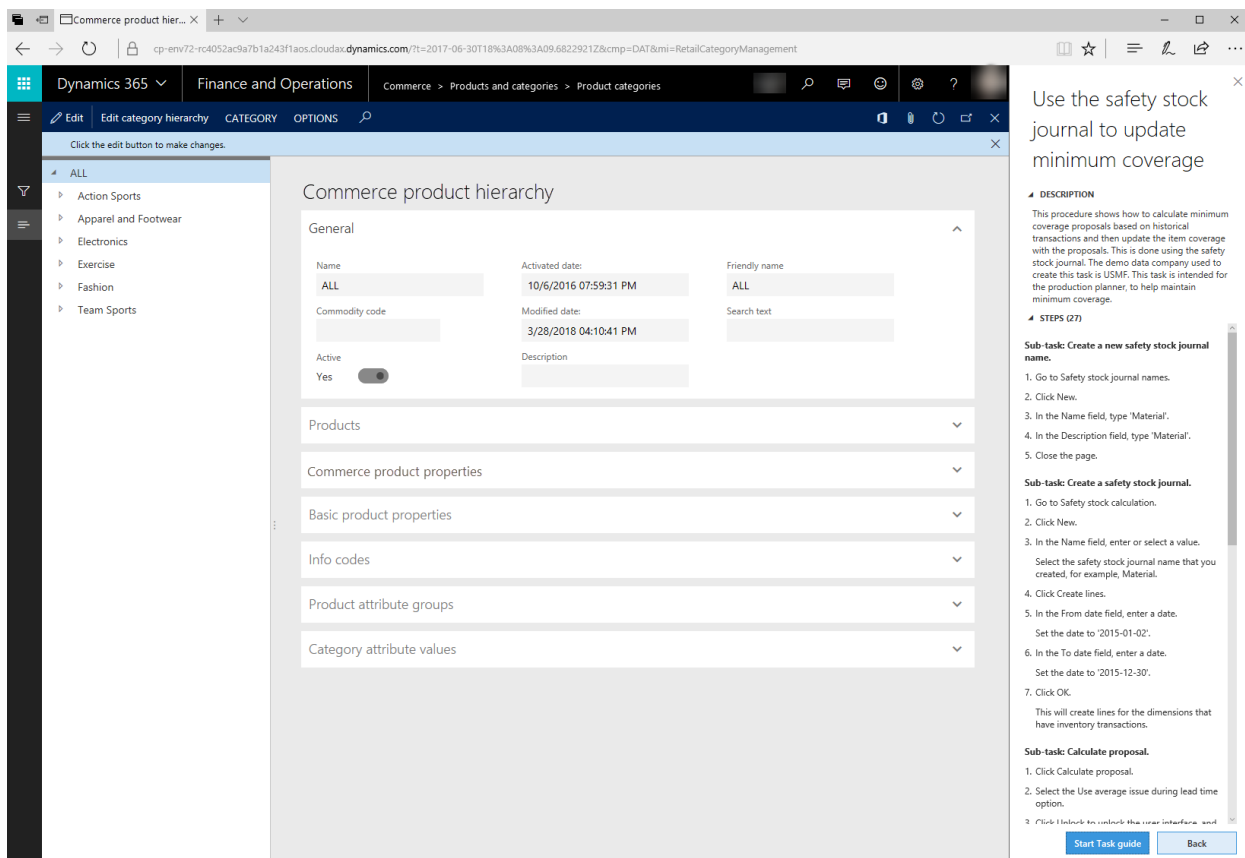
## Task guides

A task guide is a controlled, guided, interactive experience that leads you through the steps of a task, or business process. You can open (play) a task guide from the **Help** pane. When you first select a task guide, the **Help** pane

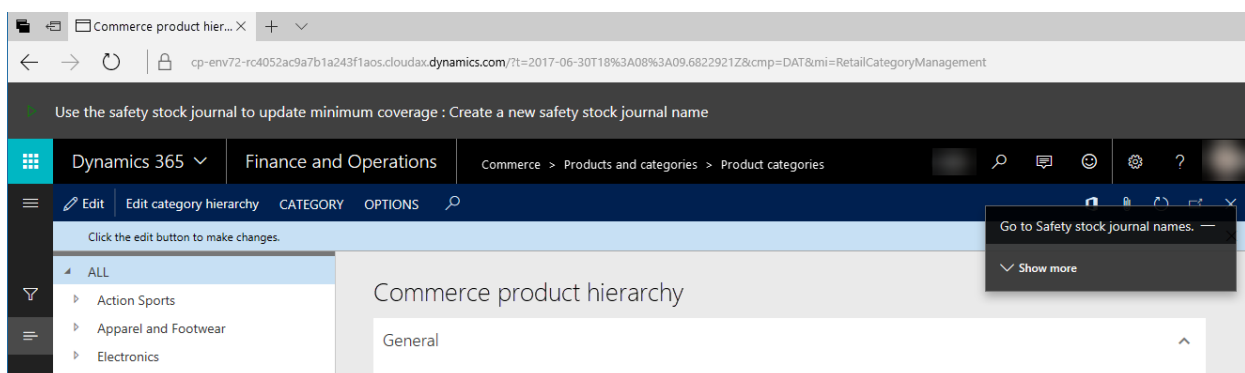


will show the step-by-step instructions for the task. Localized task guides are available.

Microsoft released task guide libraries for product versions through the December 2017 release of Dynamics 365 for Finance and Operations. The [Accessing task guides from the Help pane](#) section of this topic explains how to find the correct task guides for your product.



To begin the guided, interactive experience, select **Start task guide** at the bottom of the Help pane. A black pointer shows you where to go first. Follow the instructions that appear in the user interface (UI), and enter data as directed.



### IMPORTANT

The data that you enter when you play a task guide is real. If you're in a production environment, the data will be entered in the company that you're currently using.

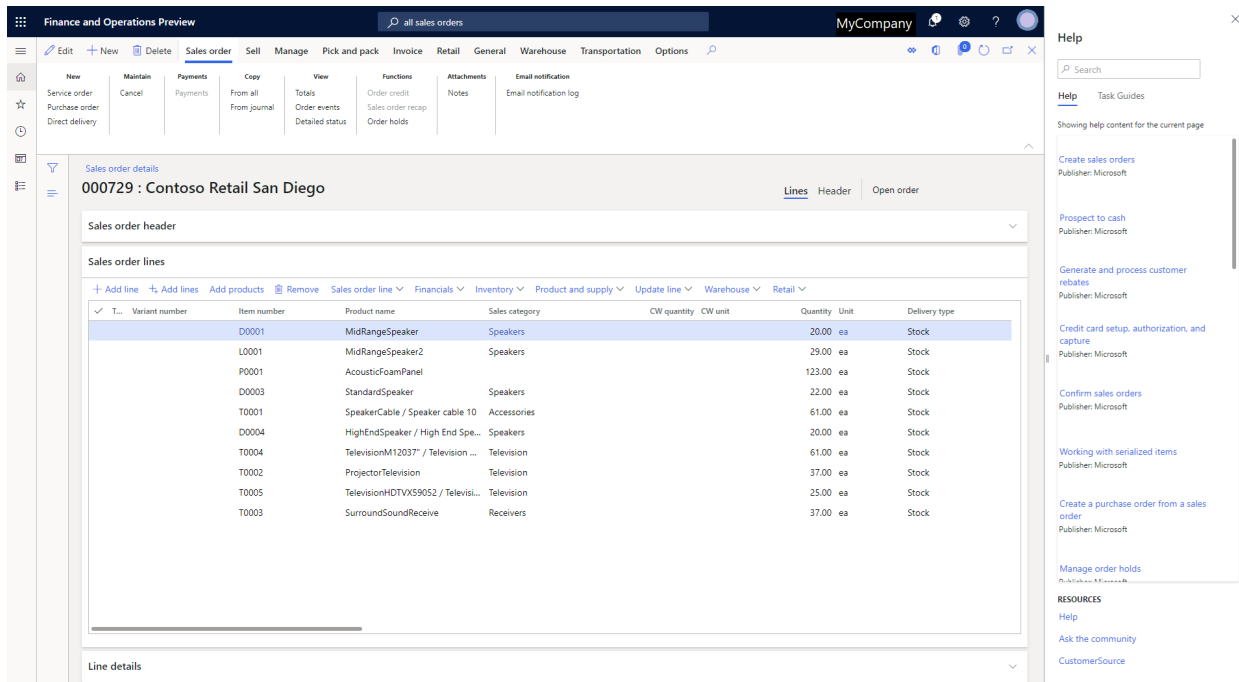
You can use Task recorder to create your own custom task guides. For more information, see [Create documentation or training with Task Recorder](#).

## In-product Help

Some fields have field descriptions to help users get unblocked when they are uncertain about the data that the

field contains, for example. Additionally, the in-product **Help** pane provides context-sensitive access to content that can help users get started, get unblocked, and learn more.

To access Help content, select the **Help** button (?), and then select **Help**. Alternatively, press **Ctrl+Shift+?**. In both cases, the **Help** pane appears. From the **Help** pane, you can access conceptual topics or task guides that are relevant to the area of the product that you're currently in.



### Accessing Help topics from the Help pane

From the **Help** pane, you can access topics that apply to the client. When you first open the **Help** pane, the **Help** tab shows the topics that apply to the page that you're currently on. If no topics are found, you can enter keywords to refine your search. When you select a topic in the **Help** pane, it's opened on a new tab in your browser.

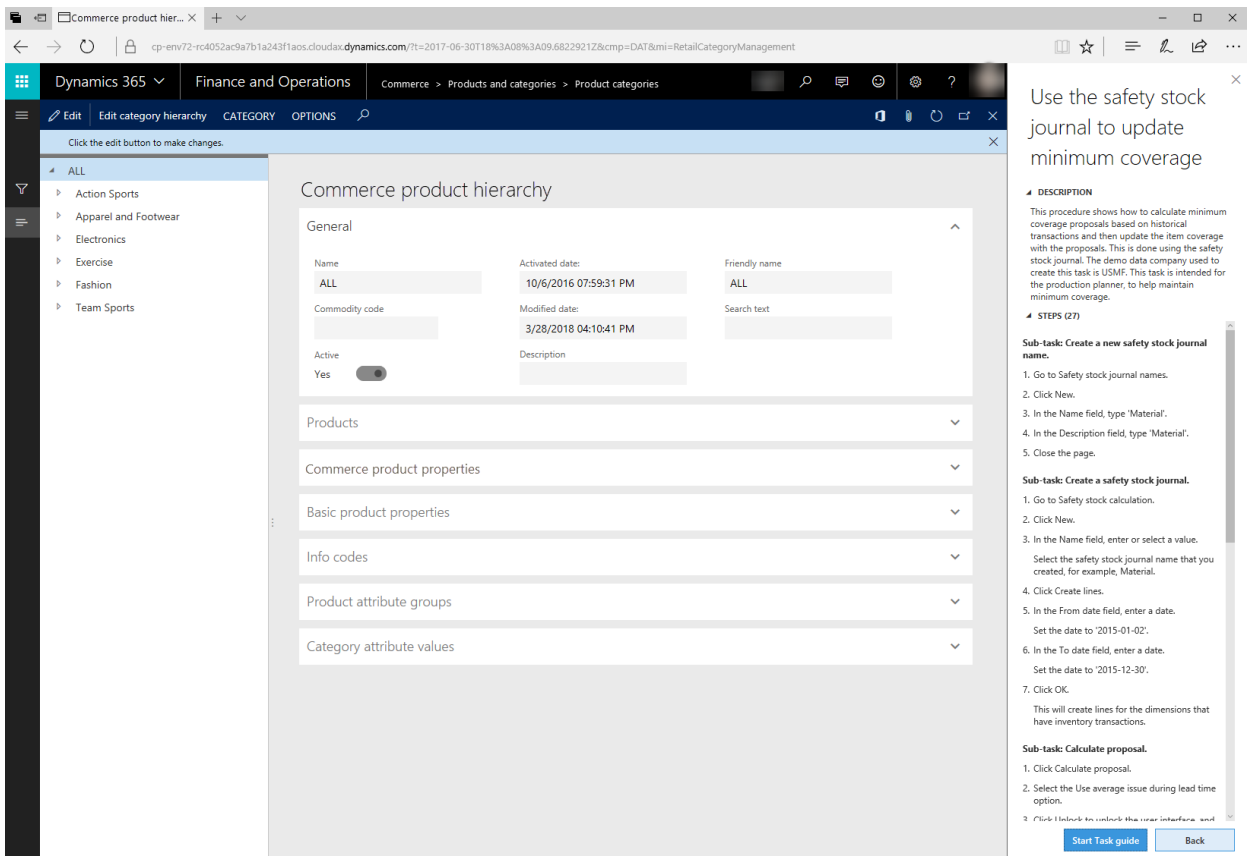
#### IMPORTANT

This section does not apply to Dynamics 365 Human Resources. The Help system for Human Resources is automatically connected to task guides for the product. Also, you cannot create custom task guides for Human Resources.

### Accessing task guides from the Help pane

Before you can access task guides from the **Help** pane, a system admin must configure some settings on the **System parameters** page in Finance, Supply Chain Management, or Commerce. For more information, see [Adding task guides](#).

After a System administrator has completed these steps, you can open the **Help** pane and select the **Task guides** tab. You'll now see the task guides that apply to the page that you're currently on. If no task guides are found, you can enter keywords to refine your search. After you select a task guide in the **Help** pane, the **Help** pane shows the step-by-step instructions, and you can play the task guide.



## Where are the translated task guides for Microsoft libraries?

Translated task guides are released in libraries that have "All languages" in the title. To view localized task guide Help, make sure that you're connected to an appropriate library. Each user can change the language that a task guide appears in by changing the language settings under **Options > Preferences**.

- If a task guide has been translated, when you open that task guide all the text of the task guide will appear in your selected language.
- If a task guide has not yet been translated, when you open it, only the text of the controls will appear in your chosen language.

## Creating custom Help

You can create Help for your users by creating custom task guides or connecting your own website to the **Help** pane. For more information, see the following topics:

- [Task recorder resources](#)
- [Custom Help overview](#)

## Additional resources

The following table lists our websites. Sites that have an asterisk (\*) next to the name require that you sign in by using an account that is associated with a service plan.

SITE	DESCRIPTION
<a href="https://docs.microsoft.com">Docs.microsoft.com</a>	This site hosts or links to all product documentation for Dynamics 365.
<a href="https://www.microsoft.com/learn">Microsoft Learn</a>	This site is the free Microsoft eLearning site.

SITE	DESCRIPTION
<a href="#">Microsoft Dynamics Lifecycle Services (LCS)*</a>	This site provides a cloud-based collaborative workspace that customers and partners can use to manage projects from pre-sales to implementation and operations. It's useful in all phases of an implementation.
<a href="#">Support blog</a>	This site provides tips and tricks that are posted by the Support team.
<a href="#">Docs.microsoft.com/previous versions</a>	This site hosts content from previous releases.
<a href="#">Dynamics Community</a>	This site hosts blogs, forums, and videos.
<a href="#">Microsoft.com/dynamics365</a>	This site provides evaluation and sales information.
<a href="#">CustomerSource*</a>	This site hosts training resources, downloadable reports, and white papers, and is the primary support site for service plan holders. Access to some resources on this site might require a service plan.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

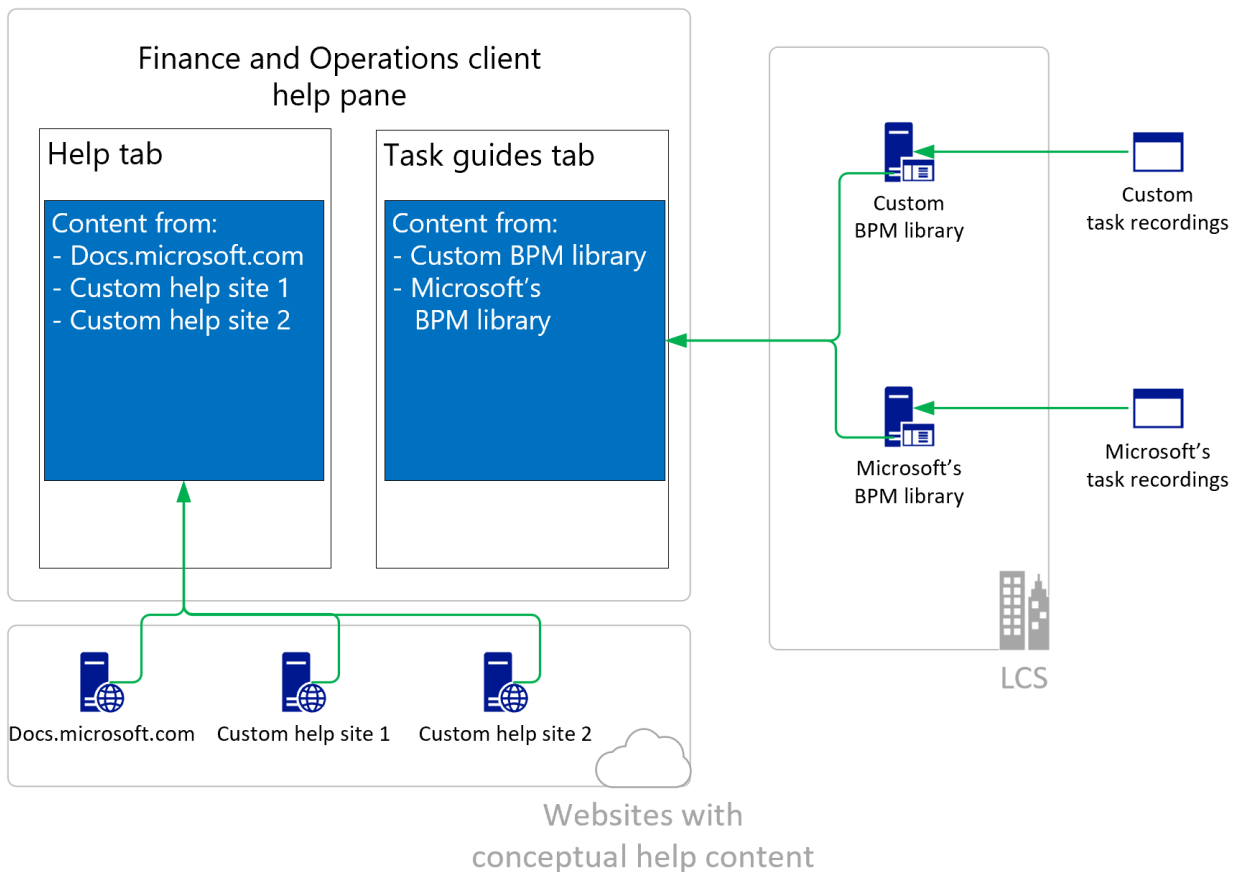
# Configure the Help experience for Finance and Operations apps

2/18/2021 • 4 minutes to read • [Edit Online](#)

In this topic, you will find an overview of the components of the Help system for Finance and Operations apps, such as Microsoft Dynamics 365 Finance, Dynamics 365 Supply Chain Management, Dynamics 365 Commerce, and Dynamics 365 Human Resources. The topic also explains how to connect these components and provides a summary of the process for creating custom Help.

## Help architecture

Finance and Operations apps include conceptual overviews and other topics that are published to the <https://docs.microsoft.com/dynamics365> site. This content can then be accessed from the in-product **Help** pane. The following illustration shows the parts of the Help system.



The in-product Help system pulls articles from docs.microsoft.com and other connected websites. It also pulls in task guides that are stored in Business process modeler (BPM) in Microsoft Dynamics Lifecycle Services (LCS).

## Adding task guides

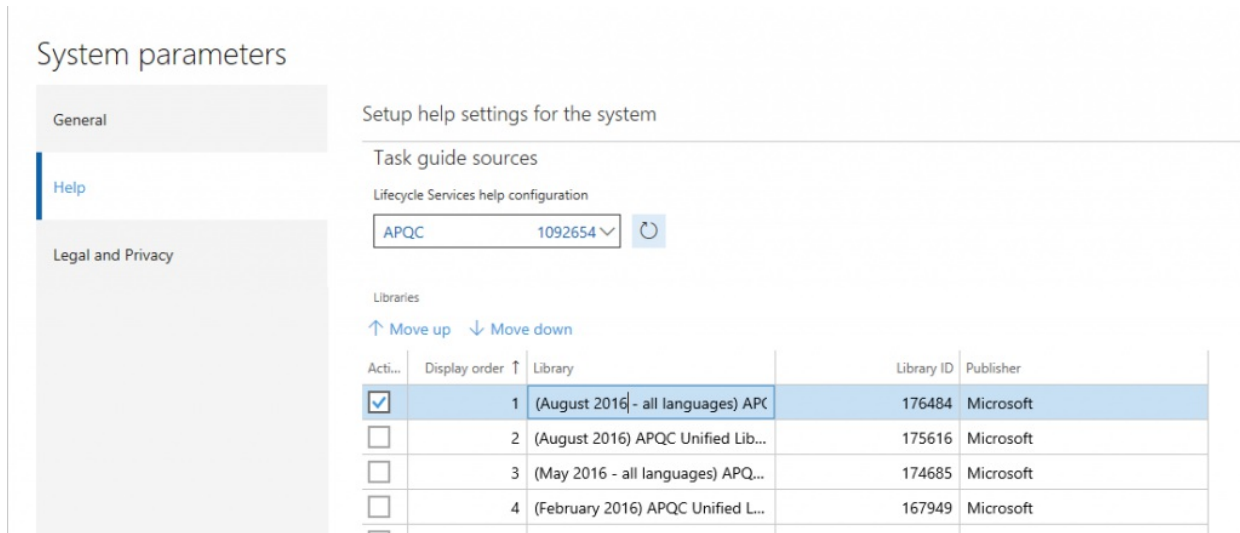
### NOTE

The **Task guides** tab isn't currently available in Human Resources or Commerce. However, the task guides in the Getting Started experience in Human Resources remain available to cover basic functionality. For both Human Resources and Commerce, procedural Help is available on the <https://docs.microsoft.com/dynamics365> site.

On the **System parameters** page, system admins can configure access to the relevant task guide libraries for an implementation.

**NOTE**

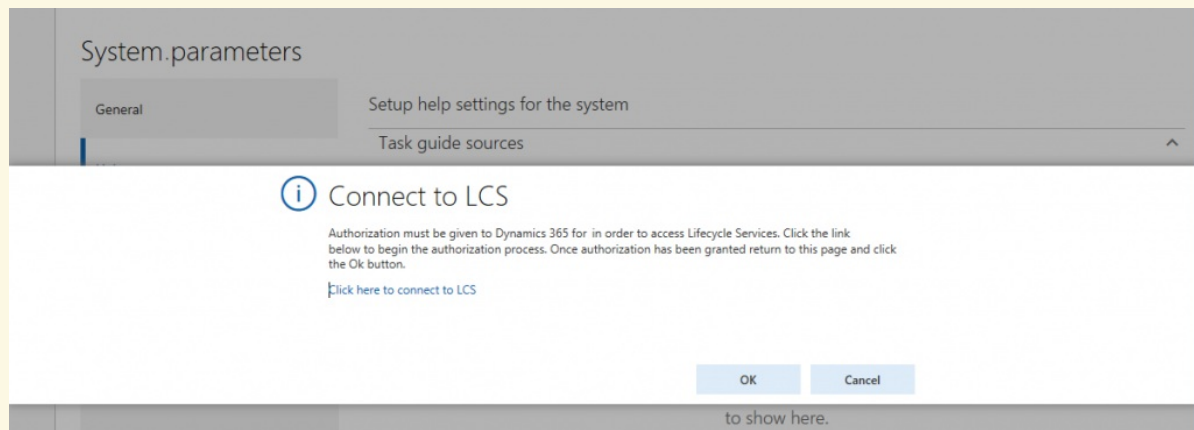
- To configure Help, you must sign in by using an account in the same tenant as the tenant where the app is deployed.
- An LCS library can't be connected from an instance of the app that is running on a local virtual hard drive (VHD).



To configure task guides for a solution, follow these steps on the **System parameters** page.

**IMPORTANT**

The first time that you open the **Help** tab, you must connect to Lifecycle Services. Be sure to select the link in the middle of the form, wait for the connection, close the dialog box, and then select **OK** to get to the **System Parameters** page.



1. Select the Lifecycle Services project to connect to.
2. Select the BPM libraries (within the selected project) to retrieve task recordings from.
3. Set the display order of the BPM libraries. The display order defines the order in which task recordings from the libraries will appear in the **Help** pane.

After you complete these steps, you can open the **Help** pane and select the **Task guides** tab. You'll now see the task guides that apply to the page that you're currently on in Finance and Operations apps. If no task guides are found, you can enter keywords to refine your search.

**Showing translated task guides**

Translated task guides were first released in the May 2016 APQC Unified Library and in the Getting Started library. To view localized task guide Help, make sure that your Dynamics 365 solution is connected to the May

2016 library. Users can change the language that a task guide appears in by changing the language settings under **Options > Preferences**.

#### **NOTE**

Although many task guides have been translated, the client doesn't currently show the translated task guide names. Additionally, in the May 2016 library, translations are available only for the task guides that were released in February 2016. Microsoft will release an updated library that includes additional translations.

- If a task guide has been translated, when you open that task guide all the text of the task guide will appear in your selected language.
- If a task guide has not yet been translated, when you open it, only some of the text (the text of the controls) will appear in your selected language.

## Adding custom Help

You can use task guides to create custom Help, or you can connect a custom Help website to the **Help** pane.

### **Create custom Help by using task guides**

You can create custom Help for the supported apps by creating task recordings that reflect your implementation and then saving them to a Business process library in LCS. You can't create custom task guides for Human Resources.

If you're a partner, and you promote a library to a corporate library and include it in a solution, it will be available to your customers. You can also make a copy of the APQC Unified Library, and then open the task recordings in the copy, edit them, and save your changes. For more information, see [Task recorder resources](#).

### **Connect a custom Help site**

Finance and Operations apps are rarely used in their out-of-box form. Instead, the solution is customized and extended to fit the organization's needs. You can also customize and extend the Help experience. For example, you can add custom Help to the in-product **Help** pane.

Microsoft has provided a toolkit to help you deploy and connect custom Help to the **Help** pane. For information about how you can set up a custom Help solution that is connected to the **Help** pane, see [Custom Help overview](#).

If you want to collaborate with Microsoft on tools and processes for customizing Help, fill in the form at <https://aka.ms/customhelpfeedback>.

## See also

[Help system](#)

[Custom Help overview](#)

[Task recorder resources](#)

[Create documentation or training with Task Recorder](#)

[Custom Help GitHub repository](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Custom Help overview

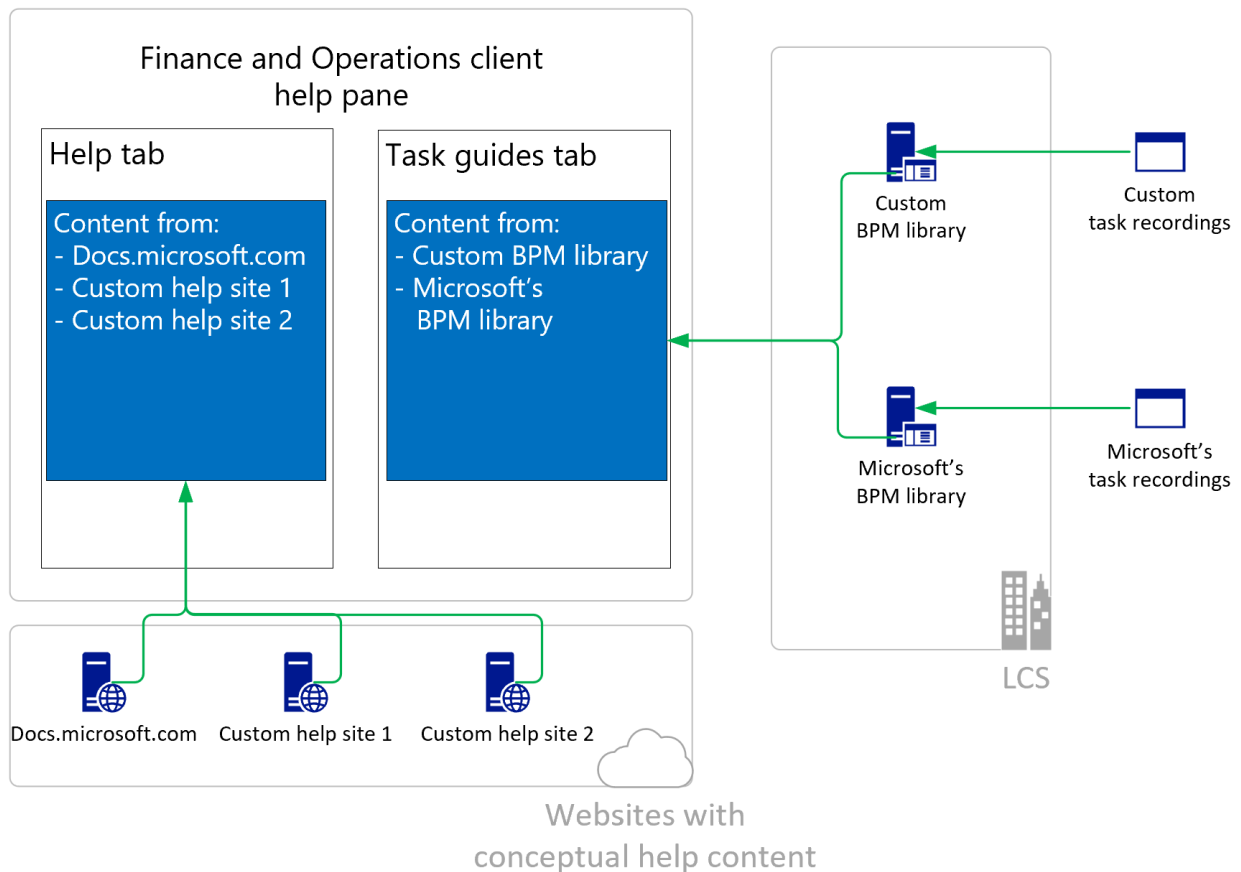
2/18/2021 • 4 minutes to read • [Edit Online](#)

Finance and Operations apps are often customized and extended to fit an organization's needs. If your solution is based on Microsoft Dynamics 365 Finance, Dynamics 365 Supply Chain Management, or Dynamics 365 Commerce, you can connect solution-specific and customer-specific Help content to the [Help pane](#) in the Finance and Operations client. This topic describes the main steps and decision points.

## NOTE

Users of Finance and Operations apps can create custom task guides to supplement conceptual content that describes the functionality of their solution. These conceptual descriptions are also referred to as Help and can be provided by Microsoft, partners, and an organization itself. For more information, see [Help system](#).

The following illustration, and this topic in general, use the term *Help* for conceptual descriptions that either include or exclude how-to guides. The term *task guides* refers to in-product task guides.



## Custom Help content

Custom Help content typically originates from one of three sources:

- Microsoft documentation repositories (repos)

You can use the [HTMLFromRepoGenerator](#) tool from the Custom Help Toolkit to clone content from any of the Finance and Operations repositories and generate corresponding HTML files. Those files can then be updated with content that is specific to your solution.



- Existing customized Dynamics AX content

You can [convert Dynamics AX custom Help content so that it can be used in Dynamics 365](#).

- HTML files that are created specifically for your solution

[Learn more about the metadata](#) that must be added to your HTML files for context-sensitive Help and search to work correctly.

## Process

The end-to-end process depends on the actual customer solution and the users' expectations. A typical process involves the following steps:

1. Create the custom Help content.
2. Publish the content on a website.
3. Index the content by using a search service.
4. Connect the custom **Help** pane to the website and the search service.

Microsoft provides a [toolkit](#) that can help you generate HTML files from the Microsoft Help repositories, generate JavaScript Object Notation (JSON) files for search services, and change the locale of HTML files so that it matches the locale of your solution.

You're welcome to share your knowledge by contributing to this documentation through the link at the bottom of the page or by joining the [Dynamics 365 community](#).

The following table outlines the main objectives that admins typically have for configuring the Help experience.

OBJECTIVE	LEARN MORE
I want to give my users a customized in-product Help experience that reflects their actual solution.	See the <a href="#">Custom Help websites</a> section of this topic and <a href="#">Create documentation or training with Task Recorder</a> .
I want to use the Microsoft Help content as a baseline for Help content that is specific to my solution.	See <a href="#">Custom Help Toolkit: The HtmlFromRepoGenerator tool</a> .
I want to contribute to the Microsoft Help content.	See <a href="#">Extend, customize, and collaborate on the Help</a> .
I want to reuse my existing Dynamics AX content.	See <a href="#">Convert Dynamics AX custom Help for use in Dynamics 365</a> .
I want to set up a website for my Help content.	See the <a href="#">Custom Help websites</a> section of this topic.
I want to add my content to the <b>Help</b> pane.	See <a href="#">Connect a custom Help website to the Help pane</a> .
Our technical writers want guidance that will help them convert our earlier content into Markdown so that it becomes easier for them to customize the Microsoft content.	See <a href="#">Moving to Markdown</a> .

## Custom Help websites

Before the product can connect to your Help content, you must customize the in-product **Help** pane so that it shows your content. The following conditions must be met:

- Your content must be available on a website.

You can deploy your content to an existing website, or you can set up a dedicated website to host your

content. The website can be private or public, but we recommend that users **not** be required to sign in to access your content.

- Your content must be indexed by a search service.

If you use the [AzureSearchCustomHelp](#) solution that is part of the [Custom Help Toolkit](#) for context-sensitive Help, the **Help** pane will generate a query that must be run against the search service's index. The query depends on specific metadata in the Help topics. For more information, see [Metadata requirements for custom Help topics](#).

The [Deploy custom Help to Azure](#) topic describes an approach for hosting content on Azure. It includes information about how to set up a search service that indexes your content so that it can be found by the in-product **Help** pane. If you don't have an [Azure subscription](#), create an account before you begin. You can start with a free account for 12 months. For more information, see [Create your Azure free account today](#).

## See also

[Connect a custom Help website to the Help pane](#)

[Deploy custom Help to Azure](#)

[Custom Help Toolkit](#)

[Language and locale descriptors in the product and in Help](#)

[Configure the Help experience for Finance and Operations apps](#)

[Help system](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# View and export field descriptions

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article describes how to view field descriptions and how to use the Field descriptions page to export descriptions.

Some of the more complex fields have field descriptions. These descriptions appear when you hover over a field. You can also view and export descriptions on the **Field descriptions** page.

Not all pages have field descriptions. We want to provide descriptions only for the more complex fields, not where the use of the field is obvious. Therefore, some pages don't have any field descriptions, some pages have a few descriptions, and some of the more complex pages, such as many of the parameters pages, have many descriptions.

If you have access to the development environment, you can add new field descriptions and customize existing descriptions. For example, you can add company-specific information to a field description. For more information, see [Customize field descriptions](#).

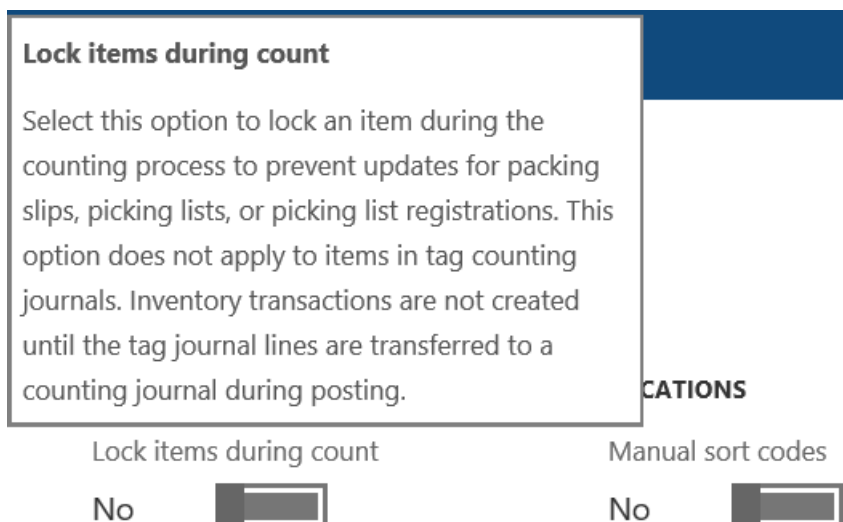
## See field descriptions in the user interface

You can view field descriptions by hovering over a field. If no description is available, you see the field name when you hover over the field.

### NOTE

In Dynamics AX 7.0 (February 2016), field descriptions can be viewed only on the **Field descriptions** page.

The following illustration shows the field description that appears when you hover over the **Lock items during count** field.



## Use the Field descriptions page to view and export field help

The **Field descriptions** page lets you view and export field descriptions. You can see the descriptions that are available for one page at a time.

**View the descriptions for a page**

To view the descriptions for a page, follow this step.

- In the **Select a page** field, type the name of the page. Alternatively, click the arrow to open a list of all the pages, and then browse or filter the list.

You can use either the name of the page that is shown in the user interface (UI) (for example, **Customers**) or the code name (AOT name) that's available when you right-click a page (for example, **CustTable**).

For information about the various ways to filter the list of pages, see the "Searching for a page" section later in this article.

If you set the **Include fields without a description** option to **Yes**, all the fields on the page are shown, even if they don't have a field description.

### **Export the descriptions for a page**

To export the descriptions for a page, follow these steps.

1. In the **Select a page** field, select a page.
2. Click the **Open in Microsoft Office** button in the upper-right corner, and then click **FieldDescriptionTmp**.

### **Searching for a page**

There are several ways to search for a page in the **Select a page** field. In many cases, you must click the arrow in the **Select a page** field to open the drop-down list, and then select from a filtered list of pages.

- Type part of the name, and then open the drop-down list to select from a filtered list of pages.
- Open the drop-down list, and then click either the **Page name** heading at the top of the list or the **Page AOT name** heading. A dialog box appears, where you can use advanced filtering options, such as **Page name begins with**.
- Type the full name of the page. When you use this option, it's best to open the drop-down list and see what else is in the list, even if field descriptions are shown.
  - If there is a single exact match to the name, the field descriptions for that page are shown.
  - If there is more than one exact match, no descriptions are shown. You must open the drop-down list and select the page that you want.
  - If the name that you typed is part of the name of another page, you see the descriptions for your page. However, if you open the drop-down list, you see additional pages that contain that name.

For example, no descriptions are shown when you type **Counting** in the **Select a page** field. You open the drop-down list, and see that there are two pages that have the name **Counting** and several pages that contain the word "Counting" in the name. If you select the page that has the AOT name **InventJournalCount**, the field descriptions are shown for that page. However, if you open the drop-down list again, you will see that the list now contains all pages that have "InventJournalCount" as part of their AOT name.

## **Troubleshooting**

This section provides information to help you troubleshoot issues that you might encounter when you use field descriptions.

### **I can't find a field description**

We're in the process of adding descriptions for the more complex fields. If you require help for a particular field, let us know by adding a comment for this topic.

### **The field description isn't helpful**

Let us know by adding a comment for this topic. If you can, describe the additional information that you require.

### **I can't find a field on the Field descriptions page**

To show all the fields on a page, set the **Include fields without a description** option to **Yes**. Click the **Select a page** field to verify that you've selected the correct page. If the name that you typed is part of another field name, you might have selected the page that has the longer name.

### **I can't find a page on the Field descriptions page**

For information about the various way to find pages, see the "Searching for pages" section earlier in this article. If you've typed the exact name of the page, the field descriptions might not be shown if more than one page has the same name. Click the arrow in the **Select a page** field to open a filtered list of the pages that are available.

## Additional resources

[Customize field descriptions](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Task recorder resources

2/18/2021 • 33 minutes to read • [Edit Online](#)

This topic describes how to use Task recorder to record business processes.

## Overview

### Task recorder

Task recorder for Finance and Operations apps is a utility that lets users record business processes for several different use cases. Here are some examples:

- Step-by-step guided tours of a specific business process in the application itself
- Documentation of a business process as a Microsoft Word document that can optionally include screenshots
- Regression tests for a business process
- Automatic playback of a business process in the application

Task recorder for Finance and Operations apps boasts high responsiveness, a flexible extensibility application programming interface (API), and seamless integration with consumers of business process recordings. Task recorder is also integrated with the [Business process modeler \(BPM\)](#) tool in Microsoft Dynamics Lifecycle Services (LCS), so that users can continue to organize their recordings. However, users can no longer produce business process diagrams from recordings.

Task recorder can automatically generate application regression tests from business process recordings and play back previously recorded processes. These features also include test-specific gestures that let users take full advantage of Task recorder.

### Architecture

Task recorder can record user actions in the client with exact fidelity, because every control is instrumented to notify Task recorder about the execution of user actions. The control notifies Task recorder that an event has occurred and passes all the relevant information about the corresponding user action in real time. From this information, Task recorder can capture the type of user action (for example, a button click, value entry, or navigation) and any data that is related to the user action (for example, the input data value and type, form context, or record context). Task recorder persists the information with enough detail to ensure that a playback of the recording can run the recorded actions exactly as they were performed by the user.

### Basic configuration

Task recorder is included with every Finance and Operations app, and lets users begin to record business processes immediately after they open the client for the first time.

#### IMPORTANT

The **Task guides** tab is currently not available in Commerce or Human Resources. We are currently working to enable this functionality in a future release. Task guides in the Getting Started experience in Human Resources remain available to cover basic functionality. Procedural help is also available on the docs.microsoft.com site (<https://docs.microsoft.com/dynamics365/>) for both Commerce and Human Resources.

## Start a new recording

The following steps show how to use Task recorder to start a new recording.

1. Open the product, and sign in. It's a good practice to refresh the browser before each new recording. A refresh creates a new user session and restarts Task recorder. Therefore, it provides the most stable recording experience.
2. Select the company that you want to use while recording. If this is your first time using Task recorder, you can follow along as this tutorial creates a sample recording based on a Fleet Management business process. You will need to load the Fleet demo data to follow along:
  - a. Go to **Dashboard > Fleet Management > Fleet setup**.
  - b. Click **Load demo data**.
  - c. When the data is finished loading, click **Close**.
  - d. Go back to the **Dashboard** by clicking the product name in the navigation bar.
3. Go to **Settings > Task recorder**.
4. The **Task recorder** pane is opened. You can click the **Close** button (X) in the upper-right corner to close the **Task recorder** pane before you begin a new recording. You can reopen the pane by following the previous steps.
5. Click **Create recording**.
6. Enter a name for the recording and click **Start**. Recording begins the moment **Start** is clicked. For the Fleet example in this tutorial, we'll use the name "Create a new rental reservation."

While you're recording, you can click the **Close** button (X) in the upper-right corner to hide the **Task recorder** pane without stopping the recording. You can reopen the pane by clicking the **Task recorder** button that appears at the top of the page. This button appears only while recording is in progress.

#### NOTE

If the **Saved views** feature is turned on, recordings should be created by using either published views or the standard view, to ensure that recordings work reliably for users.

7. Task recorder enters **recording mode**. The pane shows information and controls that are related to the process of recording.

Now you're ready to record a business process using Task recorder. If you're following this guide as a first-time user, you may complete the following Fleet Management scenario as an example. Otherwise, you can record your own application scenario.

#### Record a Fleet Management scenario

1. In the **Task recorder** pane, click **Start sub-task**.
2. Set **Name** to "Create a new rental customer". Leave the **Comment** field blank.
3. Click **OK**. The task is added to the step list.
4. Go to **Dashboard > Fleet Management > Reservation Management**.
5. Go to **All customers** under the **Summary** tab.
6. In the Action Pane, click **New**.
7. Enter a first and last name for the customer.
8. Click **Save**.
9. In the **Task recorder** pane, click **End sub-task**.
10. Return to the **Reservation Management** workspace by clicking the browser back button twice.
11. In the **Task recorder** pane, click **Start sub-task**. Name the task "Rent a vehicle to the new customer". Click **OK**.
12. Click (+) **Rental** under **Summary**.

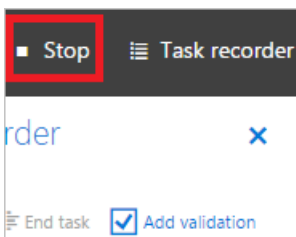
13. Under **Information**, select a "1975 Litware McKinley" as the vehicle.
14. Under **Information**, set the customer to the one just created.
15. Expand the **Discounts** section.
16. Click **Add** under **Discounts** and add the Frequent Customer discount. Click **OK**.
17. In the Action Pane, click **Start Rental**.
18. Set the return date to some date in the future.
19. Click **OK**.
20. In the **Task recorder** pane, click **End sub-task**.
21. Click **Stop** at the very top of the page.

## Recording a business process

After you've started your recording, you can perform your business process just as you would typically perform it by using the web client. As you interact with the product, new steps are added to the step list in the **Task recorder** pane. In this section, you will learn about other actions that you can perform while you're recording a business process, to take full advantage of Task recorder's capabilities.

### Stop

**Stop** is used to end the recording session. Before you click **Stop**, you should make sure that the recording is completed, because this action isn't reversible. When you click **Stop**, you're taken to the download options screen.



### Start/End sub-task

**Start/End sub-task** lets a user specify the beginning and end of a set of grouped steps in a recording. Click the **Start sub-task** button to add a "Sub-task" step to the end of the current list of recorded steps. The sub-task will include all steps that you perform from this point until you click the **End sub-task** button. When you click the **End sub-task** button, an "End sub-task" step is also added to the list of recorded steps.

#### NOTE

You must start a sub-task before you perform/record the steps that you want to include in the task. Then, after you've performed/recorded all the steps that you want to include in the task, you must end the sub-task.

Sub-tasks are purely an organization tool, and consumers of business process recordings can interpret the task groupings in useful ways. Because tasks can be nested inside other tasks, they provide the flexibility to organize very long and complex business processes.

### Delete/Restore step

**Delete/Restore step** enables a user to remove steps from the recording, or undo the removal of a step from the recording. You must first select the step in the Steps list that you want to delete/restore, and then click the **Delete/Restore step** button.



## NOTE

The behavior of the **Delete** button changes when you play back a recording. In playback mode, a deleted step can't be restored after playback has passed the point where it would have run the deleted step. For example, you load a recording that contains three steps, and then you delete step 2 before you start playback. You can restore step 2 only as long as playback hasn't run step 3. After you start playback, and playback has "skipped" step 2 (because you deleted it) and run step 3, you won't be able to restore step 2. Because step 2 wasn't run and therefore wasn't recorded, it can't be retroactively added back into the recording at its previous position.

## Add developer placeholder

**Add developer placeholder** lets the user add a placeholder step to the list of recorded steps. This placeholder step doesn't appear when the task guide is viewed, and it isn't run during maintenance of a recording. It's used only by the [Regression suite automation tool \(RSAT\)](#) or the X++ code generator that enables an X++ test to be created from a task recording. When the code generator creates an X++ test, it automatically adds a method stub to the generated code. The developer can then add X++ code into this method stub. The automated code will call the validation when the generated test is run at the point in the recording where this placeholder was added.

## Enriching steps in a recording

There are various options for enriching a step in a recording. For example, you can adjust the text that is associated with a step and add information about a specific step. This section describes the step enrichment capabilities that are available. To access these options, click the **Edit step** button on a specific step of a recording.

### Step instruction

The **Step instruction** is the primary text that is displayed for this step in the task guide. There are usually 2-3 alternative options for step instructions, and they appear in the following order when editing the annotation.

#### Step instruction

- In the First name field, type 'John'.
- In the First name field, type a value.
- In the First name field, { your example text }.

This image shows the annotation options for changing a step.

- **Preferred value instruction** This type of instruction will direct the user to enter the same data that was used when the step was recorded. *Example:* In the First name field, enter 'John'.
- **Example value label** This type of instruction will direct the user to enter their own data, indicating that the data that was used when the step was recorded was only *Example* data. *Example:* In the First name field, enter a value.

If users click the **See more** button on this step when they play the recording as a task guide, they will be able to see the data that was used when the step was recorded. This recorded data value will be labeled as an *Example* data value.

## NOTE

Steps that are not related to fields, such as clicking buttons, opening forms, or selecting records from a lookup, do not set *Example value label* as an option when annotating.

- **User-supplied value label** This step instruction contains placeholder text, which the author can fill in with their own text. For steps which have an **Example value label** option, the placeholder allows substituting the text which normally specifies the data to enter. This is useful for scenarios where neither the **Preferred value label** nor the **Example value label** sufficiently express the data that should be used for this step.
    - *Example label:* In the First name field, enter *{your example text}*.
    - *Example label after supplying the placeholder text:* In the First name field, enter the customer's name.
- For steps which do not have an **Example value label** option, the placeholder allows substituting all of the label text. Steps associated with buttons, for example, do not have **Example value labels**, so you may replace the entire label text with your own text.
- *Example label before replacement:* Click Post.
  - *Example label after replacement:* To post the order, click Post.

### Titles and notes

Titles and notes provide places for user-specified text to be associated with a step in a task guide.

- **Title** – The title lets you specify the text that appears above the step instruction for this step in the task guide. The title a good place to put text that you want users to read before they complete the action that is indicated by the step instruction.
- **Note** – You can use a note to specify text that appears in the expandable section of the pop-up for this step in the task guide. A note is a good place to put optional reading material or other information that might be useful to users, but that they aren't required to read to complete the action that is indicated by the step instruction.

### Change recorded values

Starting in version 10.0.12, you can adjust the values that are recorded in basic input controls (for example, simple text, numeric, date, and picklist fields), without having to re-record those steps. Note that lookup controls and reference groups aren't currently supported.

### Hide from task guide

The **Hide this step** option lets the author prevent specific steps from appearing in the task guide. This option is useful for hiding steps that are required for the task recording to run in playback mode, but that should not be seen by users. Examples of these steps include copy steps, system-generated steps, and data clean-up steps. If you hide a sub-task, all the steps that are recorded inside that sub-task will also be hidden.

## Using control gestures

The basic recording capability lets a user record an end-to-end business process by using Task recorder, but without adding overhead to the process. In some circumstances, more advanced recording features can be used to create even richer business process recordings. Each of the following gestures is found under the **Task recorder** option on the shortcut menu (also known as a right-click menu or context menu) for a control and causes a step to be added to the recording. If the gesture isn't supported for a control, it won't appear on the shortcut menu for that control.

### Copy

The **Copy** gesture lets you copy the value for the current control to the Task recorder "clipboard." That value can then be used later as part of a **Paste** or **Validate** gesture. Because values from multiple controls might have to be pasted, the Task recorder clipboard maintains a list of all control values that have been copied in the recording.

### Paste

The **Paste** gesture lets you paste a value from a previous **Copy** gesture in the same recording. The Task recorder

paste function works like the standard paste function that users might be familiar with, but it has an additional benefit when it's used during recordings. Because Task recorder will replay the recorded **Copy** and **Paste** commands during playback, if the copied control has a different value than it had during recording, Task recorder will paste the current value instead of the value that the copied control had during recording. This feature is useful in scenarios where the copied control has a value that can change between environments (for example, reclD values or number sequences).

There is an additional benefit from using the **Copy** and **Paste** gestures when test code is generated. For any control where the value is set via the **Paste** command, Task recorder doesn't have to create a parameterized input variable for that control's value, because it's set based on another control's value. This feature can be very useful in scenarios where an entity such as a customer is created, and an identifier for that entity is frequently entered during the recording. Instead of manually re-entering the customer name or ID throughout the scenario, and causing Task recorder to generate a parameterized input variable for each entry, the user can copy the customer name or ID one time, and then repeatedly paste it. In this case, Task recorder will generate a single parameterized input variable to represent the customer name or ID. This feature can make it much easier to change the input data for a generated test.

## Validate

The **Validate** gesture lets you insert a step that validates the value of the targeted control. This gesture always uses equality to validate the control value. *Validations aren't currently run during recording playback.* Instead, they are run only when the generated test code is run. Two kinds of validation are available:

- **Current value validation** will capture the targeted control's value at the time of recording and use it to generate an assertion in the test code. In the list of validation options on the shortcut menu, **Current value** is always first.
- **Reference value validation** will use the value of a previously copied control when generating an assertion in the test code. This allows creating assertions that are resilient to changes in the data, since the value is not hardcoded into the test code. In the list of validation options on the shortcut menu, **Reference value validation** follows the format [AOT name of copied control: current copied value].

Additional options are available in version 10.0.13 and later. Here are some examples:

- **Enabled/Disabled** validates that the targeted control's state is enabled (or disabled), and then uses that validation step to generate an assertion in the test code.
- **Read-only/Editable** validates that the targeted control's state is read-only (or editable), and then uses that validation step to generate an assertion in the test code.

## Add info step

The **Add info step** gesture lets you insert a step and supply your own text for it. This feature is useful primarily for creating task guides. An **informational step** (or **info step** for short) is a task guide step where the instruction text for the step is user-specified. Info steps are useful for describing actions that are a part of the scenario but must occur outside the client. For example, a scenario might require the user to search for item inventory or check an email for information.

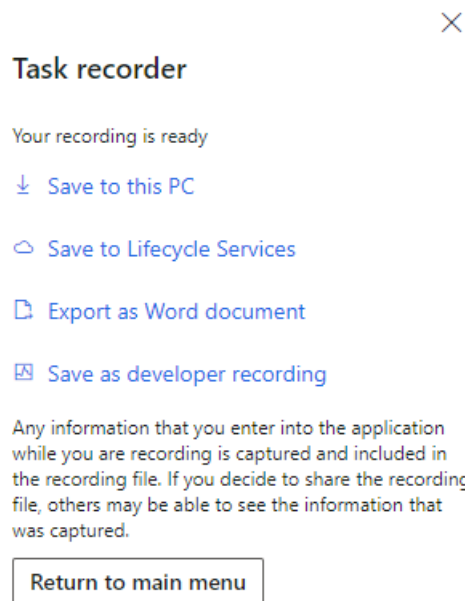
You can specify where an info step should appear in the task guide. The info step can point to a control on the page, if the step is associated with the control. Alternatively, the info step can appear in the upper right of the page, if the step is external to the client, or if it's an explanation that applies to the whole page.

### NOTE

Because info steps are manually specified steps and are not automatically recorded by Task recorder when the user takes an action on a control, the info step does not have the capability to automatically progress when a user completes the step in the task guide. Because the info step is not associated with taking an action in the client, there is no action for a task guide to detect that the user has completed in order to automatically progress to the next step.

# Options after a recording is completed

After you click **Stop** to end your recording session, several options are shown so that you can save the files that are related to the completed recording. Select **Save to this PC**, and save the task recording package to your desktop. You will use this file later.



## Save to this PC

One option after you finish your recording is to download the task recording package (an .axtr file) to your computer. This file can be loaded later via the **Task recorder** pane, so that the recording can be played as a task guide or edited.

## Save to Lifecycle Services (LCS)

When you save your recording to an LCS library, it's published on the specified business process in a BPM library. If the selected LCS library is set as a Help library, you will be able to find the task guide for the recording by searching the **Help** menu.

### NOTE

To be able to save a recording to an LCS library, the user must be in the Azure Active Directory (Azure AD) tenant that the environment was deployed from.

## Export as Word document

The Microsoft Word document for your recording contains the recorded steps as well as any screenshots that were captured.

## Save as developer recording

The raw recording file (developer recording) is useful for developer scenarios, such as test code generation and scenarios where **RSAT** is used.

# Playing back a recording

The **playback** functionality of Task recorder can automatically run the steps of an existing recording by using the pages and values that were originally recorded. Playback mode can be used to update an existing recording if changes were made to the underlying application, and those changes altered the business process steps that are required for the scenario. It's important to remember that, in this mode, Task recorder simultaneously re-

records the steps and plays them back. When the playback is completed, a new recording is produced that reflects both the steps that were run from the existing recording and any new steps that the user manually performed. Any steps that aren't run either by the user or automatically by Task recorder aren't included in this new recording.

To play back an existing recording, follow these steps.

1. Refresh the browser tab.

#### NOTE

It's a good practice to refresh the browser before each new recording.

2. Open the **Task recorder** pane.
3. Click **Playback recording**.
4. Click **Open from this PC** to load a recording from a previously downloaded Task recorder package (.axtr file).
  - If you're reading this guide for the first time and following along, choose the "Create a new rental reservation" file that you downloaded previously.
5. Click **Start**.

When you play back a recording, additional actions are available in the **Task recorder** pane.

#### **Play next pending step**

**Play next pending step** runs the next step in the recording. This action is useful because it gives you more control over the playback speed when you want to analyze the effects of a single step. This action has a side-effect that it's important to be aware of. When you click **Play next pending step**, any open lookups, drop-down dialog boxes, or Action Pane tabs might be dismissed, because this action removes focus from those elements. For these situations, we recommend that you use **Play all pending steps** instead.

#### **Play all pending steps**

**Play all pending steps** begins sequential execution of the remaining steps in the recording, and continues until either playback is paused or all steps have been run. During playback, the **Play all pending steps** button is replaced by a **Pause** button that can be used to pause playback. If playback can't successfully run a step for any reason (for example, because it can't find a button that has been renamed), Task recorder will skip that step, and playback will automatically be paused. In this way, the user has an opportunity to replace the obsolete step by completing the new steps in the client. Task recorder will record the new steps and ignore the step that was skipped. The user can then click **Play all pending steps** to continue playback for the remaining steps. After the recording is completed, the user can download the updated recording. This recording will contain all the steps of the original recording, but will exclude any skipped steps and include any new steps.

#### **Play to selected step**

**Play to selected step** behaves like **Play all pending steps**, but it lets you play only a subset of the steps instead of all the steps. In the list, select the step that you want playback to stop at, and then click **Play to selected step**. Task recorder will begin to run the steps in the list and will stop when it has run the step that you selected.

## Editing a recording

Although you can edit a recording through the playback functionality, there is also a mode that lets you make simple edits to a recording without having to replay the whole recording. To access this feature, click **Edit recording** after you open the **Task recorder** pane. You can use this feature to make the following edits:

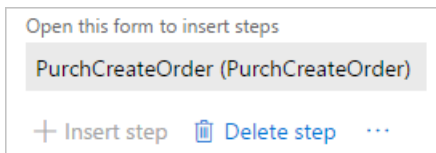
- Insert steps into a recording without re-recording the whole file.
- Move steps under a sub-task without re-recording the whole file.
- Adjust the name and description of the recording.

### Insert steps without re-recording the entire file

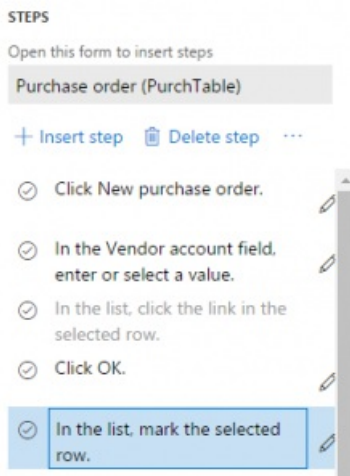
You can add a step anywhere in a task recording without playing back or re-recording the whole file.

1. Select the step after which you want the new step to be inserted. Make sure the step is highlighted.

In order for task recorder to insert a step, you must have the correct page open. The correct page is the page on which the new step occurs. Task recorder has a mechanism that determines what the active page is, and will disable the functionality if the correct page isn't open.



When you are on the correct page, **Insert step** becomes available.



2. Click **Insert step**.

When you click **Insert step**, Task recorder switches to recording mode. Any action that is performed in the user interface (UI) will now be recorded and inserted into the recording as steps.

3. Click **Stop**.

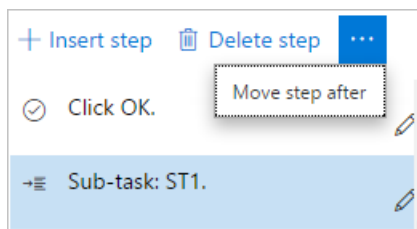
Recording mode is stopped, and you can now continue to edit the recording. For example, you can repeat this process to insert steps in other places in the recording, or you can move sub-tasks as described in the next section.

4. When you've finished editing the task recording, click **Done editing**, and then select one of the options to save or publish the recording.

### Move steps under a sub-task without re-recording the entire file

You can move steps under a sub-task without playing back or re-recording the entire file. You can also move the sub-task step or the end sub-task step if you want to group an existing block of steps.

1. Select the step or sub-task step that you want to move. Make sure that the step is highlighted.
2. Click **Move step after**. To access this command, you might have to select the ellipsis (...) button.



3. Select the step or sub-task step that you want to move the step or sub-task step after. Task recorder will move the step.
4. To move the end sub-task step, select it, click **Move step after**, and then select the step that you want the end sub-task step to be after.

If you want the first step in the task guide to be within a sub-task, create a sub-task step as the second step, and then move the first step into it. You can add or move as many steps or sub-tasks as needed.

5. When you've finished editing the task recording, click **Done editing**, and then select one of the options to save or publish the recording.

### Adjust the recording name and description

You can adjust values of the **Recording name** and **Recording description** fields. If you want to see more steps in the Task recorder editing pane, you can also collapse the section that shows the recording name and description.



## Playing a task guide

A **task guide** is a user-focused experience that lets the user follow a guided step-by-step set of instructions to complete a business scenario by using a task recording. The user is instructed to complete each step through an animated pop-up prompt that will move across the page and point to the UI element that the user should interact with. The prompt will also tell the user how to interact with the element. For example, it might state, "Click here" or "In this field, enter data." Each step that the user is instructed to complete is based on a step that was originally recorded in the task recording. Because the task recording file contains the data that describes the step that was originally recorded, the task guide can automatically determine when the user has completed the step as expected. It then automatically moves on to the next step.

### NOTE

One way that the task guide determines that a user has completed a step is by detecting when the value in a field has changed. Although the task guide doesn't require that a specific value be set, it does require that the field value be changed in order to determine that the step was completed. The user must change the field value, and then press the **Tab** key or click in an area outside the UI element. Only at that point does the client detect that the field value has changed, and it can then proceed to run any required application validation or business logic. Therefore, before the task guide can determine that the step was completed by the user, it relies on the client to detect that the field value has changed.

## What can a task guide allow a user to do?

When a user is completing a task guide, the client behaves in the same manner, with the same data, security, and validation rules as it does when the user is not completing a task guide. There is no difference of behavior in the client that would allow a user to take an action that they cannot otherwise take when they are not completing a task guide. When a user is completing a task guide:

- Any data the user enters is subject to the same data validation rules as when not playing the task guide.
- Any data the user enters may be saved, and the user may modify data according to the same restrictions and rules as when not playing the task guide.
- Any security mechanisms the user encounters behave the same as when the user is not playing the task guide.
- Any forms or controls the user accesses are subject to the same security and access mechanisms as when the user is not playing the task guide.

## The "On-rails" feature of task guides

By default, when a user begins a task guide, they are placed "on-rails". These "rails" prevent the user from *clicking* on elements other than the element the task guide is pointing to. When the user tries to click on something outside of the UI element that the task guide is pointing to, the task guide pop-up will animate to let the user know that they cannot progress until they complete the current step of the task guide.

While a user is prohibited from *clicking* on other elements, the user is not prevented from tabbing through the other controls on the form, and the user is not prevented from using keyboard shortcuts. This is by design, as the "on-rails" feature is designed for and targeted at first-time users, who are expected to primarily use the mouse as they become familiar with the application.

More advanced or experienced users can turn off the "on-rails" feature when they complete a task guide. At any point during the task guide, these users can turn off the rails by clicking the **Unlock** button that appears on the Task recorder toolbar at the top of the page. This button can also be used to restore the rails at any point during the task guide. In some situations, the task guide might automatically turn off the "on-rails" feature. When the rails are turned off, the user can click UI elements just as they do when the task guide isn't running. The "on-rails" feature might be automatically turned off in the following situations:

- The user is being directed to go to a page by using the navigation pane or navigation search.
  - Because the user can use either entry point, the task guide doesn't point to a specific entry point, and it doesn't prevent the user from using either entry point.
- The task guide enters an error state (see the next section for a list of error states).
- The task guide is showing an info step.

## Error detection

An *error state* occurs when the task guide is not able to point to the UI element that is associated with the current step because the UI element is not visible on the screen. When the task guide detects that the current step requires the user to interact with a UI element that is not visible, then the task guide pop-up will move to the upper-right side of the screen. These causes of an error state can be simplified into two categories.

### The control is not visible on the form

*This error state usually occurs when the user has opened or closed the incorrect tab, FastTab, collapsible section, FactBox, or pop-out menu.*

Because the UI element that is needed for the current step is somewhere on the current form, but it is not visible on the screen, the task guide pop-up will simply move to the upper-right side of the screen while displaying the same instruction that informs the user of the action they need to take.

Because the task guide can't find the UI element on the screen, the user must manually determine what is causing the UI element to be hidden and then make the element visible on the screen. The task guide pop-up will automatically detect that the UI element is visible and will reposition itself so that it's pointing at the now-



visible element.

#### **The control is not on the form**

*This error state usually occurs when the user has gone to the wrong form, either by navigating to the wrong form or by leaving the correct form.*

Because the UI element is not visible on the screen, the task guide pop-up will move to the upper-right side of the screen. In addition, when the task guide detects the user is on the wrong form, the task guide pop-up text will change to inform the user of the form they should navigate to.

In some cases, the task guide pop-up will not mention the form by name. This is because the user may need to navigate to a dynamic form. A dynamic form is a form that is not modeled, frequently known as a runtime-generated form. These sorts of forms do not have a proper name. Some examples of runtime-generated forms include simple and custom lookups. The way for a user to navigate to a lookup form is to re-open the lookup.

#### **Next step and Previous step**

The **Next step** and **Previous step** buttons appear in the task guide pop-up and let a user manually control the flow of the task guide. When these buttons are clicked, the task guide will go to the next or previous step. The task guide doesn't verify that the user has completed a step before it goes to the next or previous step.

The task guide **never** automatically completes any step for the user, even when the **Next step** and **Previous step** buttons are used. Use of these buttons can cause an error state if the previous or next step refers to a UI element that isn't on the current page. When the user is completing an info step, the only way to proceed is to use the **Next step** button. This action is required because an info step doesn't represent an action that was recorded on any UI element. Because no action was recorded in the task recording, the task guide doesn't have the necessary information to determine what action it should expect the user to complete.

#### **The See more button**

When the **See more** button is clicked, the task guide pop-up expands to show additional information that is related to the step. The additional information is often optional reading material that isn't required for the user to successfully complete the step. The following information might be included:

- **An Example value**
  - The Example value is the value that was originally used when the task recording was created.
  - Example values appear only for steps that use non-lookup fields. These fields include text fields, number fields, date fields, combo boxes, and check boxes.
- **A Note**
  - A Note may contain scenario-specific information that will help provide context to the user about the current step of the task guide.

## Taking screenshots in Task recorder

By using a **pre-release** Chromium browser extension that works for both the new (Chromium-based) Microsoft Edge browser and Google Chrome, Task recorder can take screenshots of the browser as a user records a business process. After the user completes the recording, Task recorder can use these screenshots to generate Microsoft Word documents. To turn on this functionality, follow these steps to install the pre-release Chromium extension that enables Task recorder to take screenshots during recording.

1. Download the **FMLabTaskRecorderScreenshot** folder that contains the extension from GitHub, at <https://github.com/Microsoft/FMLab>.
2. **On-premises deployments only:** Adjust the manifest for the extension so that it matches the following code. Replace <hostname> with the base URL for your environment.

```

...
"content_scripts": [
  {
    "matches": ["https://*.dynamics.com/*", "<hostname>"],
    "js": ["screenshot.js"]
  }
]
...

```

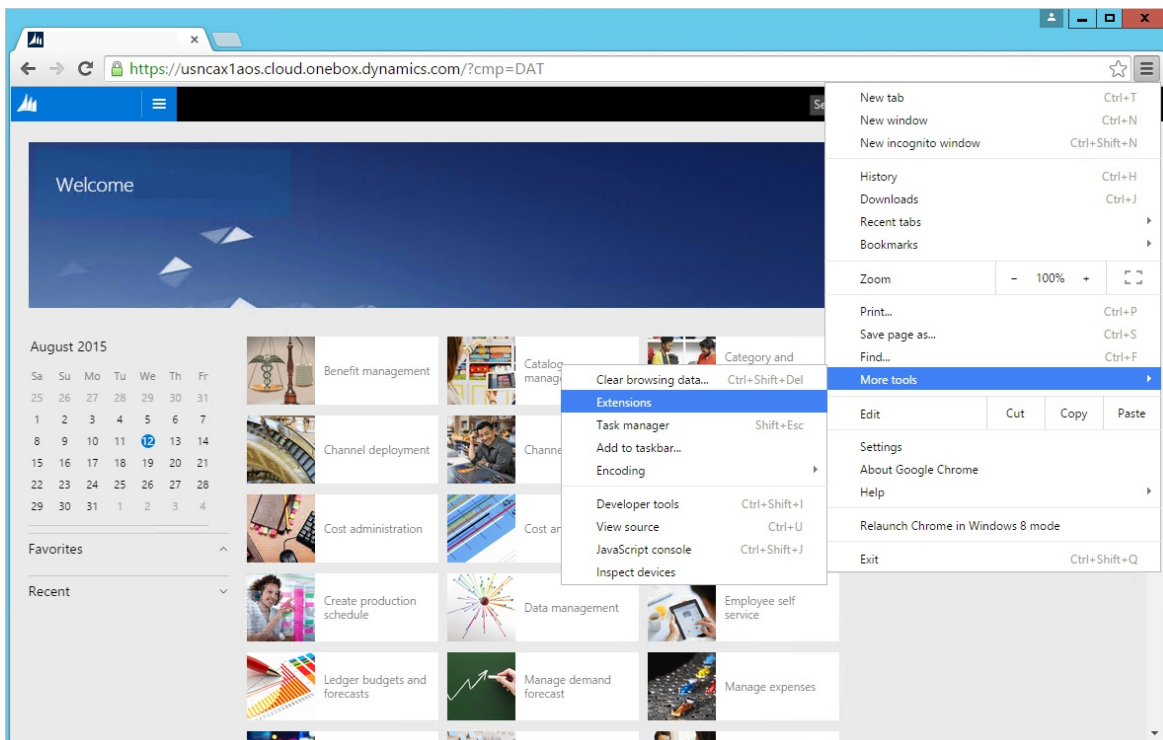
3. **21Vianet deployments only:** Adjust the manifest for the extension so that it matches the following code. Replace `.com` with `.cn`

```

...
"content_scripts": [
  {
    "matches": ["https://*.dynamics.cn/*"],
    "js": ["screenshot.js"]
  }
]
...

```

4. Open the latest Microsoft Edge browser or Google Chrome.
5. Select **Settings and more > Extensions** in Microsoft Edge (or **Customize and control Google Chrome > More tools > Extensions** in Google Chrome).



6. Select **Developer mode**.
7. Click **Load unpacked extension**.
8. Browse to the folder that contains the Task recorder extension by using the path **FMLab-master > FMLab > TaskRecorderScreenshot**, and then select **Select Folder**.
9. Make sure that **Enabled** is selected so that extension is turned on.
10. Restart the browser.

Task recorder will now take screenshots of the tab where the client is running. These screenshots are available for one week after the recording has been played. (If you're running a platform version that is earlier than

Platform update 16, the screenshots are available for only 15 minutes.) If the screenshots have expired, you can regenerate them by playing the task recording again.

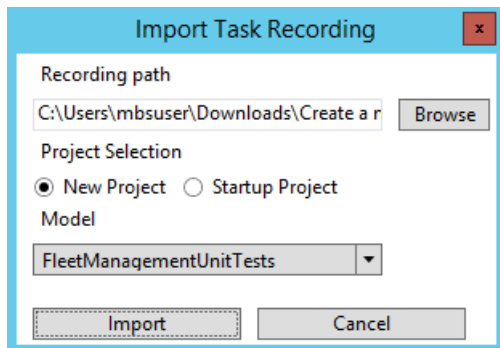
Note that Task recorder **does not** capture screenshots from other tabs or of the user's desktop.

## Generating tests from a recording

After a business process recording has been completed by using Task recorder, a developer can import the raw developer recording file (.xml file) into Visual Studio to create an X++ test. The import tool generates a human-readable X++ test from the recording, and translates any control gestures, validations, or tasks into the appropriate test code.

### Import a recorded test

1. Open Visual Studio by using the Finance and Operations development tools.
2. Go to **Dynamics 365 > Addins > Import task recording**.
3. In the **Import task recording** menu, use the **Browse** button to locate a previously downloaded recording file.
4. Optionally, choose to have the generated test code be added to the startup project. This requires that a solution containing a project is set as the startup project. This will place the generated X++ test into the same model as the project.
5. If you're creating a new project, select the model for the project. The generated X++ test will be put in this model. For the generated test to be successfully built, the model must have references to the **TestEssentials** model.
6. Click **Import**.



7. In the **New Project** dialog box, provide a name for the project.
8. After the project is created, the user can open and inspect the generated code.
9. To run the test, build the project.
10. Go to **Test > Windows > Test Explorer**.

## Appendix

### Controls that are known to have incomplete support for Task recorder

- Table
- Filter pane, which is the filter that pops out from the left side
  - When adding filters to the filter pane, the steps are delayed. The steps do not get recorded until the user clicks "Apply" on the Filter pane.
- Enhanced previews
  - No planned support for recording gestures inside of enhanced previews. While recording, enhanced

previews will be disabled.

- No extensible controls are supported out of the box, except Segmented Entry.
  - Extensible control owners need to individually build support for Task recorder.

#### **Controls that can be recorded, but have limited support for the Copy/Paste/Validate gestures**

- Date/Time
  - Doesn't support copy/pasting "Never" as a value.
- Image
  - No ability to copy/paste/validate an image value.
- Filter pane
  - Copy/Paste works, but the UI will not show the pasted data. You can proceed as if it pasted correctly.
- Message box
  - You cannot validate the text in the message box.

#### **Controls that are known to have incomplete support for being used in a task guide**

- Quick Filter, which is the filter control that appears above lists
  - Does not support displaying a "generic value" during the task guide. Currently displays the value that was used during recording.
- Filter pane, which is the filter that pops out from the left side
  - The task guide does not point to the individual elements within the Filter pane that need to be clicked on.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create documentation or training with Task Recorder

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains what Task recorder and task guides are, how to create task recordings, and how to customize Microsoft task guides and include them in your Help.

## IMPORTANT

You can record your own task guides for Dynamics 365 Human Resources, but you won't be able to save them to a Business Process Modeler (BPM) library or open them from the Help pane at this time. You can save them locally or to a network location, and then open and replay them using Task recorder.

## Learn about Task recorder

Task recorder is a tool that you can use to record actions that you take in the product user interface (UI). When you use Task recorder, all of the events that you perform in the UI that are executed against the server—including adding values, changing settings, removing data—are captured. The steps that you record are collectively called a *task recording*. Task recordings can be used in many ways:

- **Task recordings can be played as task guides.** Task guides are an integral piece of the Help experience. A task guide is a controlled, guided, interactive experience through the steps of a business process. The user is instructed to complete each step by way of a pop-up prompt (or "bubble"), which will animate across the UI and point to the UI element that the user should interact with. The "bubble" also provides information about how to interact with the element, such as "Click here" or "In this field, enter a value." A task guide runs against the user's current data set and the data that is entered is saved in the user's environment.
- **Task recordings can be saved as Word documents.** This allows you to easily produce printable training guides.

You can create your own task recordings, play task recordings provided by Microsoft, or modify Microsoft-provided task recordings to reflect your configuration. For more information about Task recorder, see [Task recorder](#).

## Plan your task recording

Whether you're creating a new task recording or basing your recording on a Microsoft task recording, keep the following information in mind.

- Plan your recording like you would a video. Make all your decisions ahead of time.
- Walk through the business process once or twice without recording it to understand the steps.
- When you walk through the process before you record, notice where you use shortcut keys or the **Enter** key, so that you can avoid using them during the actual recording.
- Identify the following:
  - Do you want to group steps together into sub-tasks? Sub-tasks visually set apart sections of a process. For example, if you are creating a recording for "Creating and releasing a product," you may want to group together the steps that are required to create a product, and then group together the steps that are required to release the product. Sub-tasks also make longer processes easier to read.

- Do you want to add annotations, and if so, where? See "Understand the different types of annotations" below for more information.
- What values will you add in the various fields as you complete the steps of the business process? It is a good idea to know what you'll select or enter as you proceed so that you don't backtrack or fix mistakes as you're recording.

### **Write your description and annotations ahead of time**

- At the beginning of each task recording, there's a description field that allows you to enter an introduction to the recording. It is a good idea to write and save the description ahead of time in a separate document so you can copy and paste it into the task recording when you are recording. That way, you can spend time refining the text when you aren't in the process of recording. Cutting and pasting the text makes the recording process go more quickly and smoothly.
- For each step in a task recording, you can create annotations. During playback of a task guide, annotations appear in the "bubble" as notes above or below the text for the step. When viewed as text in the Help pane, annotations appear as text inline in the step. As with the description, it is a good idea to write and save your annotations in a separate document. When you're recording the task recording, cut and paste the annotations in from that document.

**Understand the different types of annotations** All annotations are optional. Only add them when they'll provide helpful information to the user.

- **Title:** A title annotation will appear before the step text that task recorder automatically generates. In the task guide, the title annotation appears above the automatically generated text. Use this type of annotation to explain why the user is doing the step or to give additional context.

This is the editing pane that you see when you add an annotation as you create your recording. Enter a title annotation in the **Title** box.

Task recorder ×

Hide from task guide  
No

Step instruction

Click OK.  
 { standard example text }  
 { your example text }

Title

Double-check the information before proceeding. You are about to commit the new product to the system.

Notes

This is what the title annotation looks like in the "bubble" in the task guide.

**New product**

Product type:  Product name:

Product subtype:  Search name:

**IDENTIFICATION**

Product number:  Retail category:

**CATCH WEIGHT**

No

Double-check the information before proceeding. You are about to commit the new product to the system.

Click OK.

[Previous step](#) [Next step](#)

- **Notes:** A notes annotation will appear after the step text that task recorder automatically generates. In the task guide it will only be visible if the user clicks the **Show more** link in the task guide bubble. Use this type of annotation to describe anything that a user needs to know to complete the step.

This is the editing pane that you see when you add an annotation as you create your recording. Enter a notes annotation in the **Notes** box.



Task recorder ✕

Hide from task guide  
No

Step instruction

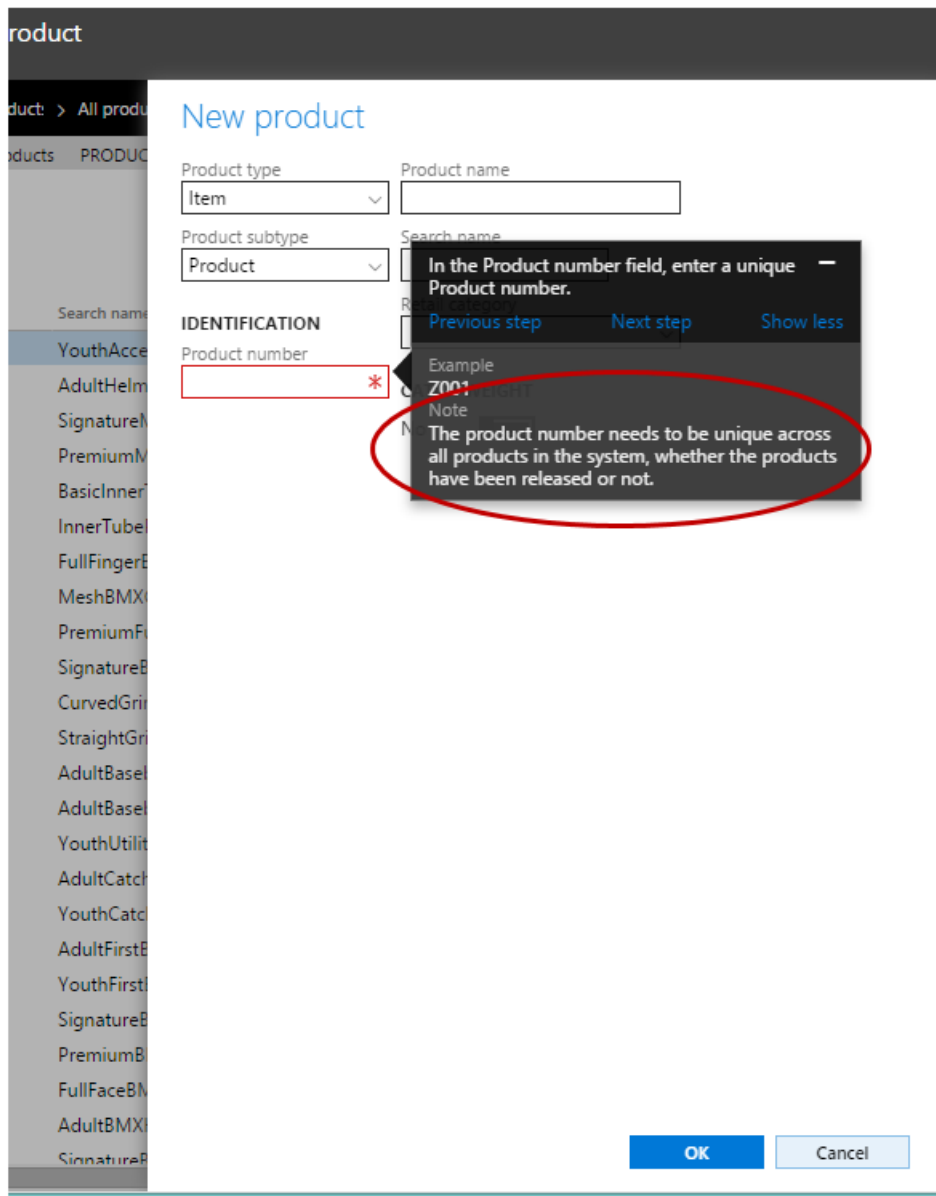
- In the Product number field, ...
- In the Product number field, ...
- In the Product number field, ...

Title

Notes

The product number needs to be unique across all products in the system, whether the products have been released or not.

This is what the notes annotation looks like in the "bubble" in the task guide.



- **Info step:** These annotations are created by right clicking on a control or anywhere on a form < **Task recorder** < **Add info step**. Info steps appear as a numbered step at whatever point you insert it, even though no action was recorded in the UI. You can add a form-level info step or an info step associated with a control. When an info step is associated with a form, the task guide “bubble” will appear someplace on the form, with no pointer; when the task guide is played. When an info step is associated with a control, the task guide “bubble” will point to the control when the task guide is played. In the Help pane, an info step annotation will appear as a numbered step with whatever text you entered. Use info steps to prepare the user for the next steps, to describe steps that need to be done outside of the application, or to refer to other recordings (although you cannot create hyperlinks in annotations).

### Determine how long to make your recording

- The user will generally either read or play the recording from start to finish, so don't combine steps or tasks that are better done separately.
- Try not to record a long scenario that spans multiple sub-processes. For example, “Operate in-store customer service desk” is too broad; break it up into shorter tasks such as “Accept returns” and “Add to gift card.”
- If a task can be carried out as part of several different business processes, create a separate recording for it, and you can refer to it in the other recordings.
- If the process involves multiple tasks that the person likely does all at once, you can keep the tasks in one

recording, for example, "Set up and assign functionality profiles."

- If it is something someone does once (such as configuration) and then another task that they can do immediately afterward but may do repeatedly, and on its own, break them up into two task recordings.

**Decide where, in the UI, to start a recording** The page that you are on when you start recording a task recording affects which page the task guide is displayed for. For example, if you want your task recording to be listed in the Help pane when a user clicks Help on the General ledger parameters page, you must start your recording on the General ledger parameters page. **Save recordings as .axtr files** When you are done creating or editing a task recording, you are presented with several options for how you want to download, or save the recording. You can download the file as a task recording package (.axtr), download it as a raw recording file (.xml), download it as a Word document, or save the file to an LCS library. It is a good idea to always save your task recording as a task recording package file (.axtr). This will help make maintenance of the file easier if procedures or annotations need to change later. If you want to download the file as a Word document, also save it as a task recording package file.

## Create your task recording

For detailed walk-through steps, see [Task recorder resources](#).

## Copy and customize Microsoft's task recordings

You can download and edit Microsoft's task recordings to use them for your own Help documentation or training materials. To download a Microsoft task recording, follow these steps:

1. Open Task recorder. Task recorder is located in the **Settings** menu.
2. In the Task recorder pane, click **Maintain a recording**.
3. Under **Where is the recording**, click **It is in an LCS library**.
4. Click **Select the LCS library**.
5. Select the Microsoft global library.
6. In the tree, select the business process library node that the task recording is associated with.
7. Click **OK**.
8. Click **Start**.
9. At this point, step through the recording, changing any steps as you go to re-record it. **Note:** If you only need to change the text of a recording, you can open the recording in **Edit a recording's annotations** mode, and then save it.
10. After the recording has played to the end, click **Stop** in the task recorder bar at the top of the screen.
11. Choose how you want to save the task recording.

## Additional resources

[Help system](#)

[Connect the Help system](#)

[Task Recorder](#)

[Create Rich Help Topics with Task Recorder \(external link\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Videos

2/18/2021 • 2 minutes to read • [Edit Online](#)

On the [Microsoft Dynamics 365 YouTube channel](#), you can find videos created by Microsoft that demonstrate a wide range of business solutions for Dynamics 365 products. This section lists the "how-to" videos for Finance and Operations apps that are hosted on the channel.

## NOTE

Some of the videos listed were published under a previous product name, but are still applicable.

## Videos for Finance and Operations development and administration

### Data integration

- [Prospect to cash integration](#)
- [Synchronize a work order between Dynamics 365 for Field Service and Finance and Operations apps](#)

### Data management

- [Use data entities and data packages](#)

### Development

- [How to resolve conflicts in Visual Studio](#)
- [Optimization advisor](#)
- [Setting up a development branch and build](#)
- [Setting up a development machine](#)
- [Azure DevOps integration with Lifecycle Services](#)

### Intelligence

- [How to edit an embedded report in an analytical workspace](#)
- [How to embed PowerBI.com reports in Finance and Operations apps](#)
- [Bring your own database \(BYOD\) to Finance and Operations apps](#)
- [How to use cost management Power BI content in Dynamics 365](#)

### Lifecycle Services (LCS)

- [Asset library in Lifecycle Services](#)
- [Cloud-hosted environments](#)
- [Creating support tickets from Dynamics 365 for Operations](#)
- [Deploying environments](#)
- [Getting started with Lifecycle Services](#)
- [Deploying code to a sandbox environment](#)
- [Deploying code to a production environment](#)
- [Implementation projects in Lifecycle Services](#)
- [Manage the code upgrade and tool process](#)
- [Managing business process libraries in Lifecycle Services](#)
- [Methodologies in Lifecycle Services](#)
- [Request a production environment](#)

- [Refreshing data in a production environment](#)
- [Uptake a new platform release after Platform update 3](#)
- [Use telemetry to monitor key performance counters in Dynamics Lifecycle Services](#)

#### **Regression suite automation tool (RSAT)**

- [How to use task recorder to create a test case for the Regression suite automation tool \(RSAT\)](#)
- [How to create a test plan in Azure DevOps to use with the Regression suite automation tool \(RSAT\)](#)
- [How to use the Regression suite automation tool \(RSAT\)](#)
- [The improved Excel experience in Regression Suite Automation Tool \(RSAT\) 2.0](#)

## Videos for Dynamics 365 Finance

#### **Customize the app**

- [Add custom fields](#)
- [Embed Power Apps](#)

#### **Financial reporting**

- [Find the version of Report designer](#)

#### **Help system**

- [How to contribute to the Microsoft Dynamics 365 documentation](#)

#### **Office integration**

- [Create an Excel template for header and line patterns](#)

#### **Organization administration**

- [Document management](#)

#### **Revenue recognition**

- [How to use revenue recognition](#)

#### **Tax engine**

- [Tax engine overview](#)

## Videos for Dynamics 365 Supply Chain Management

#### **Costs**

- [Cost control mobile workspace](#)
- [Get started with Cost accounting](#)
- [Use Excel for cost analysis](#)

#### **Customer portal**

- [Overview of the Customer portal template](#)
- [Invite customers to register and use your customer portal](#)

#### **Help system**

- [How to contribute to the Microsoft Dynamics 365 documentation](#)

#### **Master planning**

- [Master planning setup wizard](#)

#### **Office integration**

- [Create an Excel template for header and line patterns](#)

## Organization administration

- [Document management](#)

## Procurement and sourcing

- [Approve purchase orders on a mobile device](#)
- [Onboard a new vendor](#)

## Product information management

- [Change management capabilities](#)

## Production control

- [Batch balancing](#)
- [Enhancements to the production order release process](#)
- [Visual scheduling with Gantt chart for production and batch orders](#)

## Warehouse management

- [Release production picking to the warehouse in batch](#)
- [Use warehouse template to copy configuration](#)

# FastTrack Tech Talks

[FastTrack Tech Talks](#) focus on providing technical depth and best practices that provide customers and partners with detailed knowledge that's specific to the subject areas.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Glossary

2/18/2021 • 15 minutes to read • [Edit Online](#)

This glossary defines key terms and concepts in Finance and Operations apps.

## A

accounting journal

A journal that is used to record the financial consequences of accounting events in an accounting system.

accounting rule

A rule in an accounting system that controls the principles, methods, and procedures for classifying, recording, and reporting the financial consequences of accounting events.

ACH (Automated Clearing House)

An electronic funds transfer system available in the United States that facilitates the transfer of funds between receiving party and originating party bank accounts.

actual quantity

The measured quantity that is input into or output from an activity.

address verification

The service that is provided by a credit card processor that validates that the billing address provided by a card holder matches the information on file with the issuing bank.

aging

The process of classifying time periods by age.

aging period

The number of days in a time period used to report an overdue customer payment balance.

## B

balance sheet

A statement of the financial position of an organization that reports the state of assets, liabilities, and equity on a specified date.

balance sheet account

An account that describes the changes in value reported on a balance sheet.

bank reconciliation

A practice of reconciling a ledger account that represents a bank account by matching ledger account entries to bank statement entries.

batch attribute

A product attribute of a process batch or a transfer batch.

best-before date

A recommended date for obtaining the optimum quality or characteristic of a product.

best-before period

The time period in which to obtain the optimum quality or characteristic of a product.

bill of exchange

A source document that documents an unconditional request for a third-party to pay a second party on demand.

blocking

The action of placing a document or product on hold.

BOM (bill of materials)

A list of products and their quantities that are required to produce one product.

budget control

A practice of authorizing expenditure only when budget funds can be reserved to meet future payment commitments.

budget control dimension

A combination of active financial dimensions values used to allocate budget funds to pay for planned activities.

budget control rule

The encoding of a business decision to check committed and actual expenditure against available budget funds allocated for detailed or aggregate activities defined by valid budget control dimension value combinations.

budget cycle time span

A time period specified as a number of fiscal calendar periods. The budget period can be different from the accounting period.

budget group

A set of financial dimension values in a budget dimension hierarchy that is used to calculate aggregate budget funds allocated to superordinate financial dimension values by summing budget funds allocated to subordinate financial dimension values.

budget model

A planning structure used to schedule budget fund allocations and expenditures.

bulk item

A formula item input into a product delivery activity.

bundle

The combination of a number of products for sale as one unit.

business unit

A semi-autonomous operating unit that is created to meet strategic business objectives.

## C

capacity load

The maximum amount of scheduled work that a work center can perform at a required capacity level.

capacity planning

A procedure for determining the resource capacity requirements that meet the demand of future output during specific time periods.

carry-forward budget

The budget that is transferred from one fiscal year to the next and that is reserved for open purchase orders in the new fiscal year.

cash-generating unit

The smallest group of classifiable assets that generates cash independently of other assets within an organization. These groups of assets are used to measure impairment losses that will be incurred.

catch weight

The actual or nominal weight of a sales item or inventory item.

chart of accounts

A list of main accounts.

COGS (cost of goods sold)

An accounting category used to sum the financial consequences of manufacturing products and carrying inventory.

contractor

A role assumed by a person who participates in a contractor-employer relationship with a legal entity.

co-product

An item produced jointly with another item.

cost center

An operating unit whose managers are accountable for budgeted and actual expenditures.

cost variance

The difference between an expected cost and an actual cost.

costing



The process of calculating, assigning, and allocating the cost of economic resources acquired, produced, or delivered by an organization.

CTP (capable-to-promise)

The portion of product output from available operations resources and available input product required to fulfill a specific customer requirement.

currency code

An alphanumeric identifier that represents a currency unit.

cycle time

The time taken to complete an activity.

## D

dashboard

The typical start page in Finance and Operations apps. On the dashboard, users see a section for each workspace that they have access to. Each section displays the tiles from the summary sections in the related workspace. The dashboard consists of a name and sections with tiles.

data entity

An information structure that represents the data characteristic of an entity.

delegate

A delegate is a type that represents references to methods with a particular parameter list and return type.

delivery note

A business document that documents the delivery of products between two parties.

demand forecast

A prediction of future product demand.

demand forecasting

A business process that estimates future demand and creates demand forecasts based on historical transaction data.

deployable package

The vehicle used for deployment on User Acceptance Test (UAT) and production environments.

designer

A Visual Studio tool that you used to create, update, and inspect your model elements.

dimension-based product configuration

A configuration technology used to create product variants by selecting values for product dimensions.

distinct product

A uniquely identifiable product.

document management system

An application service for storing and handling an organization's documents.

duty

In the security model, a set of application access privileges that are required for a user to carry out their responsibilities.

## E

EFT (electronic funds transfer)

A networked system for transferring funds from one bank account to another.

event

An event lets a class or object to notify other classes or objects when something of interest occurs.

EPE (Every Product Every)

A lean concept that is used to establish a regular repeating production cycle.

extension

Customizing an application by adding functionality to existing code.

## F

### financial dimension

A financial data classifier created from the parties, locations, products, and activities in an organization and used for management reporting.

### financial dimension value

A data element in the domain of a financial dimension.

### financial statement

A report that documents the financial information and financial position of an organization.

### fixed cost

A cost that does not vary with changes in product delivery throughput or output.

### fixed currency

A currency that has a fixed exchange rate in relation to another currency.

### fixed quantity kanban

A type of kanban that is used when the number of kanbans that are assigned to a kanban rule is constant.

### formula

A numeric relationship among production process inputs and production process outputs.

### formula item

An output of a batch process controlled by a formula.

## G

### general budget reservation

A document that is often used by public sector entities to set aside or earmark budgeted funds so that those funds are not available for other purposes.

### grace period

The time period beyond a specified date during which an obligation can be fulfilled without penalty.

### GST (goods and services tax)

A value-added tax levied in some countries/regions.

## I

### intercompany

Occurring between or relating two or more legal entities that are part the same organization that consolidates the accounts of all legal entities.

### invoice matching

A practice of matching vendor invoice prices and product quantities to purchase orders and product receipts.

### item allocation key

A product family grouping that is used for forecast and demand scheduling.

### item relation

A reference to the item allocation group or the item and its product dimensions in a kanban rule.

## K

### kanban

A signal that communicates a requirement for a quantity of product.

### kanban flow

Defines the sequence of activities that are performed for kanbans that are created for a kanban rule.

### kanban job

A process or transfer activity in a production flow that is triggered by a Kanban.

### kanban job consumption

The withdrawal of product components from inventory in order to complete the kanban production jobs.

### kanban rule

A rule in a lean manufacturing system that realizes material planning and replenishment policies by controlling

how process and transfer activities are coordinated in production flows.

## L

lean manufacturing

A philosophy in which manufacturing operations promote lean production flows and business activities.

lean schedule group

A way in which to aggregate items for production, for example, based on a setup group, shipping group, or transport group.

ledger account

A classifier created from the combination of main account value and other financial dimension values listed in a chart of accounts and used to classify the financial consequences of economic activity.

license code

An alphanumeric key that grants a party the right to use software or software components as prescribed in the terms of the license agreement, and that activates and inactivates software modules, software capabilities, and software functions.

liquidity

The ability of a party to use current assets to settle current liabilities.

## M

master scheduling

The process for generating a timetable for matching supply with demand.

model

A model contains code elements and reference metadata. It is created and inspected in Visual Studio.

model element

A model is a group of elements (source files and metadata) that constitutes a distributable software solution. The model is a design-time concept.

## N

NBV (net book value)

The value of a fixed asset calculated as the difference between the original cost of the fixed asset minus its accumulated depreciations.

## O

object permission

A permitted create, read, update, delete, or execute operation on a securable object.

operating unit

An organization that divides the control of economic resources and operational processes among people who have a duty to maximize the use of scarce resources, to improve processes, and to account for their performance.

output product

The physical products that result from an activity.

overlaying

A method of customizing source code by providing new source code that overrides the default source code.

## P

package

A package is a compilable and deployable unit of one or more models.

party

A person or organization that participates in economic activities.

pegging

The process of tracing the quantity of a required item to its source.

pegging event

A resource flow event that signals the demand for a product.

planned intercompany demand

Predicted demand for a product by a legal entity that assumes the role of a vendor. The planned demand is generated from planned demand for the product from a downstream legal entity.

post

To record the monetary value of an economic event in a specific account, or to summarize and reclassify general and subsidiary journal account entries into general and subsidiary ledger account entries.

preparer

The person who creates a source document to initiate a request for economic resources.

procurement catalog

A listing of product offerings that are grouped by procurement category. A procurement catalog is used to request products for internal use by an organization.

product dimension

The size, color, or configuration product attributes that are used for dimension-based product configuration.

product family

A unique grouping of items, services, or rights that either participates in the same production or delivery activities or that are offered to the same market segments. The grouping is represented by using a forecast allocation key.

product master

A standard or functional product representation that is the basis for configuring product variants.

product receipt

A source document that documents the receipt of products ordered, the receipt of products returned, or the receipt of products received on consignment.

product variant

A configuration of a product master.

production flow

A production process designed using Lean principles.

production flow model

A representation of the production capacity provided by a group of work cells in a production flow process.

production order

A source document that documents the requirements for producing items to meet a demand.

production schedule

A schedule to produce a specific item and item quantity at a specific time and by specific human and operational resources.

prospect

A participant that has the existing and potential ability to provide a service or probable future economic benefit to a legal entity.

purchase agreement policy

A policy that authorizes parties to modify purchase agreement terms.

purchase order policy

A policy that authorizes parties to modify purchase order terms and to control order processing.

purchase quotation

A source document that documents an offer to purchase a quantity of product for a specified price and by a specified date in response to a request for quotation in a procurement process.

purchase requisition

A source document that documents product requests so that they can be submitted for review and be used to authorize purchasing by a purchasing organization.

## Q

quantity variance

The difference between an ordered product quantity and a received product quantity when no more receipts are expected.

## R

receipt advice

A business document that documents a summary of the products a buyer receives from a vendor.

reconciliation

A practice of adjusting two or more accounts or statements so that the figures agree.

reduction key

A method that is used to increase or decrease forecast requirements in master planning, based on user defined percentages that are applied during specific periods.

register

A record that is used to record the operational, legal, and financial consequences of resource flow events in an accounting system.

RFQ (request for quotation)

A source document that documents an invitation to bid on supplying a quantity of product for a specified price and by a specified date.

requester

The person who requests the economic resources.

## S

sales agreement

A source document that documents an agreement between two or more parties based on an understanding that a selling party will commit to selling a specific quantity or value of product over a period of time in exchange for favorable prices and discounts.

sales agreement policy

A policy that authorizes parties to modify sales agreement terms.

sales order policy

A policy that authorizes parties to modify sales order terms and to control order processing and payment processes.

sales quotation

A source document that documents an offer to supply a quantity of product for a specified price and by a specified date in response to a request for quotation in a sales process.

security role

A defined set of application access privileges. The security role assigned to a user determines which tasks the user can perform and which parts of the user interface the user can view. All users must be assigned at least one security role in order to access the system.

segregation of duties

A design principle used to reduce the risk of fraud, irregularities, and errors that separates the recording, verification, authorization, custody of assets, and periodic review duties of people who participate in, document, or record the financial consequences of economic transactions.

sequencing

The order in which jobs are processed or operations are performed at a manufacturing facility to achieve objectives.

shared asset

An asset that is used by more than one cash-generating unit (CGU). An example is a distribution center that is used to store items before they are transported to different markets that share that same distribution center.

single use kanban

A type of kanban that is used with a fixed quantity kanban rule to meet exceptionally high demand. A single use kanban does not trigger a new kanban when it is discarded.

smart rounding

A marketing practice to use odd numbers that are marginally less than their nearest round number to set prices.

source requirement

The product quantity documented on a source document line that creates a pegging requirement.

statistical baseline forecast

An estimate of future demand that is created by applying a forecasting algorithm to historical transaction data.

## T

takt time

The time that it takes to produce one unit of a product.

Task guide

A controlled, guided experience through the steps in a Task recording. A Task guide leads the user through the experience that was recorded. All security, data, and application behaviors are the same when completing a Task guide as they are without the Task guide. The Task guide uses the same instrumentation as Task recorder to know when a user has completed the intended step, so that it can prompt the user to take the next step in the recording.

Task recorder

A tool that is pre-installed in Finance and Operations apps. When recording, it records all events that the user enters in the user interface that get executed against the server—including values added, settings changed, data removed, etc.

Task recording

A file that contains the actions and annotations that are captured when Task recorder is run.

three-way matching policy

A matching policy that requires one or more vendor invoice prices to match with one or more purchase order prices and that requires one or more vendor invoice quantities to match with one or more product receipt quantities.

transfer batch

The quantity of one or more items that is transferred or that can be transferred.

two-way matching policy

A matching policy that requires one or more vendor invoice prices to match with one or more purchase order prices.

## V

value stream

An operating unit that controls one or more production flows.

variant configuration technology

A method of modeling product masters and searching for product variant configurations.

VAT (value-added tax)

A tax on products at each stage of their production based on the value added during that stage.

vendor catalog

A listing of product offerings that are available for purchase from a vendor.

vendor invoice

A source document that documents a vendor payment request. A vendor invoice can refer to one or more purchase orders. When the vendor invoice is authorized, a payment can be made to the vendor.

## W

work cell

A resource group that participates in a production flow activity.

worker

A person who assumes the role of an employee or a contractor and is paid in exchange for services.

workspace

A page that provides an overview of one of the activities that the user performs. The page uses tiles, lists, and charts to display pending work and related data from multiple sources, and the page surfaces frequent tasks related to this data. Tiles shown in the summary section of the workspace are also displayed in the related section in the dashboard.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Commerce Learning catalog

2/18/2021 • 2 minutes to read • [Edit Online](#)

Find the right online training, in person workshops, and events for your role as a user of Dynamics 365 Commerce applications.

## **Business and Technical Decision Makers**

Do you decide whether to invest in new technologies?

[Business and Technical Decision Makers Learning Catalog](#)

[Get started](#)

## **Business users**

Did you just get a new application to use?

[Business Users Learning Catalog](#)

[Get Started](#)

[Commerce](#)

[e-Commerce](#)

## **Implementation Project Managers**

Are you in charge of making sure your company's implementation goes smoothly?

[Implementation Project Manager Learning Catalog](#)

[Get started](#)

## **Administrators**

Do you need to keep systems and data flowing, provisioned, and secure round-the-clock?

[Administrators Learning Catalog](#)

[Get started](#)

[Commerce](#)

[e-Commerce](#)

## **Developers**

Do you need to write code to integrate with other data sources, extend core system functionality, or build a complex application?

[Developer Learning Catalog](#)

[Get started](#)

[Commerce](#)

## **Functional consultants**

Are you an implementation expert for a business domain?

[Functional Application Consultant Learning Catalog](#)



[Get started](#)

[Commerce](#)

[e-Commerce](#)

### **Partner Sales and Marketing**

Are you responsible for helping your customers buy the right solution?

[Microsoft Partner Sales and Marketing Learning Catalog](#)

[Get started](#)

### **Solution architects**

Do you design solutions that meet your customers' needs and budgets?

[Solution Architects Learning Catalog](#)

[Get started](#)

[Commerce](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning catalog for business and technical decision makers considering Dynamics 365 Commerce

2/18/2021 • 2 minutes to read • [Edit Online](#)

Do you decide whether to invest in new technologies?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs.

For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours
<a href="#">Get started with Dynamics 365 Commerce</a>	Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.	Free, self-paced online learning path	2 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.	Free, self-paced online learning path	35 minutes

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning catalog for users of Dynamics 365 Commerce

2/18/2021 • 3 minutes to read • [Edit Online](#)

Did you just get a new application to use?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs.

For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours
<a href="#">Get started with Dynamics 365 Commerce</a>	Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.	Free, self-paced online learning path	2 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.	Free, self-paced online learning path	35 minutes

## Commerce

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Configure and work with omnichannel prerequisites in Dynamics 365 Commerce</a>	Before any of the Dynamics 365 Commerce channels can be created and used, there are several prerequisites that must be set up. This learning path describes the configuration of those prerequisites to enable successful transacting in the Commerce channels, and how to work with them.	Free, self-paced online learning path	6.5 hours
<a href="#">Configure and use Point of sales (POS) in Dynamics 365 Commerce</a>	Transactional data is created in the Point of sale (POS) and then uploaded to the Dynamics 365 Commerce Headquarters for processing. This learning path describes how to install, and activate POS, as well as how to create daily transactions in POS.	Free, self-paced online learning path	4.5 hours
<a href="#">Manage customer engagement and shopping experience in Dynamics 365 Commerce</a>	Managing customer engagement and improving the shopping experience helps consumers make better purchase decisions and build loyalty for the retailer's brands which ultimately leads to increased revenue.	Free, self-paced online learning path	4.5 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Configure and work with call centers in Dynamics 365 Commerce</a>	Dynamics 365 Commerce is an omnichannel solution that provides a call center channel with capabilities to default specific order processing settings and data defaults to the sales orders created by call center users.	Free, self-paced online learning path	2 hours
<a href="#">Manage order fulfillment and inventory in Dynamics 365 Commerce</a>	Efficient inventory management plays an important role in the retail industry, where the ability to have an updated view of the current stock and easily transfer and register goods, can have a great impact on the retailer's success in serving customers and increasing sales revenue.	Free, self-paced online learning path	2 hours

## e-Commerce

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Configure and work with e-Commerce in Dynamics 365 Commerce</a>	The online channels and the e-Commerce site of Dynamics 365 Commerce enable retailers to create an online presence of their brand.	Free, self-paced online learning path	4 hours

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning catalog for implementation project managers working with Dynamics 365 Commerce

2/18/2021 • 2 minutes to read • [Edit Online](#)

Are you in charge of making sure your company's implementation goes smoothly?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs.

For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours
<a href="#">Get started with Dynamics 365 Commerce</a>	Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.	Free, self-paced online learning path	2 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.	Free, self-paced online learning path	35 minutes

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Learning catalog for Dynamics 365 Commerce Administrators

2/18/2021 • 2 minutes to read • [Edit Online](#)

Do you need to keep systems and data flowing, provisioned, and secure round-the-clock?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs. This catalog only includes features specific to Dynamics 365 Commerce. For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours
<a href="#">Get started with Dynamics 365 Commerce</a>	Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.	Free, self-paced online learning path	2 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.	Free, self-paced online learning path	35 minutes

## Commerce

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Deploy and extend Dynamics 365 Commerce</a>	This learning path describes how to deploy Dynamics 365 Commerce environments and perform post-provisioning steps, for successful Commerce implementations using the correct topology for each retailer's business needs.	Free, self-paced online learning path	5 hours
<a href="#">Configure and use Point of sales (POS) in Dynamics 365 Commerce</a>	Transactional data is created in the Point of sale (POS) and then uploaded to the Dynamics 365 Commerce Headquarters for processing. This learning path describes how to install, and activate POS, as well as how to create daily transactions in POS.	Free, self-paced online learning path	4.5 hours
<a href="#">Configure and work with call centers in Dynamics 365 Commerce</a>	Dynamics 365 Commerce is an omnichannel solution that provides a call center channel with capabilities to default specific order processing settings and data defaults to the sales orders created by call center users.	Free, self-paced online learning path	2 hours

## e-Commerce

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Configure and work with e-Commerce in Dynamics 365 Commerce</a>	The online channels and the e-Commerce site of Dynamics 365 Commerce enable retailers to create an online presence of their brand.		

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning catalog for Dynamics 365 Commerce developers

2/18/2021 • 2 minutes to read • [Edit Online](#)

Do you need to write code to integrate with other data sources, extend core system functionality, or build a complex application?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs.

For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Commerce</a>	Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.	Free, self-paced online learning path	2 hours
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.	Free, self-paced online learning path	35 minutes

## Commerce

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Deploy and extend Dynamics 365 Commerce</a>	This learning path describes how to deploy Dynamics 365 Commerce environments and perform post-provisioning steps, for successful Commerce implementations using the correct topology for each retailer's business needs.	Free, self-paced online learning path	5 hours

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning catalog for Commerce Microsoft Partner sales and marketing roles for Dynamics 365 Commerce

2/18/2021 • 2 minutes to read • [Edit Online](#)

Are you responsible for helping your customers buy the right solution?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs.

For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Commerce</a>	<p>Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.</p>	<p>Free, self-paced online learning path</p>	<p>2 hours</p>
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	<p>Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.</p>	<p>Free, self-paced online learning path</p>	<p>35 minutes</p>

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning catalog for Dynamics 365 Commerce functional consultants

2/18/2021 • 3 minutes to read • [Edit Online](#)

Are you an implementation expert for a business domain?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs.

For functional consultants, our job-task analysis research showed that everyone must understand the core content set as well as a domain area.

For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours



CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Commerce</a>	Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.	Free, self-paced online learning path	2 hours
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.	Free, self-paced online learning path	35 minutes

## Commerce

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Deploy Dynamics 365 Commerce</a>	This learning path describes how to deploy Dynamics 365 Commerce environments and perform post-provisioning steps, for successful Commerce implementations using the correct topology for each retailer's business needs.	Free, self-paced online learning path	5 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Configure and work with omnichannel prerequisites in Dynamics 365 Commerce</a>	Before any of the Dynamics 365 Commerce channels can be created and used, there are several prerequisites that must be set up. This learning path describes the configuration of those prerequisites to enable successful transacting in the Commerce channels, and how to work with them.	Free, self-paced online learning path	6.5 hours
<a href="#">Configure and use Point of sales (POS) in Dynamics 365 Commerce</a>	Transactional data is created in the Point of sale (POS) and then uploaded to the Dynamics 365 Commerce Headquarters for processing. This learning path describes how to install, and activate POS, as well as how to create daily transactions in POS.	Free, self-paced online learning path	4.5 hours
<a href="#">Manage customer engagement and shopping experience in Dynamics 365 Commerce</a>	Managing customer engagement and improving the shopping experience helps consumers make better purchase decisions and build loyalty for the retailer's brands which ultimately leads to increased revenue.	Free, self-paced online learning path	4.5 hours
<a href="#">Configure and work with call centers in Dynamics 365 Commerce</a>	Dynamics 365 Commerce is an omnichannel solution that provides a call center channel with capabilities to default specific order processing settings and data defaults to the sales orders created by call center users.	Free, self-paced online learning path	2 hours
<a href="#">Manage order fulfillment and inventory in Dynamics 365 Commerce</a>	Efficient inventory management plays an important role in the retail industry, where the ability to have an updated view of the current stock and easily transfer and register goods, can have a great impact on the retailer's success in serving customers and increasing sales revenue.	Free, self-paced online learning path	2 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Configure and work with e-Commerce in Dynamics 365 Commerce</a>	The online channels and the e-Commerce site of Dynamics 365 Commerce enable retailers to create an online presence of their brand.	Free, self-paced online learning path	4 hours

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning catalog for Dynamics 365 Commerce solution architects

2/18/2021 • 2 minutes to read • [Edit Online](#)

Do you design solutions that meet your customers' needs and budgets?

The following catalog is organized from core knowledge to specific domains, and from most basic to most advanced. If content exists in multiple formats, we'll let you know, so that you can choose the training format that best meets your needs.

For training about features shared with other Finance and Operations apps, see the [Dynamics 365 Finance Learning Catalog](#).

## Get started

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Learn the business value of Microsoft Dynamics 365 and the Power Platform</a>	This learning experience will take you on a journey that will begin by showing you how digital transformation and the power of business applications can transform your business. We will cover how Dynamics 365 and Power Platform help you make informed decisions based on your organization's needs and goals, through four building blocks of digital transformation—modern, unified, intelligent, and adaptable.	Free, self-paced online learning path	6.5 hours
<a href="#">Get started with Dynamics 365 Commerce</a>	Dynamics 365 Commerce, built on the proven Dynamics 365 Retail capabilities, delivers a comprehensive omnichannel solution that unifies back-office, in-store, call center, and digital experiences. Dynamics 365 Commerce enables you to build brand loyalty through personalized customer engagements, increase revenue with improved employee productivity, optimize operations to reduce costs and drive supply chain efficiencies, ultimately delivering better business outcomes.	Free, self-paced online learning path	2 hours

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Get started with Dynamics 365 Fraud Protection</a>	Dynamics 365 Fraud Protection is a cloud-based solution designed for e-commerce, brick-and-mortar stores and omnichannel merchants, to help protect their revenue and reputation by providing tools and capabilities to decrease fraud and abuse, reduce operational expenses and increase acceptance rates, while safeguarding user accounts from fraud exposure.	Free, self-paced online learning path	35 minutes

## Commerce

CONTENT	DESCRIPTION	FORMAT	LENGTH
<a href="#">Deploy and extend Dynamics 365 Commerce</a>	This learning path describes how to deploy Dynamics 365 Commerce environments and perform post-provisioning steps, for successful Commerce implementations using the correct topology for each retailer's business needs.	Free, self-paced online learning path	5 hours

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

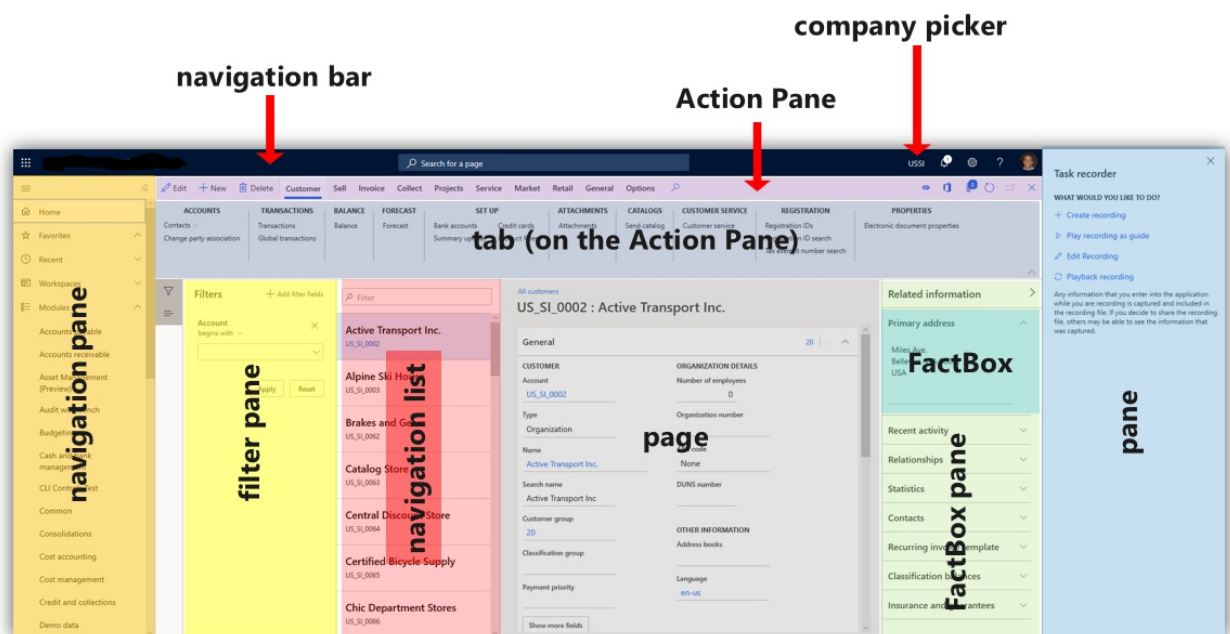
# User interface elements

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes the user interface (UI) elements used in the app. Before users can navigate the interface, it's important to know the names and functions of the elements that make up the interface.

## Overview

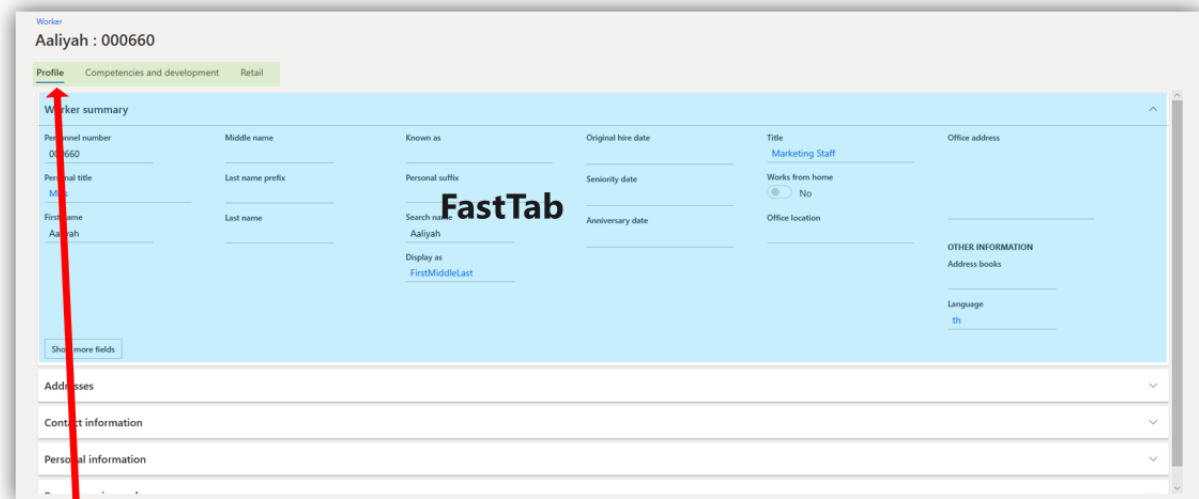
- **Action Pane** - The bar beneath the navigation bar. Here, you can select tabs to change records shown in the page. You can edit and save the records here.
- **FactBox** - You can see information and follow the activities of certain records in this pane.
- **FactBox pane** Here, you can scroll through different aspects of a record to view in the FactBox.
- **Filter pane** - On some pages, you can select **Show filters** to open this pane. It allows you to narrow the results visible to you on the page.
- **Navigation bar** - The bar at the top of the interface. It contains the **Dynamics 365 portal**, **Search**, **company picker**, **Action center**, **Settings**, **Help & Support**, and the user profile.
- **Navigation list** - On some pages, you can scroll through this pane to find a specific record. When selected, the details of the record will appear in the page.
- **Navigation pane** - The left-most pane. From here, you can find any page in the product.
- **Page** - The central focus of the interface. Selections made on the other UI components will affect what records are shown here.
- **Pane** - The right-most pane. This will open in some cases when aspects of a record need to be changed and saved.
- **Tab** - When referring to the Action Pane, it's a menu of options that appears when you select a given option in the Action Pane.



## Tabs, fields, and sections

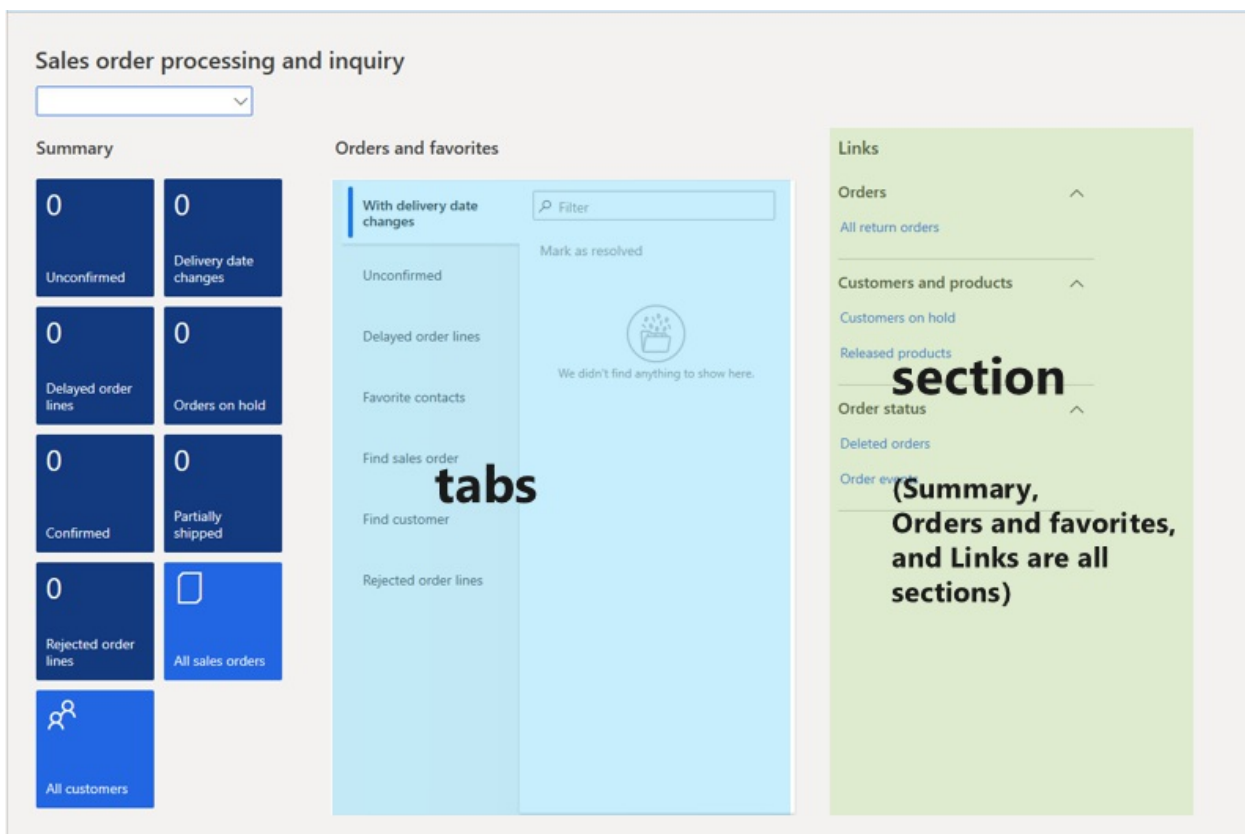
A *tab* is a selection made on the page that opens a different aspect of a record on the same page. Often, it will allow you to change certain *fields*, or UI elements that allow typed input.

A *FastTab* is a tab with the added benefit of allowing multiple tabs to be visible at the same. You can expand a FastTab by selecting the downward-pointing arrow on the right end of it.



**tabs**

A *section* is similar to a tab. The word "section" is often used to describe any area of a page that organizes a specific category of information. In the following image, Summary, Orders and favorites, and Links are all examples of sections.



## Dialog boxes and drop-down menus

A *dialog box* is a pane that opens when certain selections are made to change or create a record. Dialog boxes contain fields that allow you to enter typed input. Sometimes, a given field will allow you to select a downward facing arrow that opens a list of options to choose from. This is called a *drop-down menu*. In the following image, the **Type** and **Customer group** fields contain the option to open a drop-down menu.

In some cases, a dialog box will open near a given button when you select it. This is called a *drop-down dialog box*. In the following image, the **As of date** button was selected, which opened a drop-down dialog box.

## Notifications

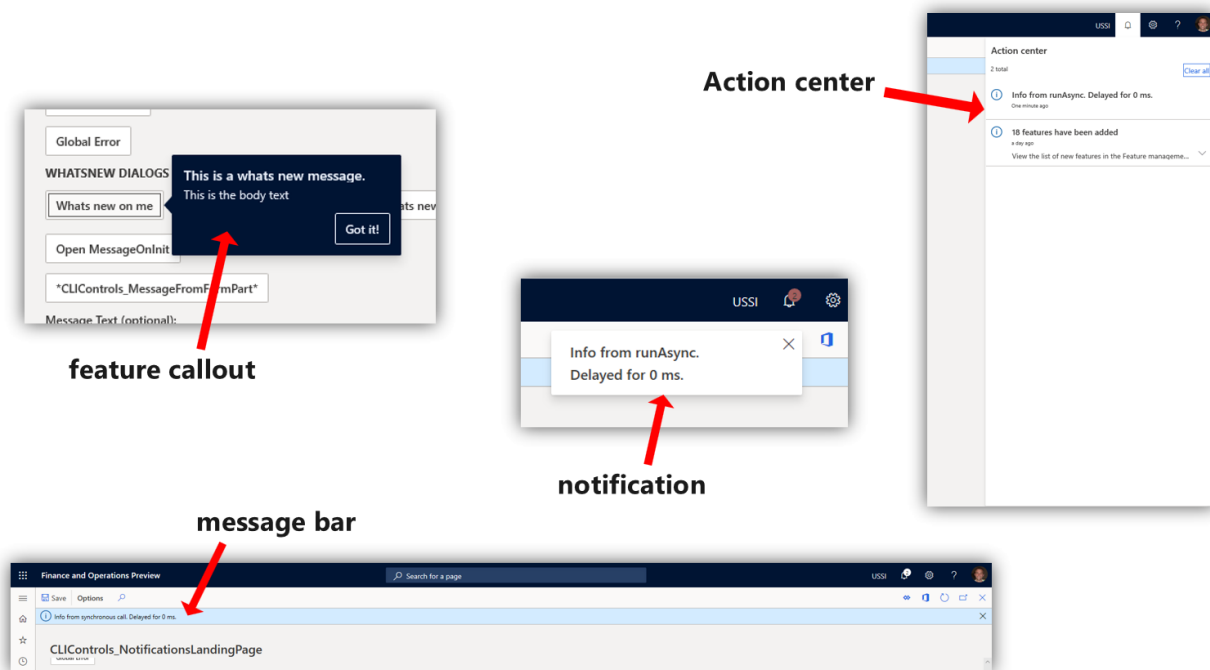
Certain changes to the objects you oversee will appear as *notifications*. Notifications may notify you when a specific customer's information has been changed, or it may alert you when the system can't accept inputs you've added in certain fields. You can learn how to customize what you receive notifications about in the [Alerts overview](#).

Notifications appear in a variety of ways.

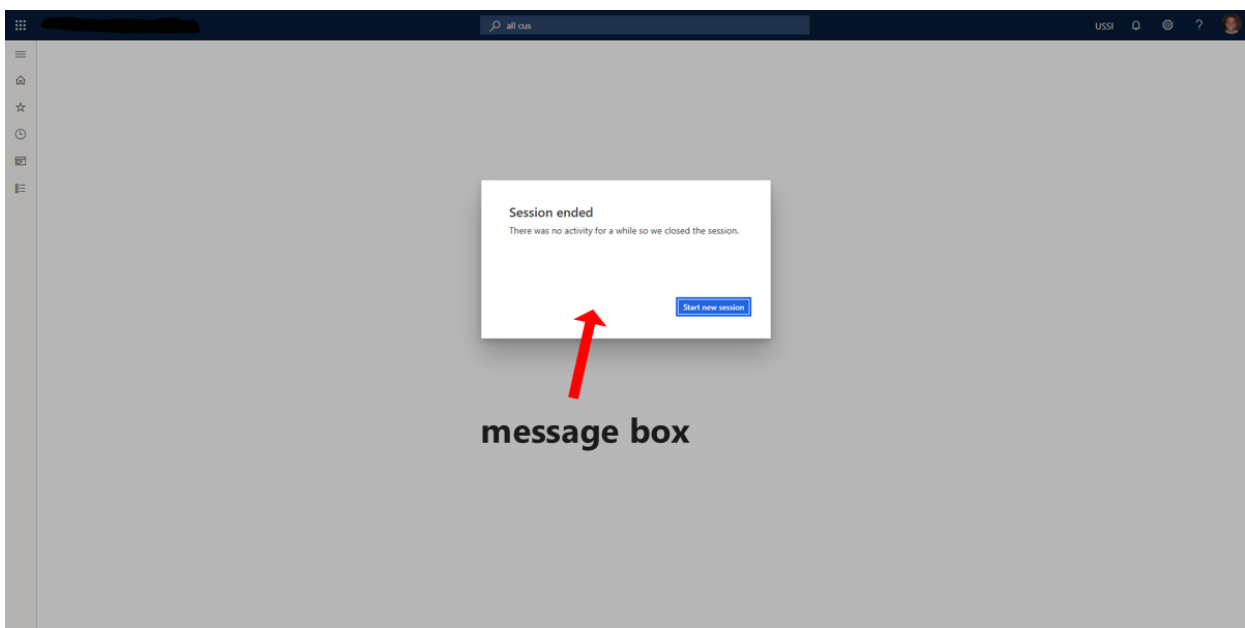
- **Feature callout** - This will appear next to a field, tab, or other button to offer an explanation of what the feature is used for.
- **Action center** - A box that contains the notification will appear next to the Action center button on the navigation bar. You can see details about the notification by selecting **Action center**.
- **Message bar** - This will appear beneath the Action Pane.



The following image shows examples of these types of notifications.



- **Message box** - This will appear over the interface and must be interacted with before you can continue to use the product.



## Toolbars, grids, and lists

A *toolbar* contains tools, such as the ability to add fields or remove records. Sometimes, a toolbar will appear on the page above a *grid*. This area, *grid*, is a name given to rows of records with various columns of data. Not all grids have toolbars above them.

A *list* is the name given to a collection of records that you can scroll through. You can bring these records into the page by selecting them. Often, this will open a grid.

toolbar

The screenshot displays the Dynamics 365 Finance and Operations Preview interface. At the top, there is a navigation bar with various tabs like 'NEW', 'MAINTAIN', 'PAYMENTS', 'COPY', 'VIEW', 'FUNCTIONS', 'ATTACHMENTS', and 'EMAIL NOTIFICATION'. Below this, a search bar and a filter are visible. The main content area shows a sales order for '000768 : Contoso Retail San Diego'. On the left, a list of sales orders is shown, with the selected order highlighted. The main grid displays the sales order lines, including columns for 'Item number', 'Product name', 'Sales category', 'Quantity', 'Unit', and 'Delivery type'. A red arrow points to the 'Warehouse' dropdown menu in the toolbar above the grid.

Item number	Product name	Sales category	Quantity	Unit	Delivery type
T0001	SpeakerCable / Speaker cable 10	Accessories	-58.00	ea	Stock
T0004	TelevisionMT2037 / Television ...	Television	-58.00	ea	Stock
T0002	ProjectorTelevision	Television	-35.00	ea	Stock
T0005	TelevisionHDTVX59052 / Televis...	Television	-23.00	ea	Stock
T0003	SurroundSoundReceive	Receivers	-35.00	ea	Stock

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Accessibility features

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes the functionality that is designed to help users who have various disabilities use this app. For example, there are features for people who use sight-assistive technologies such as Microsoft Windows Narrator.

## Windows Narrator and keyboard-only access

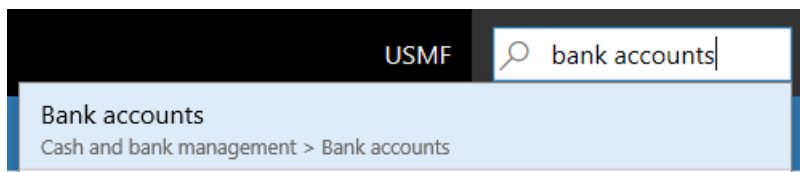
Every field and control has a label and a description of applicable shortcuts. A screen reader can read the label and description.

## Shortcuts for the most frequently performed actions

For most users, everyday system use involves lots of data entry and keyboard interaction. To enhance the user experience, we have created shortcuts to help you "jump" around the screen and shortcuts for specialized actions. For more information, see [Keyboard shortcuts](#).

## Navigation search

Any page that is accessed by using the Navigation pane menu, the left-most pane, is also available from the Search box. Press Alt+G to move focus to the Search box, and then type the name or description of the page.



For more information, see [Navigation search](#).

### NOTE

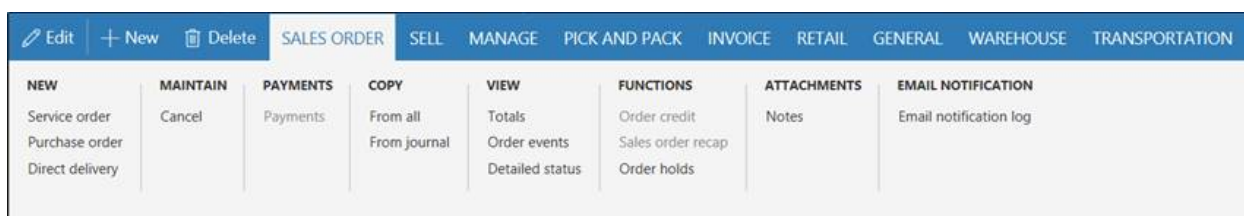
You can navigate directly to top-level pages only. Secondary pages rely on information or context from their parent page.

## Action search for keyboard-only users or for heads-down data entry

Every action that is provided on a page can be accessed from a keyboard, via the tab sequence. Information about the tab sequence is provided later in this topic. To run actions more directly, you can use the action search functionality.

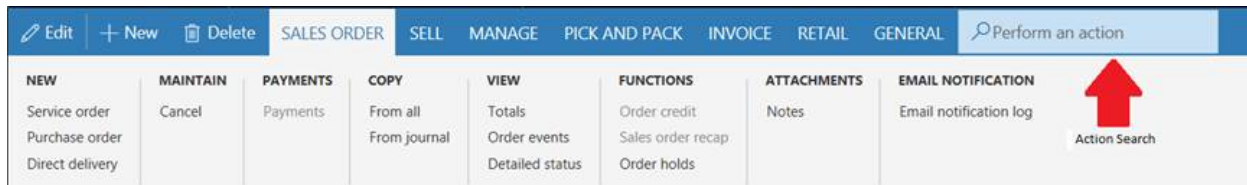
### Example

You want to run the **Email notification log** action that appears in the **Email notification** group on the **Sales order** tab on the Action Pane.

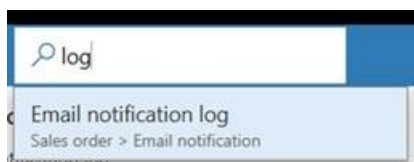
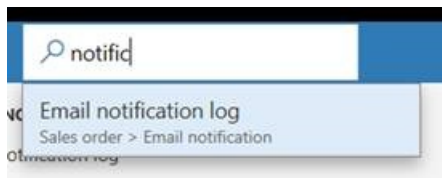
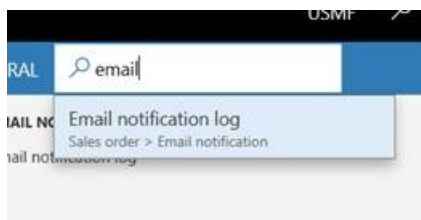


One option is to use your keyboard. Press Ctrl+F6 to move focus to the Action Pane, and then press Tab repeatedly to move through all the tabs and actions, until the **Email notification log** action has focus.

However, you can also run the action more directly. From anywhere on the page, press Ctrl+Apostrophe (') to show the search box for actions.



In the search box, type words that describe the action. The action is made available to you, and you can run it directly. For example, by typing **email**, **notific** (a partial word), or **log**, you can "jump" to the Email notification log functionality.



When you've finished, you can press Ctrl+Apostrophe again to return focus to the field that you were working with before you ran the action search.

For more information, see [Action search](#).

## Tab sequence

In everyday system use, not every field is required in order to perform typical tasks. Therefore, by default, the tab sequence is "optimized." Tab stops are set only on those fields that are essential for typical scenarios.

However, you might find that some fields that you often use to perform tasks aren't included in the default tab sequence. In this case, if you use Windows Narrator, you can use Windows Narrator's keyboard actions to access those fields and inspect their content. Alternatively, you can turn on the **Enhanced tab sequence** option on the **Options** page. This option makes all editable and read-only fields part of the tab sequence. You can then use page personalization to create a custom tab sequence and omit fields that don't have to be part of the tab sequence. For more information about personalization, see [Personalize the user experience](#).

### Accessibility

Enhanced tab sequence  
Yes

## Form patterns

Almost 90 percent of the pages in the app are based on a small set of patterns. These patterns are referred to as *form patterns*. Each form pattern is used to provide the actions that are most often performed on the page. A form pattern helps guarantee familiarity and ease of understanding, because frequently used actions and data are always presented in the same location on different pages. Because of the small number of form patterns, users can easily learn the system, regardless of the number of pages in it, and can confidently use it after they recognize the form patterns.

To learn more about form patterns, see [Form styles and patterns](#).

## Responsive layout

The product is designed to work on various devices and form factors, from the smallest screens to large screens that have the highest resolution. Our responsive layout engine lets users zoom in to a magnification level of 200 percent (or, in some scenarios, more than 200 percent).

On smartphones and other small screens, the controls and the form layout will responsively adapt to ensure that the core data is favored. These responsive behaviors can also include reducing the number of columns in groups and tabs to a single column, hiding shell elements, and collapsing the Action Pane.

## Guidance to help developers and customers incorporate accessible thinking in their customizations

To learn more about Microsoft best practices for enabling accessibility, see [Accessibility in forms, products, and controls](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Feature management overview

2/18/2021 • 13 minutes to read • [Edit Online](#)

Features are added and updated in every release. The Feature management experience provides a workspace where you can view a list of features that have been delivered in each release. By default, new features are turned off. You can use the workspace to turn them on and view the documentation for them.

## The Feature management workspace

You can open the **Feature management** workspace by selecting the appropriate tile on the dashboard. You will see a page that shows a list of features for all releases that are supported by the Feature management experience. Over time, Microsoft will enhance the Feature management experience so that it includes more functionality to help you manage features.

The feature list includes the following information:

- **Feature name** – A description of the feature that was added.
- **Enabled status** – A symbol indicates whether a feature has been turned on (check mark), hasn't been turned on (blank), is scheduled to be turned on (clock), is mandatorily turned on (lock), requires attention before you turn it on (warning), or can't be enabled (X). The setting that is shown is used for all legal entities. Note that even when a feature has been turned on, it's still controlled by security. Therefore, the feature will be available only to users who have access to it, based on their security role. It will also be available only in legal entities that the user has access to.
- **Enable date** – The date when the feature was turned on or is scheduled to be turned on.
- **Feature added** – The date when the feature was added to your environment. This date is automatically entered when you update your environment during the monthly release cycles.
- **Module** – The module that is affected by the new feature.

When you select a feature, more information appears in the details pane to the right of the feature list. At the top of the pane, you will see the feature name, the date when the feature was added, the module that is affected by the feature, and a **Learn more** link. Select this link to view the documentation for the feature. If documentation isn't available, you're taken to a temporary page. The details pane also includes a **Comments** field where you can add your own comments about the feature.

The **Feature management** workspace also has several tabs, each of which shows a list of features.

- **New** – This tab shows all features that have been added since the last monthly update. If you've skipped any monthly updates, the tab shows all the new features that have been added since the last time that you updated. The newest features appear at the top of the list. The total number of new features is also shown on a tile at the top of the page.
- **Not enabled** – This tab shows all features that haven't been turned on. The newest features appear at the top of the list. The total number of new features that haven't been turned on is also shown on a tile at the top of the page.
- **Scheduled** – This tab shows all features that have been scheduled to be turned on in the future. The features that have the earliest scheduled date appear at the top of the list. The total number of schedule new features is also shown on a tile at the top of the page.
- **All** – This tab shows all features. The newest features appear at the top of the list.

## Turn on a feature

If a feature hasn't been turned on, an **Enable Now** button appears in the details pane. You can use this button to turn on the feature.

- Select the feature to turn on, and then, in the details pane, select **Enable Now**. The feature is turned on.

Some features can't be turned off after you turn them on. If the feature that you're trying to turn on can't be turned off, you receive a warning. At that point, you can select **Cancel** to cancel the operation and leave the feature turned off. However, if you select **Enable** to turn on the feature, you won't be able to turn it off later.

Some features will display a message that provides additional information before you turn them on. These features are indicated by a yellow warning symbol. You should read the additional information carefully to better understand what will happen when the feature is enabled. However, you can still select **Enable** to turn on the feature.

Some features will display a message that the feature can't be enabled until an action is taken. These features are indicated by a red X symbol. You must take the actions described in the description before the feature is enabled. For example, if you can't use a feature until a configuration key is disabled, then you must disable the configuration key first and then return to Feature management to enable the feature.

After a feature is turned on, a message appears below the **Learn more** link in the details pane. This message either states that the feature was turned on or it indicates that the future date when the feature is scheduled to be turned on. It appears every time that you select the feature in the feature list.

Features that are scheduled to be turned on in the future appear on the **Scheduled** tab. A batch process will turn them on at midnight on the specified date, based on the time zone that is represented by the system date.

## Reschedule a feature

If a feature has been scheduled to be turned on in the future, a **Schedule** button appears in the details pane. You can use this button to change the **Enable date** value to a different date.

1. Select the scheduled feature to reschedule, and then, in the details pane, select **Schedule**.
2. In the dialog box that appears, in the **Enable date** field, specify the new date when the feature should be turned on.
3. Select **Enable** to reschedule the feature or **Disable** to cancel the schedule.

## Turn off a feature

If a feature has already been turned on, a **Disable** button appears in the details pane. You can use this button to turn off the feature. The **Disable** button isn't available if the feature can't be turned off after it's turned on.

- Select the feature to turn off, and then, in the details pane, select **Disable**. The feature is turned off, and the **Enable date** field is cleared.

After a feature is turned off, a message appears below the **Learn more** link in the details pane. This message states that the feature hasn't yet been turned on. It appears every time that you select the feature in the feature list. Features that haven't been turned on appear on the **Not enabled** tab.

## Features that must be turned on

Sometimes, a critical feature is delivered that must be turned on automatically when you do an update. These features will be turned on automatically on the date that is specified in the **Enable date** field. For these features, a message appears below the **Learn more** link in the details pane. This message either states that the feature was turned on or indicates the future date when the feature will be turned on. It appears every time that you select the feature in the feature list.

## Enable all features

By default, all features that are added to your environment are turned off. You can enable all features by selecting the **Enable all** button.

When you select **Enable all**, an option will appear where you need provide the following information:

- A list of all features that require confirmation before they can be enabled. If you want to enable the features in the list, select **Yes** for the **Enable features requiring confirmation** button.
- A list of all features that can't be enabled will be shown. Those features will not be enabled.

All features that can be enabled will be enabled. If a feature is already scheduled to be enabled in the future, the schedule will not change.

## Turn on all features automatically

By default, all features that are added to your environment are turned off, unless they are mandatory features. However, if you want to automatically turn on all new features, you can use the drop-down list under the workspace title to change what occurs when new features are added.

- Select `Enable new features automatically` to automatically turn on all new features when they are added to your environment.
- Select `Do not enable new features automatically` to default all new features to off when they are added to your environment.

When you enable all feature automatically, it will enable all of the features that would be enabled when you click the **Enable all** button. It will not enable the features that require confirmation or the features that can't be enabled until an action is taken.

## Check for updates

Features are added to your environment after each update. However, you can manually check for updates by clicking on the **Check for updates** button. Any feature that was added to the system after the update will be added to the list of features. For example, if a flighted feature is enabled after a release, then you can check for updates and the feature will be added to your list.

## Assigning roles

The **Feature management** workspace can be opened by system admins, and also by users who are assigned to the Feature manager role or the Feature viewer role. These two roles were created to support the Feature management experience. Users in the Feature manager role can turn any feature on or off. They can also update the **Comments** field for the feature. Users in the Feature viewer role can only view the **Feature management** workspace. They can't turn features on or off.

The Feature manager role and Feature viewer role don't override the existing security that a user has. They just control whether the user can turn features on and off. They don't provide access to the features themselves.

## Features that use configuration keys

If a feature uses a configuration key, but the configuration key isn't turned on, the **Feature management** workspace doesn't show the feature in the list of available features. After you turn on the configuration key, you must update the feature list by using the **Check for update** menu item. The feature then appears in the feature list.

If you turn off the configuration key, the feature isn't removed from the feature list.



## Data entities

A data entity that is named **Feature management** lets you export the Feature management settings from one environment and then import them into another environment. This entity updates only existing features. The business logic in the entity also helps guarantee that the same rules that are used on the **Feature management** workspace will be applied when the import is done. For example, you can't override a mandatory feature setting by removing the date during import.

The following examples describe what occurs when you use the **Feature management** entity to import data.

- If you change the value of the **Enabled** field to **Yes**, the feature is turned on, and the **Enable date** field is set to the current date.
- If you change the value of the **Enabled** field to **No** or leave the **EnableDate** field blank, the feature is turned off, and the **Enable date** field is cleared. You can't turn off a mandatory feature or a feature that can't be turned off after it's turned on.
- If you change the value of the **EnableDate** field to a future date, the feature is scheduled for that date.
- If you change the value of the **Enabled** field to **Yes** and change the value of the **EnableDate** field to a future date, the feature is scheduled for that date.
- If you change the value of the **Enabled** field to **No**, but you also change the value of the **EnableDate** field to a future date, the feature is scheduled for that date.
- If a feature is turned on, and you add an **EnableDate** field that is set to a future date, the feature remains turned on. To reschedule the feature, you must change the **Enabled** field to **No**.

## Feature management and flighting

Feature management lets you control the features that are delivered in each release. Flighting lets Microsoft teams release features to a limited number of customers, so that those features can be tested and validated without affecting all customers. Feature management doesn't control the flighting of any features.

## New features are optional for 12 months

When a new non-critical feature is installed, it will be optional for a 12-month period. This allows you and your organization time to plan ahead for when to uptake a feature and have it tested against your daily operations. For more information, see [One Version service updates FAQ](#).

## Using Feature management to turn on ISV features or custom features

Feature management is currently unavailable for features from independent software vendors (ISVs) and custom features. However, Microsoft is adding more functionality to enhance Feature management. After those enhancements are completed, Microsoft will make Feature management available to all features and provide instructions for updating your features to use it.

## Frequently asked questions (FAQ)

### **When are features added, removed, or changed?**

Features are added, removed, and changed through code changes. Environments need to be updated to receive those changes.

### **Does a feature become mandatory automatically?**

No, a feature becoming mandatory is not an automatic action. The product teams need to make a code change.

### **When do features become mandatory?**

The policy is that all new features will be opt-in for a 12-month period and will not require any change management until you enable the feature. The product teams can choose whether to make a feature mandatory after that period has ended.

### **Why isn't there a specific 'mandatory-enabled date'?**

Update release timing is variable, environment update timing is variable, and customers can opt to skip some updates. As a result, specific dates are difficult to determine.

### **Where's the documentation for features that are being made mandatory?**

This documentation comes from the application teams. Often, these will be mentioned in [Removed or deprecated features](#).

### **Is there an in-product notification or signal that a feature is going to be mandatory-enabled?**

A notification mechanism related to making a feature mandatory does not exist today.

### **Do features ever get enabled without the customer knowing about it?**

Yes, if features don't have a functional impact then they can be enabled by default.

### **What is feature flighting and how does it relate to feature management?**

Feature flights are real-time on/off switches that Microsoft controls. They are separate from the customer control provided by Feature Management.

- Private Preview features will not be listed in Feature Management until they are flighted on. In production, the customer needs to agree to be part of a special program for that to occur.
- Public Preview and Released (generally available) features will be listed in Feature Management unless they are flighted off. Flighting a feature off is considered a last resort option for product teams if a critical issue is found and would usually be a per-customer operation.

### **Do features ever get flighted off without the customer knowing about it?**

Yes, if a feature is impacting the functioning of an environment that doesn't have a functional impact then they can be enabled by default.

### **How can feature enablement be checked in code?**

Use the `isFeatureEnabled` method on the `FeatureStateProvider` class, passing it an instance of the feature class. Example:

```
if (FeatureStateProvider::isFeatureEnabled(BatchContentionPreventionFeature::instance()))
```

### **How can feature enablement be checked in metadata?**

The `FeatureClass` property can be used to indicate that some metadata is associated with a feature. The class name used for the feature should be used, such as `BatchContentionPreventionFeature`. This metadata is visible only in that feature. The `FeatureClass` property is available on menus, menu items, enum values, and table/view fields.

### **What is a feature class?**

Features in Feature Management are defined as *feature classes*. A feature class **implements** `IFeatureMetadata` and uses the feature class attribute to identify itself to the Feature Management workspace. There are numerous examples of feature classes available that can be checked for enablement in code using the `FeatureStateProvider` API and in metadata using the `FeatureClass` property. Example:

```
[ExportAttribute(identifierStr(Microsoft.Dynamics.ApplicationPlatform.FeatureExposure.IFeatureMetadata))]
internal final class BankCurrencyRevalGlobalEnableFeature implements IFeatureMetadata
```

## What is the IFeatureLifecycle implemented by some feature classes?

IFeatureLifecycle is a Microsoft-internal mechanism for indicating the feature lifecycle stage. Features can be:

- `PrivatePreview` - Needs a flight to be visible.
- `PublicPreview` - Shown by default but with a warning that the feature is in preview.
- `Released` - Fully released.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Client FAQ

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides answers to frequently asked questions about the Finance and Operations client.

## Why aren't symbols loaded?

The security settings on your browser might prevent the symbols from being loaded correctly. To resolve this issue, try the following steps:

- If you're experiencing this issue in Internet Explorer, click **Tools**, and then click **Internet Options**. In the Internet Options dialog box, on the **Privacy** tab, click **Custom level**, and make sure the **Font download** option is selected.
- Otherwise, you might have to add the app site to the list of trusted sites.

## I miss the ribbon from Dynamics AX 2012. Can I keep Action Pane tabs open all the time?

We are planning to implement this feature soon. Users will then be able to choose to keep the tabs on Action Panes open all the time. Otherwise, the tabs will be collapsed when they aren't being used, to gain more screen space for the page.

## Why do I sometimes see different shortcut menus when I right click?

If you right-click in an editable field (or if text is selected), the browser's shortcut menu is displayed. This menu gives you access to the **Cut**, **Copy**, and **Paste** commands. We can't embed these commands into the shortcut menus because, for security reasons, browsers don't allow us to programmatically access the system clipboard.

If you right-click a field label or the value of a read-only control, you'll see the shortcut menu.

To make keyboard access easier, we plan to implement a keyboard shortcut in the future that will open the shortcut menu.

## Where is the View details functionality?

The **View details** option is available in a couple of ways:

- If a control has **View details** capabilities, and if the control has a value, that value is displayed as a hyperlink. You can click the hyperlink to open a page that contains additional details.
- **View details** is also an option on shortcut menus. For more information about when shortcut menus are displayed when you right-click, see the previous section.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

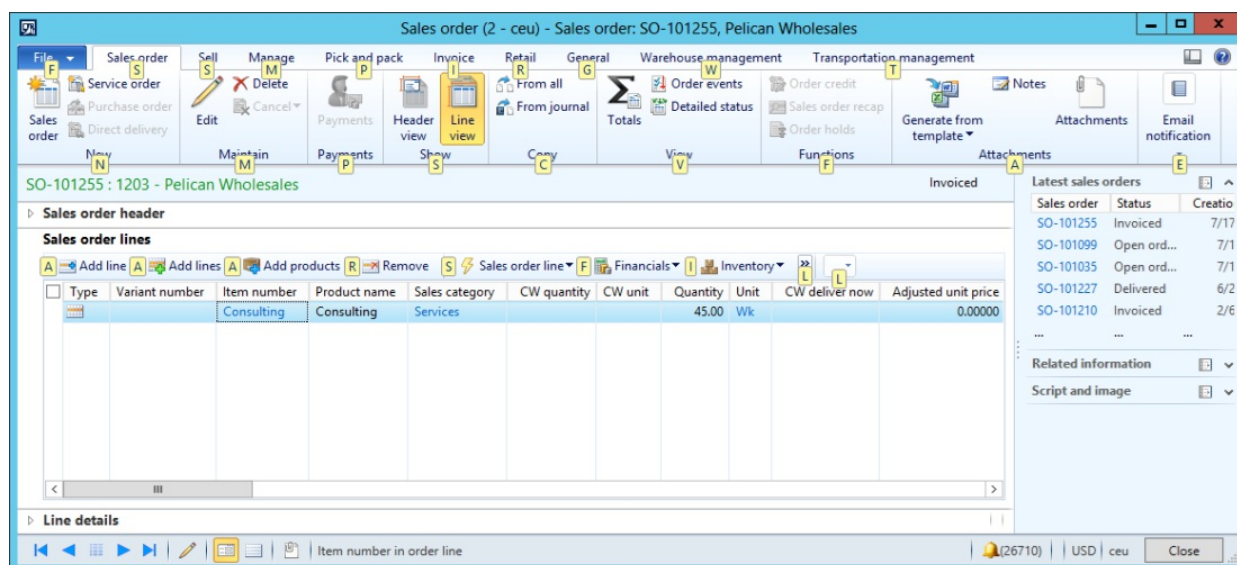
# Action search

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article describes the action search functionality. Action search will help you find and run actions on a page.

## Introduction

Pages primarily expose commands on Action Panes, both the standard Action Pane that appears at the top of a page and the toolbars that appear in various sections of the page. In previous versions, a Key Tips feature let you quickly access any button on an Action Pane by pressing the Alt key and then a series of letters.



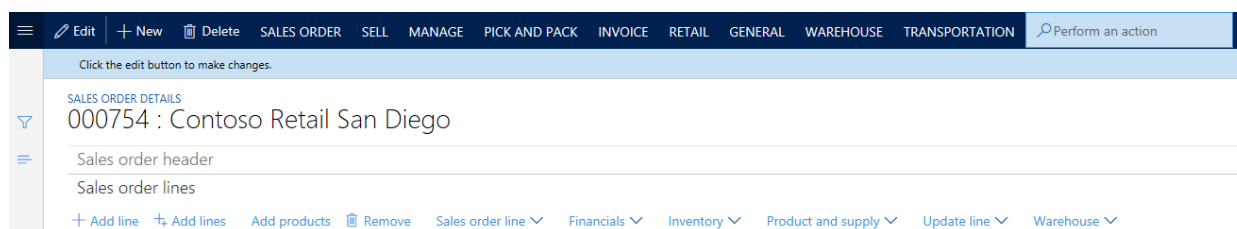
The action search feature replaces Key Tips, which are no longer available. This new feature lets you quickly search for and run a button from any visible Action Pane.

## Using action search

To use the action search feature, follow these steps.

1. On the Action Pane, click in the **action search** field. (The **action search** field contains a magnifying glass icon.)
2. Type all or part of the name of the button that you want to run. You can also search by using words from the button's "path." (For more information, see the next section of this article.) Typically, a button will appear near the top of the results list after you've typed two to four characters.
3. Find and run the button in the results list (by using your mouse or keyboard).

After the button is run, the focus is returned to your last position on the page, so that you can continue to work.



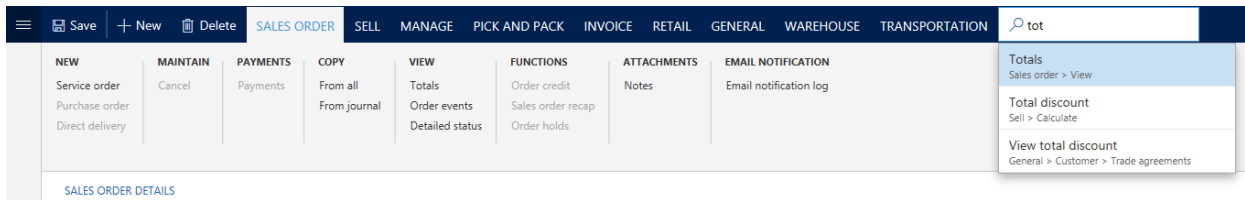
You can also start action search by pressing Ctrl+/ or Alt+Q. Press the keyboard shortcut again to return the focus to your last position on the page.

# Understanding the results list

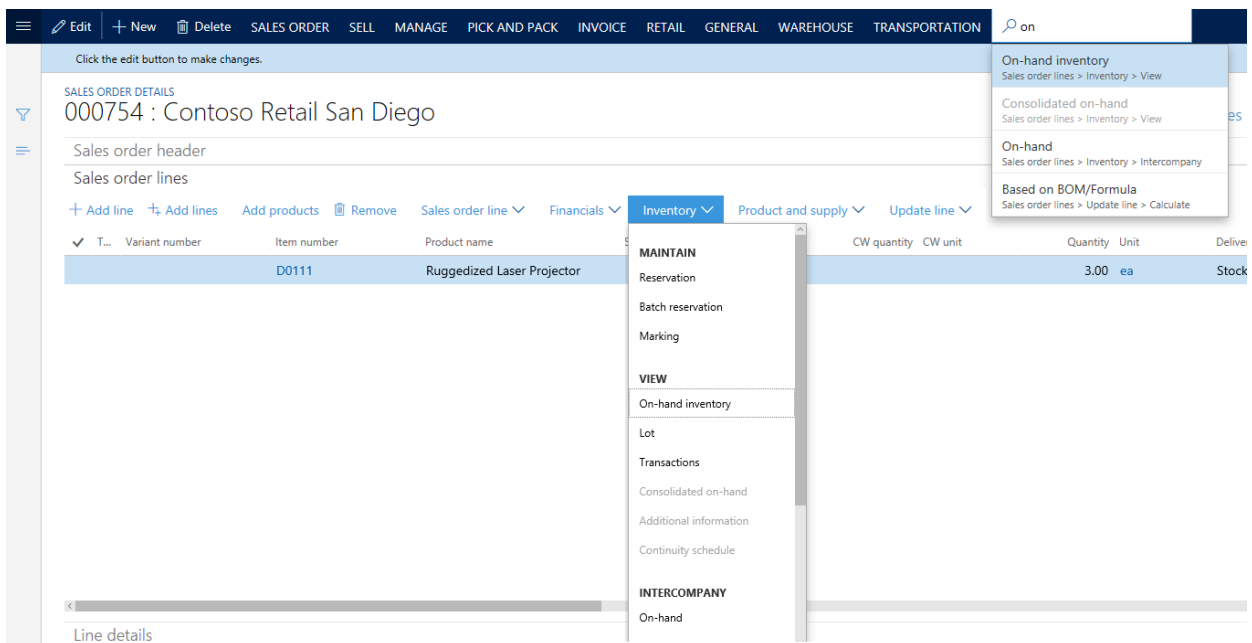
Often, you must know both the location and the context of a button to fully understand the purpose of that button. Therefore, the results list shows additional information to help you understand exactly which buttons appear in the list. In particular, the "path" of the button is shown. This path might include the labels of the following UI elements, as relevant:

- Action Pane tab
- Button group
- Menu button (if the button is inside a menu button)
- Menu separator (if the button is inside a named group inside a menu button)
- Group or tab on the page (for example, the name of a FastTab)

For example, you typed **tot** in the **action search** field and are now examining the results list. The first entry, for a button that is named **Totals**, is highlighted. A button path of **Sales order > View** is also shown. The **Sales order** part of the path corresponds to the **Sales order** tab on the Action Pane, and the **View** part of the path corresponds to the **View** group on that tab. Similarly, the path of the **Total discount** button (**Sell > Calculate**) informs you that this button is located in the **Calculate** group on the **Sell** tab of the Action Pane. Therefore, this information helps you understand exactly which button will be triggered by action search (if you select that button in the results list).



In the previous example, action search showed results from the standard Action Pane at the top of a page. However, action search also shows results from visible toolbars that are in other places on the page. For example, you're searching for the **On-hand inventory** button that is on the **Sales order lines** FastTab. In this case, the button path in the results list (**Sales order lines > Inventory > View**) informs you that this button is under the **View** heading on the **Inventory** menu button on the **Sales order lines** FastTab.



**NOTE**

There are some buttons that do not show up in Action search. These include drop dialog buttons and buttons from subforms.

## Action search vs. Navigation search

Whereas action search is intended to find and run actions on a page, there is a separate search mechanism for finding and navigating to pages. For more information about that feature, see the [Navigation search](#) article.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Advanced filtering and query syntax

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes the filtering and query options that are available when you use the Advanced filter/sort dialog or the **matches** operator in the Filter pane or grid column header filters.

## Advanced query syntax

SYNTAX	CHARACTER DESCRIPTION	DESCRIPTION	EXAMPLE
<i>value</i>	Equal to the value that is entered	Type the value to find.	<b>Smith</b> finds "Smith".
! <i>value</i> (exclamation point)	Not equal to the value that is entered	Type an exclamation point and then the value to exclude.	<b>!Smith</b> finds all values except "Smith".
<i>from-value..to-value</i> (double period)	Between the two values that are separated by double periods	Type the from-value, then two periods, and then the to-value.	<b>1..10</b> finds all values from 1 through 10. However, in a string field, <b>A..C</b> finds all values that start with "A" and "B", and values that are exactly equal to "C". For example, this query won't find "Ca". To find all values from "A" <i>through</i> "C", type <b>A..D</b> .
<i>..value</i> (double period)	Less than or equal to the value that is entered	Type two periods and then the value.	<b>..1000</b> finds any number that is less than or equal to 1000, such as "100", "999.95", and "1,000".
<i>value..</i> (double period)	Greater than or equal to the value that is entered	Type the value and then two periods.	<b>1000..</b> finds any number that is greater than or equal to 1000, such as "1,000", "1,000.01", and "1,000,000".
<b>&gt;</b> <i>value</i> (greater than sign)	Greater than the value that is entered	Type a greater than sign (>) and then the value.	<b>&gt; 1000</b> finds any number that is greater than 1000, such as "1000.01", "20,000", and "1,000,000".
<b>&lt;</b> <i>value</i> (less than sign)	Less than the value that is entered	Type a less than sign (<) and then the value.	<b>&lt; 1000</b> finds any number that is less than 1000, such as "999.99", "1", and "-200".
<i>value*</i> (asterisk)	Starting from the value that is entered	Type the starting value and then an asterisk (*).	<b>S*</b> finds any string that starts with "S", such as "Stockholm", "Sydney", and "San Francisco".



SYNTAX	CHARACTER DESCRIPTION	DESCRIPTION	EXAMPLE
* <i>value</i> (asterisk)	Ending with the value that is entered	Type an asterisk and then the ending value.	<b>*east</b> finds any string that ends with "east", such as "Northeast" and "Southeast".
* <i>value</i> * (asterisk)	Containing the value that is entered	Type an asterisk, then a value, and then another asterisk.	<b>*th*</b> finds any string that contains "th", such as "Northeast" and "Southeast".
? (question mark)	Having one or more unknown characters	Type a question mark at the position of the unknown character in the value.	<b>Sm?th</b> finds "Smith" and "Smyth".
<i>value,value</i> (comma)	Matching the values that are separated by commas	Type all your criteria, and separate them by using commas.	<b>A, D, F, G</b> finds exactly "A", "D", "F", and "G". <b>10, 20, 30, 100</b> finds exactly "10, 20, 30, 100".
"" (two double quotes)	Matching a blank value	Type two consecutive double quotes to filter for blank values in that field.	Two consecutive double quotes ("" ) finds rows with no value for the current column.
(Finance and Operations query) (Finance and Operations query between parentheses)	Matching a defined query	Type a query as an SQL statement between parentheses using the Finance and Operations query language.	<b>((AccountNum LIKE "US*") &amp;&amp; (DirPartyTable.Name LIKE "Cont*"))</b>  as an example of syntax for a filter condition on a field from the root datasource as well as a field from a different datasource (for the All customers page)
T	Today's date	Type T.	<b>T</b> matches today's date.

SYNTAX	CHARACTER DESCRIPTION	DESCRIPTION	EXAMPLE
(methodName(parameters)) (SysQueryRangeUtil method between parentheses)	Matching the value or range of values that are specified by the parameters of the SysQueryRangeUtil method	Type a SysQueryRangeUtil method that has parameters that specify the value or range of values.	<ol style="list-style-type: none"> <li>1. Click <b>Accounts receivable &gt; Invoices &gt; Open customer invoices</b>.</li> <li>2. Press Ctrl+Shift+F3 to open the <b>Inquiry</b> page.</li> <li>3. On the <b>Range</b> tab, click <b>Add</b>.</li> <li>4. In the <b>Table</b> field, select <b>Open customer transactions</b>.</li> <li>5. In the <b>Field</b> field, select <b>Due date</b>.</li> <li>6. In the <b>Criteria</b> field, enter (yearRange(-2,0)).</li> <li>7. Click <b>OK</b>. The list page is updated and lists the invoices that match the criterion that you entered. For this example, invoices that were due in the previous two years are listed.</li> </ol> <p>See the table in the next section for additional details about SysQueryRangeUtil date methods, and several examples.</p>

## Advanced date queries that use SysQueryRangeUtil methods

METHOD	DESCRIPTION	EXAMPLE
Day (_relativeDays=0)	Find a date relative to the session date. Positive values indicate future dates, and negative values indicate past dates.	<ul style="list-style-type: none"> <li>• <b>Tomorrow</b> – Enter (Day(1)).</li> <li>• <b>Today</b> – Enter (Day(0)).</li> <li>• <b>Yesterday</b> – Enter (Day(-1)).</li> </ul>
DayRange (_relativeDaysFrom=0, _relativeDaysTo=0)	Find a range of dates relative to the session date. Positive values indicate future dates, and negative values indicate past dates.	<ul style="list-style-type: none"> <li>• <b>Last 30 days</b> – Enter (DayRange(-30,0)).</li> <li>• <b>Previous 30 days and next 30 days</b> – Enter (DayRange(-30,30)).</li> </ul>
GreaterThanDate (_relativeDays=0) GreaterThanUtcDate (_relativeDays=0)	Find all dates after the specified relative date.	<ul style="list-style-type: none"> <li>• <b>More than 30 days from now</b> – Enter (GreaterThanDate(30)).</li> </ul>

METHOD	DESCRIPTION	EXAMPLE
GreaterThanOrEqualTo ()	Find all date/time entries after the current time.	<ul style="list-style-type: none"> <li>• <b>All future date/times</b> – Enter (GreaterThanOrEqualTo()).</li> </ul>
LessThanDate (_relativeDays=0) LessThanOrEqualToDate (_relativeDays=0)	Find all dates before the specified relative date.	<ul style="list-style-type: none"> <li>• <b>Less than seven days from now</b> – Enter (LessThanDate(7)).</li> </ul>
LessThanOrEqualTo ()	Find all date/time entries before the current time.	<ul style="list-style-type: none"> <li>• <b>All past date/times</b> – Enter (LessThanOrEqualTo()).</li> </ul>
MonthRange (_relativeFrom=0, _relativeTo=0)	Find a range of dates, based on months relative to the current month.	<ul style="list-style-type: none"> <li>• <b>Previous two months</b> – Enter (MonthRange(-2,0)).</li> <li>• <b>Next three months</b> – Enter (MonthRange(0,3)).</li> </ul>
YearRange (_relativeFrom=0, _relativeTo=0)	Find a range of dates, based on years relative to the current year.	<ul style="list-style-type: none"> <li>• <b>Next year</b> – Enter (YearRange(0, 1)).</li> <li>• <b>Previous year</b> – Enter (YearRange(-1,0)).</li> </ul>

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

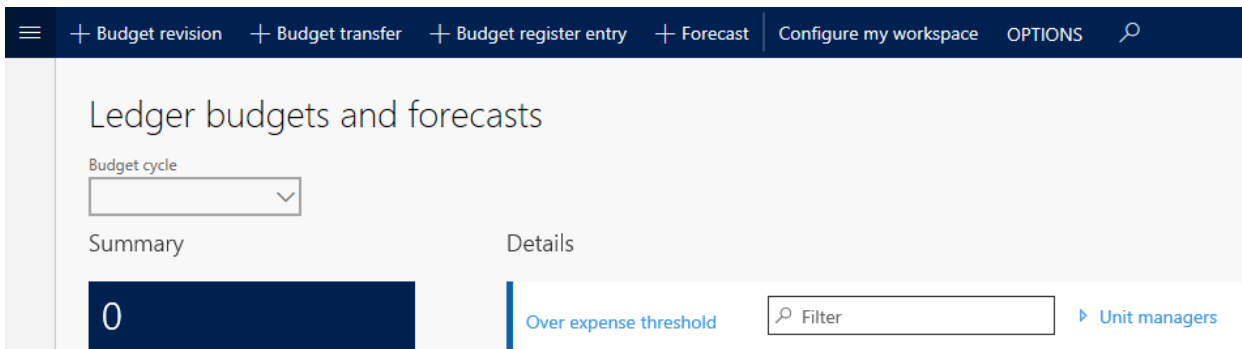
# Configure and filter workspaces

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides an overview about how to configure and filter workspaces.

## Configuring a workspace

You can change the appearance and behavior of some workspaces by updating settings that apply to the whole workspace. When a workspace can be configured, the Action Pane includes a button that instructs you to click it to make configuration changes. For example, in the following illustration, the button is named **Configure my workspace**.



When you click the button, a dialog appears, where you can modify the predefined settings for the workspace. The specific settings that you see in this dialog vary by workspace, and depend on the specific controls and business data that are available in the workspace.

Configure my workspace

[Restore default values](#) [View default values](#)

**WHICH DATA DO YOU WORK WITH?**

Show amounts

Organization hierarchy

Budget model

Active forecasting process

**WHAT INFORMATION WOULD YOU LIKE TO SEE IN THE DETAIL**

Expense dimension set

Expense budget threshold percent

Revenue dimension set

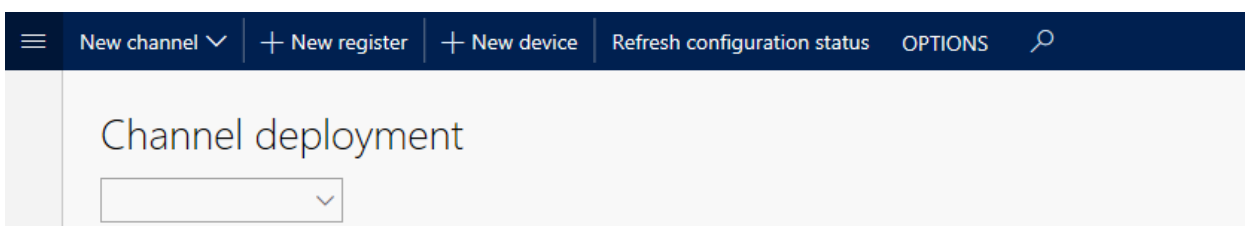
Revenue budget threshold percent

## Filtering a workspace

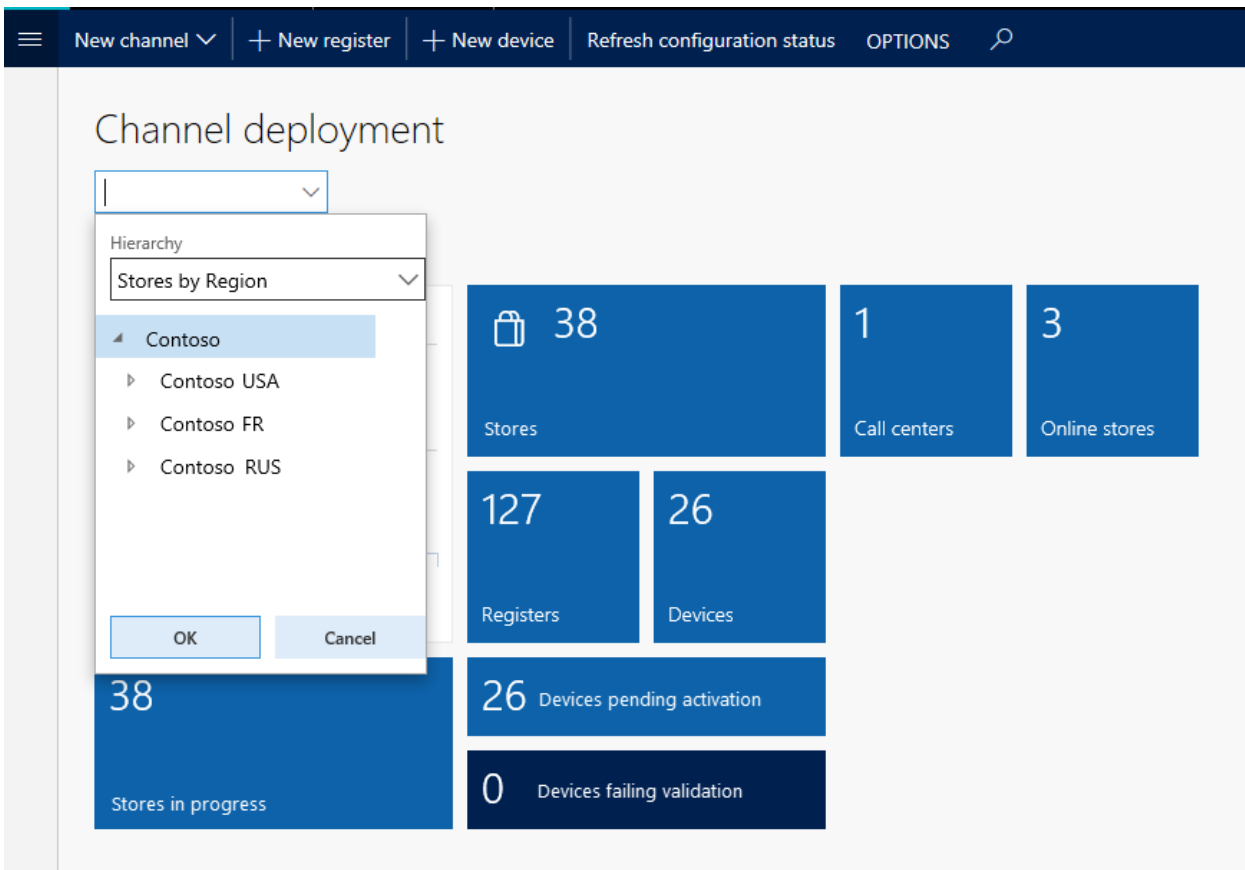
Many workspaces let you filter the content that appears in them. The controls that are available might let you filter all the content in the workspace or only the content in a specific section of the workspace. The filters on workspaces can be lookups, combo boxes, free-form text fields, or other types of controls. However, every type of filter has the same effects, as described in the following sections.

### Workspace-wide filters

You can filter the whole workspace by using a workspace-wide filter. A workspace-wide filter appears in the upper-left corner of the workspace. When you select a specific value in the drop-down list, the contents of the workspace are filtered based on that selection.



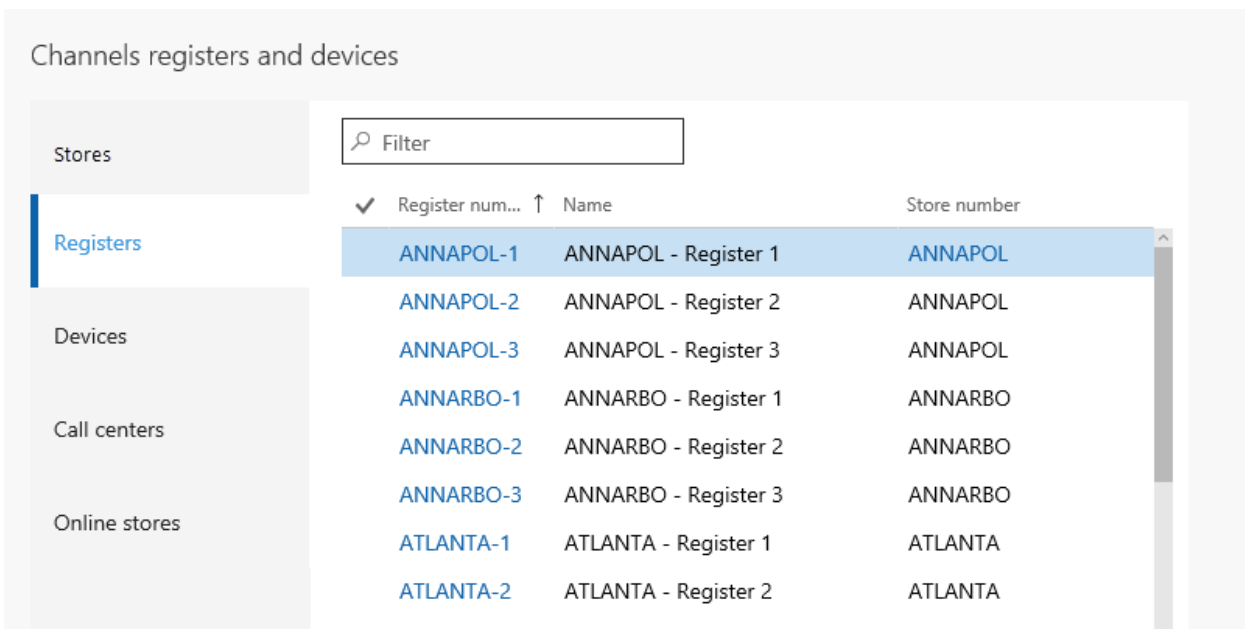
When you click to open the filter, you're presented with several options.



Select an option to filter the workspace based on that option.

### Workspace section filters

If individual sections of the workspace have filters, you can filter each section separately. In the following illustration, the filter (the field that contains the text "Filter") is an example of a free-form text field filter.



As with a workspace-wide filter, select or enter a value in the field to filter the contents of the section.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

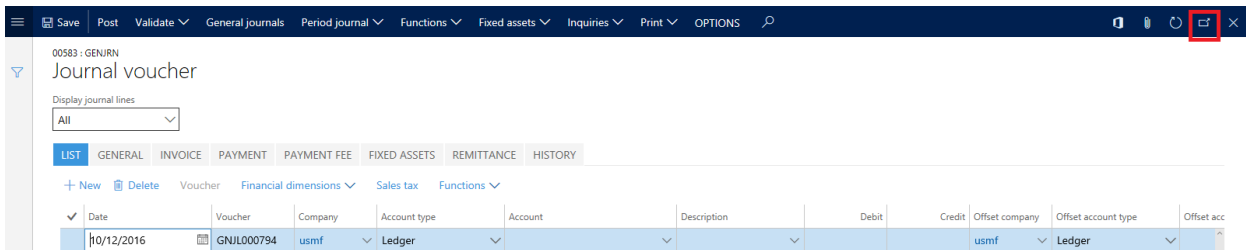
# Show pages side-by-side using the Open in new window feature

2/18/2021 • 2 minutes to read • [Edit Online](#)

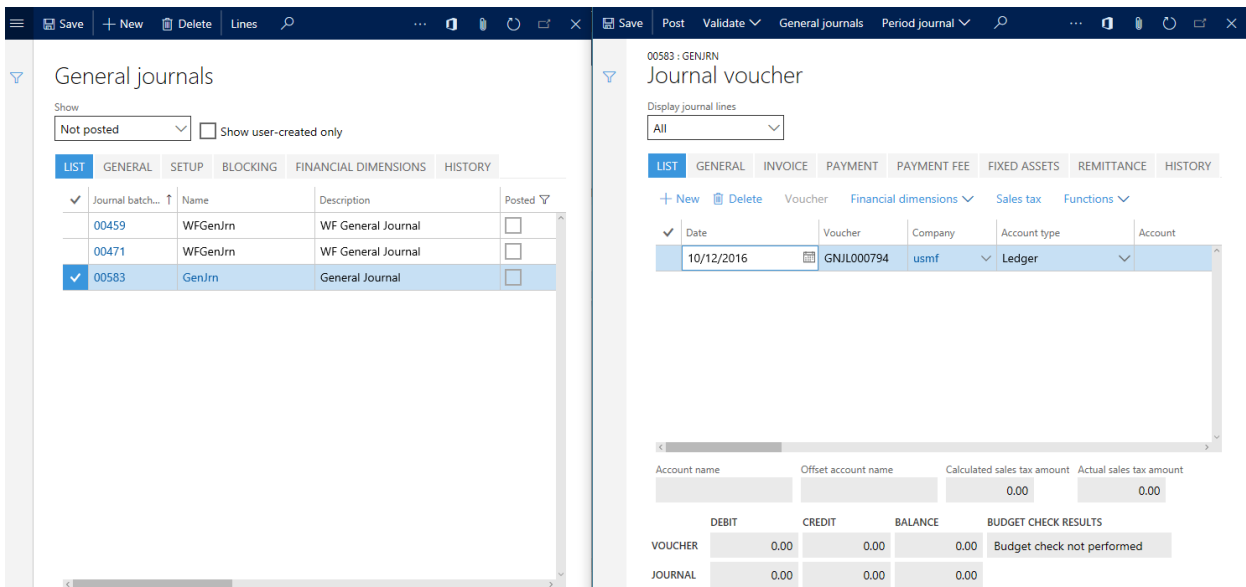
This article explains how to display pages side by side.

You may want to view multiple pages side by side to complete tasks quickly. As an example, you might want to validate or enter lines in more than one journal. Typically, to validate or enter lines in more than one journal, you would have to go back and forth between the page that displays a list of journals, and the page that displays lines for a given journal. However, the **Open in new window** feature enables you to display these pages side-by-side so that you can perform your tasks quickly.

Continuing with the example mentioned above, when viewing the lines, you can click the **Open in new window** icon.



Clicking the **Open in new window** icon opens the lines page in a new, pop-up browser, and then navigates the original browser back in history to the page that displayed the list of journals. You can then display both pages side by side. After viewing a journal, you can change the selected journal on the journal list page, and the lines page in the pop-up window will automatically display the lines of the newly selected journal.



The dynamic linking and refreshing happens due to the relations that exist between the data that is backing these pages. If the system is not aware of the relation between the data, the pop-up window will not refresh automatically in response to a change in the window it originated from.

Some pages have multiple views such as the Grid view, Header view, and Details view. The **Open in new window** icon causes the entire page to open in the new browser window. Therefore, you cannot keep two views of the same page side by side using the **Open in new window** feature. Almost all such pages have a

navigation list that you can use to switch between records and achieve a similar experience.

Before using the **Open in new window** feature, you should configure your browser's pop-up blocker to allow pop-ups from the URL of the site. As an example, you could allow pop-ups from "\*.dynamics.com".

The **Open in new window** feature is only available when there is more than one page open in the window. Also, the pop-up window automatically closes when there are no more pages open (that is, when you close the last page in that window). The system also closes open pages when you navigate to a different area in the application. Therefore, if you have pop-up windows open and navigate to a different area in the application, the pop-up windows close automatically because the system closed the pages in those windows.

The top bar in the pop-up windows displays information about the company the page was opened in and is read-only. The pop-up windows also rely on the main browser window. If the main window is closed or refreshed, all open pop-up windows will become read only. If this situation occurs, you can still view the information in these windows, but you will not be able to interact with it.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Keyboard shortcuts

2/18/2021 • 9 minutes to read • [Edit Online](#)

The following keyboard shortcuts can help you quickly and efficiently enter data in Finance and Operations apps

## NOTE

The keyboard shortcuts described here refer to the United States keyboard layout. Keys on other keyboard layouts might not correspond exactly to the keys on a US keyboard.

Some of the shortcuts on this page are *key chords*, meaning they require two consecutive sets of key combinations (separated by a comma) that need to be independently pressed to trigger the required action. For example, the shortcut "Alt+M,A" requires the user to first press "Alt+M", release the keys, and then press "A".

## Finding a shortcut

As of Platform update 11, users can discover currently available shortcuts directly from the user interface. Simply right-click on a control and select **View shortcuts**. This will open a dialog box showing the shortcuts you can use based on where you are on the page.

## Action shortcuts

TO DO THIS	PRESS
Open action search	Ctrl+' or Alt+Q
Move to the standard Action Pane	Alt+M,A or Ctrl+F6
Open a tab on the Action Pane or a menu	Enter or Space or Alt+Down arrow
Move to next/previous option in a menu	Down arrow / Up arrow
Close a tab on the Action Pane or a menu	Esc
Simulate a right-click	Shift+F10
Open the context menu	Ctrl+F10
Execute the default button on a form/dialog box	Alt+Enter
Click a button or tile	Enter or Space
View refresh information for a count tile	Alt+Up arrow
View currently available shortcuts	Alt+Shift+K

## Date picker shortcuts

TO DO THIS	PRESS
Open the date picker	Alt+Down arrow
Move between dates in the date picker	Ctrl+Arrow keys
Move to the next/previous month	Page down / Page up
Move to the next/previous year	Ctrl+Shift+Page down / Ctrl+Shift+Page up
Move to today	Ctrl+Home
Pick today's date	T
Clear the selected date	C
Pick Never (or the max date)	N

## FactBox shortcuts

TO DO THIS	PRESS
Open the FactBox pane (or move focus to the FactBox pane if it is already open)	Alt+M,B or Ctrl+F2
Close the FactBox pane (with focus in the FactBox pane)	Esc
Move to the next/previous FactBox (with focus in the FactBox pane)	Alt+Shift+Down arrow / Alt+Shift+Up arrow
Move to the <n>th FactBox (with focus in the FactBox pane)	Alt+<n> (<n> = 1-9)
Expand a FactBox (with focus on the FactBox header)	Space or Enter
Collapse the current FactBox	Alt+0

## Filtering shortcuts

TO DO THIS	PRESS
Open grid filtering for the current column	Ctrl+G
Close grid filtering for the current column	Esc
Open the Filter pane (or switch focus between the Filter pane and the main form if the Filter pane is already open)	Alt+M,F or Ctrl+F3
Close the Filter pane (with focus in the Filter pane)	Esc
Open advanced filtering/sort	Ctrl+Shift+F3

## Form shortcuts

TO DO THIS	PRESS
Create a new record	Alt+N
Delete a record	Alt+Del or Alt+F9
Save record	Alt+S or Ctrl+S
Revert (restore)	Ctrl+Shift+F5
Data refresh	Shift+F5
Move to the visible first field on the form	Alt+Shift+F
Toggle edit mode	F2
Attach a document	Ctrl+Shift+A
Export to Excel	Ctrl+Shift+E
Move to the previous record (outside a grid)	Ctrl+Up arrow
Move to the next record (outside a grid)	Ctrl+Down arrow
Move to the first record (outside a grid)	Ctrl+Home
Move to the last record (outside a grid)	Ctrl+End
Close the form (click Back)	Esc
Close the form with explicit save	Shift+Esc
Close the form discarding any unsaved changes	Alt+Shift+Q

## Form navigation shortcuts

TO DO THIS	PRESS
Move to the next/previous field	Tab / Shift+Tab
Move to the next/previous tab	Alt+Shift+Right arrow / Alt+Shift+Left arrow
Move to the <n>th tab	Alt+Shift+<n> (<n> = 1-9)
Move to the next/previous FastTab	Alt+Shift+Down arrow / Alt+Shift+Up arrow
Move to the <n>th FastTab	Alt+<n> (<n> = 1-9)
Move to the next/previous blade (vertical tab)	Alt+Shift+Right arrow / Alt+Shift+Left arrow

TO DO THIS	PRESS
Move to the <n>th blade (vertical tab)	Alt+Shift+<n> (<n> = 1-9)
Expand a FastTab (with focus on the FastTab header)	Space or Enter
Collapse the current FastTab	Alt+0
Switch to grid view	Ctrl+Shift+G
Switch to details view	Ctrl+Shift+D
Switch to header view	Ctrl+Shift+H
Switch to lines view	Ctrl+Shift+L

## Grid shortcuts

TO DO THIS	PRESS
Move to the next/previous column	Tab / Shift+Tab
Move to the next/previous row	Down arrow / Up arrow
Move to the next/previous row without selecting <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> <p>[!NOTE] This shortcut applies to multi-select scenarios only.</p> </div>	Ctrl+Up arrow / Ctrl+Down arrow
Select/clear the current row <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> <p>[!NOTE] This shortcut applies to multi-select scenarios only.</p> </div>	Ctrl+Space / Ctrl+Click
Add the next/previous row to the selected set <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> <p>[!NOTE] This shortcut applies to multi-select scenarios only.</p> </div>	Shift+Space
Add a range of rows to the selected set <div style="border: 1px solid #ccc; padding: 2px; margin-top: 5px;"> <p>[!NOTE] This shortcut applies to multi-select scenarios only.</p> </div>	Shift+Click
Go to the next/previous page of data	Page up / Page down
Create a new row at the bottom of the grid	Down arrow (from the last row)
Move to the first record	Ctrl+Home

TO DO THIS	PRESS
Move to the last record	Ctrl+End
Select or clear all rows	Ctrl+Shift+M
Move to the first marked row	Alt+Shift+M, F
Move to the next marked row	Alt+Shift+M, L
Move to the previous marked row	Alt+Shift+M, P
Moved to the last marked row	Alt+Shift+M, N
Execute the default action in a grid  <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> <p>[!NOTE] This shortcut is enabled when focus is on a cell containing a hyperlink and all cells in that column have hyperlinks.</p> </div>	Enter
Toggle focus between the selected row and the header row	Alt+Shift+H
Make the current column larger/smaller (with focus in the header row)	Right arrow / Left arrow
Open grid filtering for the current column (with focus in the header row)	Enter

## Input control shortcuts

TO DO THIS	PRESS
Open the hyperlink	Ctrl+Enter
Enter the session date in a date field	D
Enter the current date in a date field	T
Open lookup, combo box, date picker, drop dialog box	Alt+Down arrow
Close lookup, combo box, date picker, drop dialog box	Esc
Move focus into a lookup (when the lookup is already open)	Alt+Down arrow
Open the control's enhanced preview	Alt+Up arrow
Select text in the current field	Ctrl+A
Enter/leave the text area in an HTML editor control	Alt+Down arrow / Alt+Up arrow
Switch focus between the text area and the toolbar in an HTML editor control	F6

## Messaging shortcuts

TO DO THIS	PRESS
Go to the Message Center	Ctrl+Shift+F7
Go to the Message Bar	Ctrl+F7

## Navigation shortcuts

TO DO THIS	PRESS
Go to the dashboard	Alt+Shift+Home
Move to the navigation bar	Alt+M,N or Alt+Shift+F1
Move to the company picker	Ctrl+Shift+O
Search for a page	Ctrl+/ or Alt+G
Open the help pane	Ctrl+?
Open the trace parser	Alt+Shift+T
Move to the navigation pane	Alt+F1
Add/remove form as a favorite (with focus on a form in the navigation pane)	Shift+F
Move to the standard Action Pane	Alt+M,A or Ctrl+F6
Move to the Filter pane (which may include opening it)	Alt+M,F or Ctrl+F3
Move focus to the page content (with focus in the Filter pane)	Alt+M,M or Ctrl+F3
Move to the navigation list on Details form (which may include opening it)	Alt+M,S or Ctrl+F8
Move focus to the page content (with focus in the navigation list)	Alt+M,M or Ctrl+F8
Close the navigation list on Details form (with focus in the navigation list)	Esc
Move to the main page content (with focus in another pane)	Alt+M,M
Move to the FactBox pane (which may include opening it)	Alt+M,B or Ctrl+F2

## Personalization shortcuts

TO DO THIS	PRESS
Transition the page into personalization mode	Ctrl+Shift+P
Use the Select tool (when in personalization mode)	S
Open the selected control's quick personalization dialog box (when using the Select tool)	Space or Enter
Use the Move tool (when in personalization mode)	M
Select the current control as the one to move (when using the Move tool and no control has been selected to move yet)	Space or Enter
Clear the control to move (when using the Move tool)	Esc
Move to the next position for the selected control (when using the Move tool)	Tab or Right arrow or Down arrow
Move to the previous position for the selected control (when using the Move tool)	Shift+Tab or Left arrow or Up arrow
Use the Hide tool (when in personalization mode)	H
Switch whether the current control is visible or hidden (when using the Hide tool)	Space or Enter
Use the Skip tool (when in personalization mode)	K
Switch whether the current control is in the tab sequence (when using the Skip tool)	Space or Enter
Use the Edit tool (when in personalization mode)	E
Switch whether the current control is editable or read-only (when using the Edit tool)	Space or Enter
Use the Summary tool (when in personalization mode)	U
Switch whether the current control is a summary field in the current fast tab (when using the Summary tool)	Space or Enter
Use the Add tool (when in personalization mode)	A
Select the control whose container will be used to insert the new fields (when using the Add tool)	Space or Enter
Import a personalization (when in personalization mode)	I
Export a personalization (when in personalization mode)	X
Clear this page's personalizations (when in personalization mode)	Ctrl+C

TO DO THIS	PRESS
Move focus between the personalization toolbar and the page (when in personalization mode)	T
Exit personalization mode (when in personalization mode)	Esc

## Segmented entry shortcuts

TO DO THIS	PRESS
Open the drop-down list (when the drop-down list is closed)	Alt+Down arrow
Move focus into the input field for the current segment in the drop-down list (when the drop-down list is already open)	Alt+Down arrow
Close the drop-down list	Alt+Up arrow
Close/open the right portion of the drop-down list	Alt+Left arrow / Alt+Right arrow
Switch between "Show valid" and "Show all" modes	Alt+W
Select the value from drop-down list and move to the next segment	Enter
Move to the next/previous control on the page (when focus is in input control)	Tab / Shift+Tab
Move to the next/previous input field in the flyout (when focus is in the drop-down list)	Tab / Shift+Tab
Move up/down a row in the lookup	Up arrow / Down arrow
Move up/down a page in the lookup	Page up / Page down
Move to the top/bottom of the lookup	Home / End

## Task recorder shortcuts

TO DO THIS	PRESS
Stop the recording (while recording)	Alt+R,S
Toggle the visibility of the Task recorder pane (while recording)	Alt+R,T
Toggle Rails mode (while playing a task guide)	Alt+R,L
Go to the previous step (while playing a task guide)	Alt+R,P
Go to the next step (while playing a task guide)	Alt+R,N



TO DO THIS	PRESS
Toggle focus between the page and the animated pop-up prompt (while playing a task guide)	Alt+R,F
Expand/collapse the animated pop-up prompt (while playing a task guide)	Alt+R,C
Show more/less information in the animated pop-up prompt (while playing a task guide)	Alt+R,M

## Additional resources

### [Keyboard shortcuts for missing account analysis](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Change the banner or logo

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following procedure lists the steps that system administrators can use to update the banner or logo image that is displayed for a legal entity. The demo data company used to create this procedure is USMF.

1. Go to **Navigation pane > Modules > Organization administration > Organizations > Legal entities**.
2. In the list on the left, select the legal entity for which you want to update the banner or logo. If it is already selected, go to the next step.
3. Expand **Dashboard image** tab.
4. Select **Change**.

The ideal resolution for a banner image is 1920 x 281 pixels. The ideal width for a logo image is 350 pixels.

5. Select **Save**.
6. Go to **Navigation pane > Modules > Common > Common > Default dashboard**. You should see the new banner or logo image on the dashboard.

You may need to change your company, using the company picker, to the one you uploaded the banner for.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Navigation search

2/18/2021 • 2 minutes to read • [Edit Online](#)

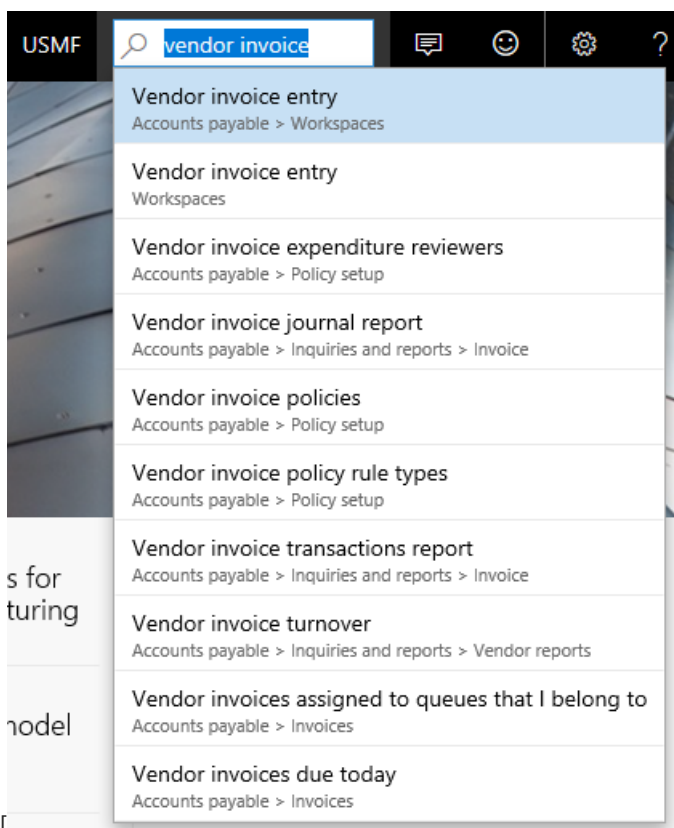
This topic explains how to use the search functionality to navigate to pages.

The application includes a number of areas and pages to help you perform various tasks. To quickly find the pages that you need to complete your tasks, use the navigation search feature.

To use this feature, click the **Search** icon to display the **Search** box. You can then type one or more words in the box. The system instantly searches for relevant pages in the application that match the words that you entered. For example, you could type "vendor invoice" as the input, and then the system displays results that match that input.

## NOTE

The **Search** box helps you find and navigate to pages. It will not help you find specific data or actions.



## Quickly navigate to a particular page

The navigation search feature also serves as a great way for you to quickly navigate to a particular page. For example, if you are an accounts payable clerk who frequently uses the **Payment journal** page, you could enter "payment journal" in the **Search** box. Because the input is an exact match for the page title, the page is listed at the top of the search results, and you can quickly navigate to it.

The search results list displays the page title as well as the navigation path. This shows the location of the page in the application. It also helps you differentiate between two or more similar pages in the results.

When you search for a page, your input is matched against the page title, as well as its navigation path. For

example, if you enter "receivable" in the **Search** box, you will see results for the pages available to you in the Accounts receivable area – even though the page titles do not include the word "receivable."

## Quickly navigate to a page based on the technical form name

The navigation search functionality also includes a much-requested feature for power users: the ability to quickly navigate to a page based on the technical form name. Many users are so familiar with the system that they know the exact form names they work with. If you are one of these users, you can enter **form:** followed by the name of the form you are looking for. For example, if you enter **form: vendinvoice**, the search results will show all pages where the form name starts with **vendinvoice**.

## Administration and security

From an administration and security perspective, the navigation search functionality only surfaces two types of results:

- Pages that are enabled in the current configuration (via configuration keys).
- Pages that the user has access to based on the user's role.

The list of search results is limited to 10 items. If you do not find what you're looking for in the results, you should try refining or updating the input.

## Development

From a development perspective, the navigation search functionality is easy to leverage because there is virtually no delay between the deployment of menu items and their ability to show up in search results. As long as the menu items are linked to from either the navigation pane or the dashboard, they will automatically become searchable.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Personalize the user experience

2/18/2021 • 22 minutes to read • [Edit Online](#)

This topic explains how you can personalize the app and covers the following subjects:

- **System-wide options** – These personalization options are made on a setup page and are available to all users. Examples include the color theme and time zone.
- **Restricted personalization access** – At this access level, user actions that are associated with typical page usage are automatically saved by the app and restored the next time that you visit the page. For example, the app stores the width of grid columns if you adjust them, and the expanded or collapsed state of FastTabs.
- **Full personalization access** – At this access level, users have access to all personalization capabilities in the app. In particular, they have access to the **Personalization** toolbar.
- **Sharing personalizations** – Users who have full personalization access can export their page personalizations and share them with other users.
- **Administration of personalizations** – Privileged users can access the **Personalization** administration page to manage all personalizations at an organizational level.

## System-wide options for the current user

The **User options** page contains several system-wide settings for the current user. These options are available to all users, even users who haven't been given any access to personalization. To open the **User options** page, select the **Settings** button on the navigation bar, and then select **User options**. The **User options** page has four tabs that contain various user settings:

- **Visual** – Select a color theme and the default size of elements on pages.
- **Preferences** – Select default values that are used every time that you open the system. These values include the default company, the initial page, and the default view/edit mode. (The view/edit mode determines whether a page is locked for viewing or opened for editing every time that you open it.) This tab also includes options for the language, the time zone, and date, time, and number formats. Finally, this tab includes several miscellaneous preferences that vary from release to release.
- **Account** – View or adjust your user name and other account-related options.
- **Workflow** – Select workflow-related options.

In addition to changing your user settings, you can also view and delete your usage data and personalizations from the **User options** page. To see your usage data, select **Usage data** on the Action Pane. On the **Personalization** tab, you can view and manage the personal changes that you've made to pages in the system. On this tab, you can also reset feature callouts (that is, the pop-up windows that introduce new system features). You will then be alerted again about previously encountered features.

### NOTE

If the [Saved views](#) feature is turned on, you can view and manage your personalizations by selecting **Personalization** on Action Pane on the **User options** page.

## Restricted personalization access (formerly implicit personalizations)

At the **restricted personalization access** level, user actions that are associated with typical page usage are automatically saved by the app and restored the next time that you visit the page. No explicit save action is required.

Here is a list of the actions that fall under typical page usage and are covered by restricted personalization access:

- **Grid column widths** – You can adjust the width of a column in a grid by selecting the sizing bar to the left or right of the column header, and then sliding it left or right until the column is the desired width. The app stores the width that you set for a column. Then, the next time that you open that page, the column will be resized to that width.
- **Grid footer and column totals** – (*Available only when the new grid control is turned on*) You can decide whether a total should be shown at the bottom of any numeric column in a grid, and whether the grid footer should be visible. The app stores these preferences and applies them the next time that you open the page. For more information, see [Grid capabilities](#).
- **FastTabs** – Some pages have expandable sections that are known as *FastTabs*. The app stores information about the FastTabs that you've expanded or collapsed. The next time that you open the page, the same FastTabs will be either expanded or collapsed, based on your last interaction with the page. In some cases, you can help improve system performance by collapsing a FastTab, because the app doesn't have to retrieve the information for FastTabs until they are expanded. As is explained later in this topic, you can also change the order of the FastTabs on a page.
- **FactBoxes** – Some pages have a **Related information** pane that shows read-only information that is related to the current subject of the page. Each section in the **Related information** pane is known as a *FactBox*. You can expand or collapse the **Related information** pane, and you can also expand or collapse individual FactBoxes. The app stores these preferences. The next time that you open the page, the **Related information** pane and the individual FactBoxes will be either expanded or collapsed, based on your last interaction with the page. In some cases, you can help improve system performance by collapsing the **Related information** Pane or a FactBox, because the app doesn't have to retrieve the information for FactBoxes until they are expanded.
- **Action Panes** – An *Action Pane* appears near the top of most pages. The Action Pane contains buttons for many of the actions that you can perform on the current page. These buttons are often organized on tabs. You can *pin* the whole Action Pane open, or you can have it collapsed by default. The next time that you open the page, the Action Pane will be either open or collapsed, based on your last interaction with the page. If you pinned the Action Pane open, the last tab that you were using will be shown.
- **QuickFilters** – A *QuickFilter* appears above many grids. The QuickFilter lets you filter the grid based on a single column that you select. The app stores the column that you filtered on. Then, the next time that you open that page, the grid will use that same column for filtering by default. However, you can still select a different column to filter the grid on.
- **Column header filters** – When you filter a grid by using *column header filters*, you can change the filter operator as you require to find the data that you want. For example, you can change the operator from **begins with** to **is exactly**. Every time that you use a column header filter and change the filter operator, the app stores the change. Then, the next time that you filter on that column, the filter operator will be restored.
- **Navigation pane** – You can open the *navigation pane* by selecting the **Expand the navigation pane** button in the upper left of any page. (This button is sometimes referred to as the **Menu button**, *hamburger*, *hamburger menu*, or *hamburger button*.) You can pin the navigation pane open, or you can have it collapsed by default. After you pin the navigation pane open, the app will keep it open until you collapse it.

## Full personalization access (formerly explicit personalizations)

At the **full personalization access** level, users have access to all the personalization capabilities that the app provides. Because different people and companies have different needs when they interact with the app, especially in terms of utilized fields, personalization provides tools that let users and organizations tailor the way that information is ordered and interacted with in the app. These capabilities are key to providing simplified, optimized experiences in the app that are tailored to you and your organization.

If the [Saved views](#) feature is turned on, an explicit save is required to persist these changes to the user

experience for a specific view. When the **Saved views** feature is turned off, these changes are automatically saved.

The following sections cover the extent of personalization capabilities that are available to users at the **full personalization access** level. Here are some of these capabilities:

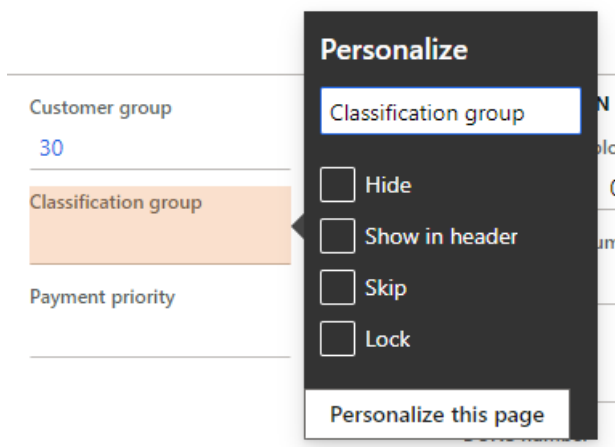
- Shortcut menu options
- The **Personalization** toolbar
- Adding tiles, lists, and links to workspaces
- Adding a summary from a workspace to a dashboard
- Personalizing the dashboard

### Shortcut menu options

Shortcut menus provide one way to change a page's interface so that it better meets your requirements or the requirements of your organization. (A shortcut menu is also known as a *right-click menu* or a *context menu*.)

Some of the most typical and important changes that can be made to a page are available directly as options on a shortcut menu. For example, if you want to add or hide columns in a grid, just right-click a column header, and then select **Insert columns** or **Hide this column**.

Additionally, the most basic types of personalizations are available by right-clicking an element and then selecting **Personalize**. (Note that not all elements on your page can be personalized.) When you use this personalization method, the element's *property window* appears.



You can use the property window to personalize an element in the following ways:

- Change the element's label.
- Hide the element so that it isn't shown on the page. The data in the field isn't deleted or modified. The information just isn't shown on the page any longer.
- Include the information in the FastTab's summary section (if the element is on a FastTab).
- Skip the field so that it never receives focus when you tab through the page.
- Prevent data in the field from being edited (for any record).
- Designate a field to be required for data entry. If no value has been entered in this field, it will appear with a red border and an asterisk to indicate this state. This option is only available starting in version 10.0.11 when the [Saved views](#) and **Designate fields as required using personalization** features are turned on.

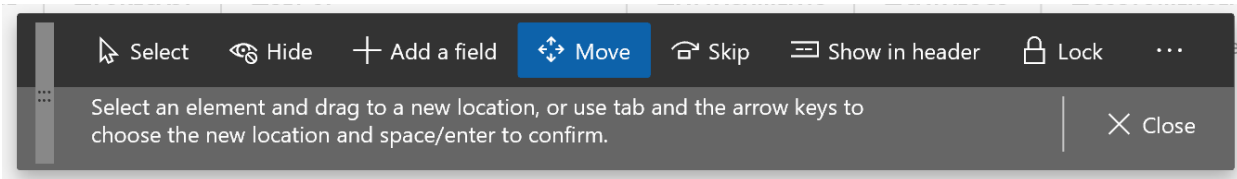
The property window might include other personalization capabilities, depending on the element. For example, the property window for a tile might let you promote that tile to a dashboard, and property windows for elements on the default dashboard might let you create a new custom workspace.

### The Personalization toolbar

If you want to make multiple changes to a page, or changes that aren't available through other mechanisms (for

example, if you want to reorder elements), you can use the **Personalization** toolbar. To open the **Personalization** toolbar, follow one of these steps:

- Select **Ctrl+Shift+P** from any element on the page.
- Select **Personalize this page** in an element's property window.
- Select **Personalize this page** in the **Personalize** group on the **Options** tab of any page's Action Pane.
- Select the **Settings** button (the gear symbol) on the navigation bar, and then select **Personalize**.



### Navigating the page

When the **Personalization** toolbar is open, the underlying page is read-only (in other words, you can't edit data), but it's still interactive. Specifically, you can expand or collapse the **Related information** pane, switch tabs, and expand or collapse sections, just as you would usually perform those actions on the page. To apply a personalization to a collapsible section or tab (for example, to hide a FastTab), you just have to select the button that appears next to that section or tab when it gains keyboard focus or when you hover over it.

### Personalization tools

The following tools are available on the **Personalization** toolbar:

- Use the **Select** tool to select and change the properties of an element. To use this tool, select the **Select** button on the toolbar, and then select the desired element. The element's property window appears, where you can change any of the properties of that element. You can repeat the process for other elements that can be personalized on the page. Note that some personalization properties might not be available in some scenarios. For example, you can't lock a field that is required.
- Use the **Hide** tool to hide an element on the page. To use this tool, select the **Hide** button on the toolbar, and then select the element to hide. When you use the **Hide** tool, all elements that are currently hidden are made visible, but they are shown in a shaded container. You can then make an element visible by selecting it. To see how the page will look when elements are hidden, switch to another personalization tool or close the personalization toolbar.
- Use the **Add fields** tool to add fields to your page. When you use this tool, you can add only fields that are part of the page definition. For information about how to create new fields that aren't part of the current page definition, see [Create and work with custom fields](#). After you select the **Add fields** button on the toolbar, you must first select the grid or section where you want to add a field. A dialog box will show the list of fields that are related to the selected grid or section. In the dialog box, select one or more fields to add, and then select **Update**. To remove a field that you previously added, repeat the process, but clear the selection of the field in the dialog box.
- Use the **Move** tool to move an element to a different location in the current group of elements. Note that you can't move an element outside its parent group. To use this tool, select the **Move** button on the toolbar, and then select the element to move. When you select an element, the app determines the locations where the element is allowed to be moved. These locations are known as *drop zones*. As you drag the element around in the current group, each drop zone is shown as a colored, bold line next to the area where the element can be dropped.
- Use the **Skip** tool to remove an element from the page's keyboard tab sequence. When you select the **Skip** button on the toolbar, all elements that are currently skipped are shown in a shaded container. You can interactively remove or add fields to the tab sequence.
- Use the **Show in header** tool when you want a field to appear in the FastTab's summary section. When



you select the **Show in header** button on the toolbar, all fields that have been selected as summary fields are shown in a shaded container. You can interactively add fields to the FastTab summary and remove fields from the summary by selecting the fields.

- Use the **Require** tool to designate an element as required for data entry. When you select the **Require** button on the toolbar, all elements that have been personalized to make them required are shown in a shaded container. You can then make them not required again. This option is available in version 10.0.12 and later when the **Designate fields as required using personalization** feature is turned on.
- Use the **Lock** tool to mark an element as either editable or noneditable. When you select the **Lock** button on the toolbar, all elements that are currently noneditable are shown in a shaded container. You can then make them editable again. Note that some fields are required and can't be made noneditable. A padlock symbol appears next to those fields.
- Use the **Add an app from Power Apps** tool to embed an app that was created by using Microsoft Power Apps into the page. For detailed information about how to embed an app from Power Apps into a page, see [Embed apps from Power Apps](#). This option is available only when the [Saved views](#) feature is turned off.
- Use the **Add an app** button to embed an app, either one created from Microsoft Power Apps or a third-party, into the page. This option is only available when the [Saved views](#) feature is turned on.
- Use the **Clear** tool to reset the page to its default, installed state. All personalizations on the current page will be cleared. You can't undo this action. Therefore, use this tool only if you're sure that you want to reset the page. When the **Saved views** feature is turned on, this tool clears the personalizations for the current view.
- Use the **Import** tool to load a personalization from a file that you or someone else previously created.
  - When the **Saved views** feature is turned off, you can choose whether to add or replace your existing personalizations with the personalizations that are being imported for the page. You can't undo this action. Therefore, after you import personalizations, you must manually clear or undo any changes that you don't want.
  - When the **Saved views** feature is turned on, the imported personalizations will become a view on the page. If the view already exists, you will have the option to skip the import, replace the current view that has the same name, or rename the imported view.
- Use the **Export** tool to save your personalizations for the page to a file. You can then share your personalizations with other users. Those users just have to import the file that contains your personalizations for the page. When the **Saved views** feature is turned on, this tool saves your current view to a file for sharing.
- Select the **Close** button to close the **Personalization** toolbar and return the page to its previous interactive state.

Traditionally, when the **Personalization** toolbar is used, your personalizations take effect as soon as you make them. However, if the [Saved views](#) feature is turned on, you must explicitly save personalizations to a view that you choose.

In some cases, when you select a tool, a padlock symbol appears next to an element. This symbol indicates that you can't modify the element properties that are related to the selected tool, because changes to those properties will prevent the page from working correctly.

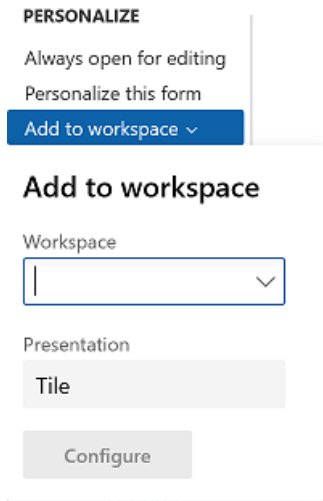
### **Adding tiles, lists, and links to a workspace**

For some pages that include lists, the **Add to workspace** personalization feature is available in the **Personalize** group on the **Options** tab of the Action Pane. This feature lets you push relevant information from the current list to a specific workspace. The information that appears in the workspace can be based on either

the whole list, or a filtered and sorted version of the list. You can also specify whether the information appears in the workspace as a list, a summary tile that can show the number of items in the list, or a link.

#### NOTE

If the [Saved views](#) feature is turned on, the content that you push to a workspace is directly linked to a view. The view's query is used to retrieve data into the workspace, and the corresponding tile or link in the workspace opens the page to that view, so that the view's query and personalizations are applied to it. If the view is updated, the corresponding workspace elements will be adjusted to the new view definition.



- To add a list to a workspace, first sort or filter the list on the page so that it shows the information as you want it to appear in the workspace. (If the **Saved views** feature is turned on, you can't continue until you save a view that has these conditions.) Then select **Add to workspace**. Select a workspace, and then, in the **Presentation** field, select **List**. After you select **Configure**, a dialog box appears, where you can select the columns that should appear in the list in the workspace. You can also specify the label that is used for the list in the workspace.
- To add a tile to a workspace, first filter the list on the page so that it shows the data that should be summarized or that you want quick access to. (If the **Saved views** feature is turned on, you can't continue until you save a view that has these conditions.) Then select **Add to workspace**. Select a workspace, and then, in the **Presentation** field, select **Tile**. After you select **Configure**, a dialog box appears, where you can specify the label that should be used for the tile in the workspace. You can also specify whether the tile should show a count. After the tile is added to the workspace, you can select it to open the current page from the workspace. You can then view the filtered list that is associated with the tile.
- To add a link to a workspace, first filter the list on the page so that it shows the data that you're interested in. (If the **Saved views** feature is turned on, you can't continue until you save a view that has these conditions.) Then select **Add to workspace**. Select a workspace, and then, in the **Presentation** field, select **Link**. After you select **Configure**, a dialog box appears, where you can specify the label that should be used for the link. You can also optionally specify a label for a new section that contains this link.

After you've added a list, tile, or link to a workspace, you can open that workspace and rearrange the elements in it as you want.

#### Adding a summary from a workspace to a dashboard

Some workspaces contain count tiles (that is, tiles that have numbers on them), and you might want those tiles to appear on your dashboard too. In a workspace, right-click a count tile, select **Personalize**, and then, in the tile's property window, select **Pin to dashboard**. The next time that you open and refresh the dashboard, the count will appear below the navigation tile for that workspace. You can select that count to go directly to the data that it represents.

## Personalizing your dashboard

The dashboard is often the first page that you see when you open the app. It can be personalized like any other page in the system, by using the same mechanisms that are described earlier in this topic.

### WARNING

Currently, when you hide content on the dashboard, it's important that you directly target a tile, not the space around it. If you hide the group around a tile, there could be unexpected results if more tiles are added later, or if the system is switched to a different language.

One unique personalization capability that is available on the dashboard is the ability to add tiles.

- If the **Full-page apps** feature is turned off, you add a new tile by right-clicking an element on the dashboard and then selecting **Add a workspace**. A new workspace tile is created at the bottom of the dashboard. You can rename this new workspace tile as you want. You can also add lists, tiles, and links to the workspace, as described in the [Adding tiles, lists, and links to a workspace](#) section of this topic.
- If the **Full-page apps** feature is turned on, you add a new tile by right-clicking an element on the dashboard and then selecting **Add an app**. In the dialog box, select whether you want to add a tile for a new workspace or a tile that has content from Power Apps or a website. Then follow the steps to configure the option that you selected. A new tile is created at the bottom of the dashboard.

## Sharing personalizations

After you personalize a page, you can share your personalizations with other users by exporting the personalized page. You can then ask other users to import the personalization file. Alternatively, you can give your personalizations to a user who has admin privileges. That user can then apply your personalization file to many users at the same time by using the **Personalization** administration page.

## Administration of personalizations

The **Personalization** page is the central hub for managing personalizations at an organizational level. The content and capabilities on this page depend on whether the **Saved views** feature has been turned on.

For customers who have turned on the **Saved views** feature, see the "Managing views globally" section in the [Saved views](#) topic.

For customers who haven't yet turned on the [Saved views](#) feature, this page has four tabs:

- **Apply** – You can import or select a personalization for one or more users. To apply a personalization to one or more users, first select a role and users who have that role. Then either select an existing personalization to apply to the selected users, or import a personalization file. The personalization is validated and will be applied to all the selected users the next time that they open the selected page.
- **Clear** – You can clear all personalizations for a page or workspace for one or more users. First select a page or workspace to see a list of the users who have personalized it. Then select the users who should have personalizations for that page or workspace cleared, and select **Clear**. All personalizations that the selected users have applied to the selected page or workspace are deleted. This action can't be undone. However, if a personalization was saved for the page or workspace, that personalization can be reimported.
- **Users** – Select a user to see a list of the pages that the user has personalized. You can then turn that user's ability to use personalizations for specific pages, or for the whole system, on or off. You can also import, export, or clear a personalization for the user. In addition, you can reset feature callouts for the user. In this case, if the user previously dismissed any pop-up windows that introduce new features, they will appear again the next time that the user encounters those features.
- **System** – You can temporarily turn off personalization for all users in the system. In this case, all

personalizations are deleted for all users, and all pages are reset to their default state. If you turn personalization back on later, all personalizations are reapplied. You can also permanently delete all personalizations for all users in the system. Personalizations that have been deleted can't be recovered. Therefore, before you perform this task, be sure to export any personalizations that you might want later.

## Personalizing inventory dimensions

When you personalize the setup of inventory dimensions on a page, consider the settings that have been created by using the **Display dimension** option. For example, you use personalization to hide a column for the Batch number inventory dimension, but the column appears the next time that the page is opened. This behavior occurs because the **Dimension display** settings control the inventory dimension columns that are shown. The **Dimension display** settings apply across all pages and override any personalized setup of inventory dimension fields on individual pages.

Therefore, in the preceding example, if you don't want the column for the Batch number inventory dimension to appear on a page, you must clear that dimension as part of the **Display dimensions** option for that page.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Saved views

2/18/2021 • 20 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [Service update availability](#).

## Introduction

Personalization plays an important role in allowing users and organizations to optimize the user experience to meet their needs. For more details on personalization, see [Personalize the user experience](#).

Traditional personalization let users have only one set of personalizations per page. The **Saved views** feature expands on personalization in several important ways:

- Views permit users to have multiple named sets of personalizations per form, which they can quickly switch between as needed. This allows a user to create multiple optimized views of a page, where each view has been tailored to fit the needs of performing a particular business task.
- Views created for particular page types can also include user-added filters or sorts, which allows users to quickly return to commonly filtered datasets. See the [What pages support views](#) section for more details.
- Views can be published to users in specific security roles and specific legal entities. Therefore, any user who has a specified role and access to a specified legal entity can access and use that view, even if that user doesn't not have permission to personalize. This publish capability lets organizations define corporate, standard views that are optimized for their business. For more information, see the [Managing personalizations at an organizational level with views](#) section.
- Unlike traditional personalization, views aren't automatically saved when a user performs personalizations or filters a list. Explicit saves are required to give users the flexibility to create a view before or after the changes that are associated with that view have been made. This requirement also ensures that view definitions aren't unintentionally changed by filters or personalizations that aren't intended for long-term use. Items that the system automatically stores as part of typical page usage (for example, column widths, or the expanded or collapsed state of sections) will be saved per view.
- Views can be added to workspaces as tiles, lists, or links. Therefore, a filtered data set can be surfaced in a workspace, and users can associate a set of personalizations that is relevant to that data set with a tile or link.

## Switching between views

After views have been made available for an environment, the top of any page that supports views will include a collapsed view selector control that shows the name of the current view.

There are two size variations to the view selector:

- **Large view selectors** – Pages that prominently feature a list will have a larger view selector for a few reasons. Most importantly, the larger view selector indicates the pages where the view can include user-defined filters. Because filters are included in the views, the larger selector size is also warranted as the view names will often be the best description of the data shown on the screen and the expectation is that users will switch between views more often on these page types.
- **Small view selectors** – All other full-screen pages (except workspaces and the dashboard) have a smaller view selector that appears next to the page caption. Views on these pages include only personalizations, not

user-defined filters. On these pages, the caption or record title is often the most important information at the top of the page. The smaller size of the view selector also reflects the lower frequency of view switching that is expected on these pages.

If you select the view name, the view selector is opened and shows the list of available views for the page.

- **Standard view** – The **Standard** view is the out-of-box view of the page, where no personalizations are applied.
- **Personal views** – The views without padlocks represent your personal views. These are views that either you have created or that an administrator has given to you.
- **Locked views** – Some views (such as the **Standard** view and any views that are published to your role) have a padlock symbol next to them in the view selector. This symbol indicates that you can't edit those views. However, changes that reflect page usage are automatically saved. These changes include changes to the width of a grid column, and changes to the expanded or collapsed state of a FastTab. Nevertheless, if you have personalization privileges, you can use the **Save as** action to make a personal view that is based on a locked view.
- **New views** – Published views that haven't yet been opened have a spark symbol to the left of the view name.

To switch to a different view, first open the view selector and then select the view that you want to load.

## Creating and modifying views

Unlike traditional personalization, views aren't automatically saved when a user personalizes the page, or when a user applies a filter to a list or sorts it. An explicit action is required to save these changes to a view. This requirement gives users the flexibility to create a view before or after the changes that are associated with that view have been made. It also ensures that view definitions aren't unintentionally changed by one-time filters or personalizations. Note that typical page usage items (for example, column widths, or the expanded or collapsed state of sections) are automatically saved to the current view, even for locked views.

To ensure that the current state of the view is known, when you start to change a view by personalizing or filtering it, an asterisk (\*) appears next to the current view name. This symbol indicates that you're looking at an unsaved, modified version of that view.

If you want to save those changes, follow these steps.

1. Select the view name to open the view selector.
2. To modify the existing view, select **Save**. Note that this action isn't available for locked views.
3. To create a new view:
  - a. Select **Save as**.
  - b. Enter a view name and (optionally) a description.
  - c. Select **Save**.

## Changing the default view

The default view is the view that the system tries to open when you first open the page. You should set the default view to the view that you expect to use most often.

### NOTE

There is a single, global default view across companies. If you change the default view, that view will be opened by default, regardless of the legal entity that you're currently in.

To change the default view for a page, follow these steps:

1. Switch to the view that you use as the default.
2. Select the view name to open the view selector.
3. Select **More** and then **Pin as default**.

Alternatively, when you create a new view (by using the **Save as** action), you can make that new view the default view by setting the **Pin as default** option before you save the view.

Note that, in some cases, the query that is associated with the default view isn't run when you first open a page. For example, if you open the page through a tile, the tile's query will be run, regardless of the query that is associated with the default view. Additionally, if you open a page that has a **Standard** view that already has a defined query, the original query will be run instead of the default view's query. In this case, you will receive an informational message when the view is loaded. If you switch views after the page has been loaded, the view query should be able to be run as expected. In version 10.0.10 and later, the informational message that you receive will have an embedded action that lets you load the default view's query directly.

## Managing personal views

The **Manage my views** dialog box gives you basic maintenance capabilities over your personal views and the order of views in the view selector. To open this page, select the view name to open the view selector drop-down menu, select **More**, and then select **Manage my views**.

For a list of available views for that page, the following set of actions are available.

- **Change the default view** – Use the **Pin as default** action to make the currently selected view the default view for this page.
- **Reorder your views** – Use the **Move up** and **Move down** actions to rearrange your views in a specific order.
- **Rename a view** – Use the **Rename** action to change the name of the currently selected personal view. This action is turned off for locked views.
- **Delete a view** – Use the **Delete** action to permanently delete the currently selected view from the page. There is no way to recover a view after you remove it.

Any changes made in this dialog box will take effect after you select the **Save** button.

## Managing personalizations at an organizational level with views

To help you understand how saved views help improve management of personalizations at an organizational level, this section describes some differences in personalization management with and without the **Saved views** feature.

Without views, administrators would apply a set of personalizations for a page to a user or a group of users via the Personalization page. If those users had personalization rights, the personalizations would be applied to that page. However, there was no ability to prevent users from further personalizing the page, which meant the organization could not ensure that its users had a consistent user interface. If any of those users didn't have personalization rights, the personalizations given to them by an administrator were not loaded. Further, if new users were hired into an organization, administrators needed to manually load a set of personalizations for the user. There was no automatic mechanism for specifying that a certain set of personalizations should be available for users in that role.

The **Saved views** feature makes organizational management of personalizations much easier, primarily because views can be published to groups of users. After a view has been published, any user who has one of the defined security roles and access to one the specified legal entities can see and use the view, even if that user doesn't have access to personalization. Although every user has a copy of the published view, where page usage

items are automatically applied, no user can save personalizations or query updates to a published view. In other words, published views are locked. Additionally, if new users are assigned to roles in legal entities that views were published to, they will automatically see the views that are associated with their roles and legal entities. No additional action is required by the admin. Likewise, if users change roles in an organization or are given access to different legal entities, they might no longer be able to access the views that were previously published to them. Again, no additional action is required by the admin.

Updates to a published view can easily be distributed to users by republishing the view to the appropriate security roles and legal entities.

The publish capability allows organizations to define corporate standard views that are optimized for their business, targeted at users in specific security roles.

## Publishing views

During the publishing process, views can be assigned to one or more security roles for one or more legal entities. Therefore, any user who has access to a legal entity and is assigned to one of those roles can access and use the views. However, the user can't edit the views. By default, system admins have access to the **Publish** action in the view selector drop-down menu. However, other trusted users in your organization can also be given access to view publishing via the new **Saved views administrator** role.

To publish a view, follow these steps:

1. Create and save a personal copy of the view that you want to publish.
2. With that view currently loaded, select the view name to open the view selector drop-down menu.
3. Select the **More** button and then select **Publish**. The Publish dialog box will open.
4. Enter a name for the view. The name that you enter is the name that users who receive this view will see in their view selectors. The names of published views for a page must be unique. No duplicate names are allowed, even if the list of roles or legal entities that the views are applied to differ.
5. **Update 10.0.17 or later:** If the **(Preview) Translation support for organization views** feature is turned on, you can add translations for your view name in as many languages as your organization requires by selecting the **Translations** button next to the **Name** field. The view name will then be shown to users in their current language. You can also set the default language to specify the translation that will be shown to users who are running languages that no translation is defined for.
6. Optional: Enter a description for the view, so that users who receive this view can better understand its purpose.
7. Determine whether the view should be published as the default view for the selected users. When a view is made the default view, users will see it the next time that they open the target page. The single, global default view of every targeted user will be changed. However, users can still change their default view after publishing has occurred.
8. Add the security roles that correspond to the users who are being targeted by this view.
9. Determine whether you want to publish the view to the child roles of each security role that is selected. If you do, select the **Include child roles** check box in the row for the appropriate security roles. Note that this check box isn't available for roles that don't have child roles.
10. Add the legal entities that this view should be available for.
11. Select **Publish**.

Note that in some environments, it may take some time (up to an hour) before users see the published view.



#### NOTE

Be aware of the following expectations when you publish a view to a legal entity, or when you publish a view as the default view.

- If you publish a view as the default view to all or some legal entities, you change the single, global default view of every targeted user. If a user has roles where multiple views are published as the default view, the last view that was published will be used as the user's default view.
- If you publish a view to a legal entity, but you don't publish it as the default view, users will initially see the view in the view selector only for the specified legal entities. However, after the view is loaded for the first time, it will always be in the user's view selector for that page, regardless of the legal entity.

## Modifying a published view

After you publish a view, you might find that you want to change it. Although you can't make live changes to a published view, because these views are locked for editing for all users (including publishers), you can republish a view to update it.

If the changes that you want to make to a published view only involve the publish parameters (the name and description of the view, or the security roles the view is published to), do the following:

1. Switch to the published view for the parameters that you want to update.
2. On the view selector drop-down menu, select **Republish**. If you're using version 10.0.12 or earlier, you must select **Publish** and then **Yes** to update the existing view.
3. Update the name, description, security roles, and legal entities for the view.
4. Select **Publish**. If you originally selected this published view as the default view, it will be the default view for users again after you republish it.

If the changes to the published view involve modifications of the personalizations or filters that are associated with the view, follow these steps.

1. Load the published view that you want to change.
2. Make the required changes to the local draft.
3. On the view selector drop-down menu, select **Republish**.
4. Select **Yes** to indicate that you want to publish the view together with its unsaved changes.
5. Adjust any publishing parameters that require adjustment, and then select **Publish**.

## Managing published views

Like managing personal views, the **Manage my views** dialog box gives users with publish privileges basic maintenance capabilities over that page's published views (in addition to their own personal views). To open this page, select the view name to open the view selector drop-down menu, select **More**, and then select **Manage my views**.

Although all users have a **My views** tab that show their personal views, users who have publish privileges also have an **Organization views** tab that shows all the published and unpublished views for that page. Because several users might be publishing views, it's important that you be able to manage the full list of published views, even if you aren't the user who published a given view.

For the list of all published views for the page, the following set of actions are available.

- **Republish** – Use the **Republish** action to republish a view after publishing parameters (name, description, security roles, or legal entities) are changed.
- **Publish** – Use the **Publish** action to publish a view that is currently unpublished.

- **Unpublish** – Use the **Unpublish** action to make a view inactive. The view will still be available in the system, but users won't see it in the view selector until the view is published again.
- **Save as personal** – Use the **Save as personal** action to create a personal draft copy of the published view. This capability can help you understand the contents of a view that wasn't published to you or that hasn't yet been published. You can also use it to edit and then republish a view.
- **Delete** – Use the **Delete** action to permanently delete a published or unpublished view. This action also removes the view for all users in the system. The removal of published views takes effect after the **Save** button is selected. After a view is deleted, it can't be recovered.

## Managing views globally

Although some management capabilities are surfaced on every page, as indicated in this topic, **system administrators** and **saved view administrators** can manage views more holistically for the system via the **Personalization** page. In particular, this page has the following sections and capabilities:

- **Published views** – This section lists all views that have been published for your organization. From here, you can republish a view after you adjust the security roles or legal entities that the view targets. You can also export, delete, or unpublish views. You can use the **Save as personal** action to create a personal copy of a view, so that you can update the view or gain a better understanding of its contents.
- **Unpublished views** – This section lists all the organization views in your system that aren't currently published. These views most often come into the system through the import capability. You can publish, export, or delete these views. The **Quick publish** action that was added in version 10.0.12 enables multiple views from this section to be published in one action, by using the existing security role and legal entity configurations. You can use the **Save as personal** action to create personal copies of these views, so that you can gain a better understand their contents.
- **Personal views** – This section lists all views that have been created by users in the system. From here, you can publish a personal view to the organization, or copy one or more of these views to other users. You can also export or delete these views as required.
- **User settings** – Select a user to view, or adjust the user's ability to use personalization either for the whole system or for specific pages that the user has visited. You can view and interact with the user's personalizations in the system. You can also delete all personalizations for that user or reset feature callouts for the user. If feature callouts are reset, any pop-up windows that introduced new features and that the user previously dismissed will appear again the next time that the user encounters those features.
- **System settings** – You can temporarily turn off personalization for all users in the system. In this case, no personalizations are applied for any user, and all pages are reset to their default state. If you turn personalization back on later, all personalizations are reapplied. You can also permanently delete all personalizations for all users in the system. Personalizations that have been deleted can't be recovered. Therefore, before you perform this task, be sure to export any personalizations that you might want later.

Users who have access to the **Personalization** page can also import personal or organization views by using the **Import views** button on the Action Pane. For organization views, you can select **Publish immediately** to make the views available to users without an additional explicit publish.

## Known issues

For a list of known issues with saved views, please see [Build forms that fully utilize saved views](#).

## Frequently asked questions

**How do I enable saved views in my environment?**

## NOTE

The **Saved views** feature requires the Personalization system in Finance and Operations to be enabled. If personalization is turned off for the entire environment, views will be disabled even if you follow steps below.

You can turn the **Saved views** feature on and off through Feature management in any environment. After it's turned on, saved views will be enabled in all subsequent user sessions.

### **What happens to existing personalizations when views are enabled?**

When views are enabled, any existing personalizations for a user and form are saved into a new view called **My view** that is automatically set as the default view. This is meant to ensure that there is a consistent user experience before and after views are enabled, except for the view selector control appearing on forms.

### **What pages support views?**

Views are available on most, but not all pages. Specifically, views are currently available on all full-screen pages except for dashboards and workspaces. Non-full-screen pages, which include dialog boxes, drop-down dialogs, lookups, enhanced previews, currently do not support views. View support for additional page types, such as workspaces and dialog boxes, may be considered for a future update.

### **Who is allowed to publish views?**

Only system admins and users who have been assigned to the **Saved views administrator** role have the rights to publish views.

### **Why am I not able to save filters with this view?**

There are a few reasons why a filter may not appear to save with a view:

- The page may not support saving filters as part of the view definition. Note that only pages with large view selectors allow personalizations and query modifications to be saved as a view. See the **Switching views** section for more information.
- The page in question may not properly support views, as it may ignore the view query completely or may operate on a temporary table whose data is not persistent.

### **What data will I see when I visit a page?**

For pages that have small view selectors (only personalizations can be saved to the view), you will see the same data as you always have when you visit the page.

For pages that have large view selectors (both personalizations and queries can be saved to the view), you will typically see the data that is linked to the query that is associated with your default view. There are two main exceptions:

- If you navigate to a page from a tile, the tile query will execute regardless of the query associated with the default view. If you created that tile after views have been enabled, selecting a tile will open the page with the view associated with that tile.
- If you navigate to a page and that entry point includes a query, the original query will execute originally in place of the default view's query. You should be alerted when this occurs via an informational message when the view is loading. You can also confirm by switching to this view after the page loads, as that should allow the view query to execute regardless.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Grid capabilities

2/18/2021 • 21 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [Service update availability](#).

The new grid control provides several useful and powerful capabilities that you can use to enhance user productivity, construct more interesting views of your data, and get meaningful insights into your data. This article will cover the following capabilities:

- Calculating totals
- Typing ahead of the system
- Evaluating math expressions
- Grouping tabular data (enabled separately using the **(Preview) Grouping in grids** feature)
- Freezing columns

## Calculating totals

In Finance and Operations apps, users have the ability to see totals at the bottom of numeric columns in grids. A footer section at the bottom of the grid shows these totals.

### Showing the grid footer

There is a footer area at the bottom of every tabular grid in Finance and Operations apps. The footer can show valuable information that is related to the data that appears in the grid. Here are some examples of this information:

- The number of selected rows in the table (when you select more than one record)
- Grand totals at the bottom of configured, numeric columns
- The number of rows in the dataset

This footer is hidden by default but you can turn it on. To show the footer for a grid, right-click on a column header in the grid and select the **Show footer** option. After you turn on the footer for a particular grid, that setting will be remembered until the user opts to hide the footer. To hide the footer, right-click on a column header and select **Hide footer**. (The placement of the **Show footer/Hide footer** action might move to a new location in a future update.

### Specifying columns with totals

Currently, no columns show totals by default. Instead, this is considered a one-time setup activity, similar to adjusting the widths of columns in grids. Once you specify that you want to see totals for a column, that setting will be remembered the next time you visit the page.

There are two ways to configure a column to show a total:

- Right-click in the column that you want to see a total for, and then select **Total this column**. This action causes three events to occur:
  1. The footer becomes visible.
  2. Your preference for seeing a total for this column is saved.

3. A calculation of totals is initiated for this column and any other columns that you previously configured to see totals for. The time that is required to show a total depends on the size of the dataset that you're totaling.
- After the footer is visible, select **Show total** in the footer area at the bottom of the column that you want to see a total for. If there are no configured columns, the **Show total** button will be available for all numeric columns.

After at least one column is configured for totals, the **Show total** buttons will be available only on hover or focus. The action of selecting **Show total** just saves your preference for seeing a total in this column, so that the preference is applied during future visits to the page. In the footer, this state is indicated by a dash that appears in the column. (Alternatively, if the dataset is small enough, a total is immediately shown.)

If you make a mistake and no longer want to see a total in a particular column, right-click on the column and select **Hide total** or select the **Hide total** button in the footer in that column. This preference will also be saved for future visits to the page.

### Calculating totals

When you come to a page with the footer visible and columns already configured for totals, totals may or may not be shown in the footer. The behavior is dependent on the size of the dataset on the page. If the dataset is sufficiently small, totals will be shown automatically, along with the number of rows in the dataset. If there are dashes in the footer under the columns you configured for totals, then the dataset is too large for the system to show totals immediately, and an explicit action is needed to calculate the totals. To do this, click the **Calculate** button in the footer, or right-click on a column you want a total for and select **Total this column**.

If the calculation is taking too long, you can cancel the operation by selecting the **Cancel** button. Sometimes, however, the dataset will be too large to calculate totals (a limit imposed by your organization), and you will instead be notified to filter your data more.

Totals will update automatically as you update, delete, or create rows in the dataset.

## Typing ahead of the system

In many business scenarios, the ability to quickly enter data into the system is very important. Before the new grid control was introduced, users could change data only in the current row. Before they could create a new row or switch to a different row, they were forced to wait for the system to successfully validate any changes. In an attempt to reduce the amount of time that users wait for these validations to be completed, and to improve user productivity, the new grid adjusts these validations so that they are asynchronous. Therefore, the user can move to other rows to make changes while previous row validations are pending.

To support this new behavior, a new column for the row status has been added to the right of the row selection column when the grid is in edit mode. This column indicates one of the following statuses:

- **Blank** – No status image indicates that the row has been successfully saved by the system.
- **Processing pending** – This status indicates that the changes in the row haven't yet been saved by the server but are in a queue of changes that must be processed. Before you take action outside the grid, you must wait for all the pending changes to be processed. Additionally, the text in these rows is italicized to indicate the unsaved status of the rows.
- **Invalid state** – This status indicates that some warning or message was triggered during the processing of the row, and it might have prevented the system from saving the changes in that row. In the old grid, if the save operation was unsuccessful, you were forced back into the row to fix the issue immediately. However, in the new grid, you're notified that a validation issue was encountered, but you can decide when you want to fix any issues in the row. When you're ready to fix an issue, you can manually move focus back to the row. Alternatively, you can select the **Fix this issue** action. This action immediately moves focus back to the row that has the issue, and lets you make edits inside or outside the grid. Note that the processing of subsequent

pending rows is stopped until this validation warning is resolved.

- **Paused** – This status indicates that processing by the server is paused because validation of the row triggered a pop-up dialog box that requires user input. Because the user might be entering data in some other row, the pop-up dialog box isn't immediately presented to that user. Instead, it will be presented when the user chooses to resume processing. This status is accompanied by a notification that informs the user about the situation. The notification includes a **Resume processing** action that will trigger the pop-up dialog box.

When users are entering data ahead of the place where the server is processing, they can expect a few degradations in the data entry experience, such as a lack of lookups, control-level validation, and entry of default values. Users who need a drop-down list to find a value are encouraged to wait for the server to catch up to the current row. Control-level validation and entry of default values will also occur when the server processes that row.

### Pasting from Excel

Users have always been able to export data from grids in Finance and Operations apps to Excel by using the **Export to Excel** mechanism. However, the ability to enter data ahead of the system enables the new grid to support copying tables from Excel and pasting them directly into grids in Finance and Operations apps. The grid cell that the paste operation is initiated from determines where the copied table begins to be pasted in. The contents of the grid are overwritten by the contents of the copied table, except in two cases:

- If the number of columns in the copied table exceeds the number of columns that remain in the grid, starting from the paste location, the user is notified that the extra columns have been ignored.
- If the number of rows in the copied table exceeds the number of rows in the grid, starting from the paste location, the existing cells are overwritten by the pasted content, and any extra rows from the copied table are inserted as new rows at the bottom of the grid.

## Evaluating math expressions

As a productivity booster, users can enter mathematical formulas in numeric cells in a grid. They don't have to do the calculation in an app outside the system. For example, if you enter  $=15*4$  and then press the **Tab** key to move out of the field, the system will evaluate the expression and save a value of **60** for the field.

To make the system recognize a value as an expression, start the value with an equal sign (=). For more information about the supported operators and syntax, see [Supported math symbols](#).

## Grouping tabular data

Business users often need to perform ad-hoc analysis of data. While this can be done by exporting data to Microsoft Excel and using pivot tables, the **Grouping in grids** feature, which is generally available in version 10.0.16/Platform update 40 and is dependent on the new grid control feature, allows users to organize their tabular data in interesting ways within Finance and Operations apps. Because this feature extends the **Totals** feature, **Grouping** allows you to get meaningful insights into the data by providing subtotals at the group level.

To use this feature, right-click the column that you want to group by, and select **Group by this column**. This action will sort the data by the selected column, add a new **Group by** column to the beginning of the grid, and insert "header rows" at the beginning of each group. These header rows provide the following information about each group:

- Data value for the group
- Column name (this information is especially useful when you have multiple levels of grouping)
- Number of data rows in this group
- Subtotals for any column configured to show totals

With [Saved views](#) enabled, this grouping can be saved by personalization as part of a view for quick access the next time you visit the page.

### Multiple levels of grouping

After you've grouped data by a single column, you can group the data by a different column by selecting **Group by this column** on the desired column. This process can be repeated until you have 5 nested levels of grouping, which is the maximum supported depth. At this point, you will no longer be able to group by additional columns.

At any point, you can remove the grouping on any column by right-clicking that column and selecting **Ungroup**. You can also remove the grouping from all columns by selecting **Grid options** and then **Ungroup all**.

Note, prior to version 10.0.16/Platform update 40, only one level of grouping is supported. In these versions, if the data is grouped and you select **Group by this column** for a different column, the original grouping is replaced.

### Expanding and collapsing groups

The initial grouping of data will have all groups expanded. You can create summarized views of the data by collapsing individual groups, or you can use group expanding and collapsing to assist in navigating through the data. To expand or collapse a group, select the chevron (>) button in the corresponding group header row. Note that the expand/collapse state of individual groups is **not** saved in personalization.

### Selecting and unselecting rows at the group level

In the same way that you can select (or unselect) all rows in the grid by selecting the check box at the top of the first column in the grid, you can also quickly select (or unselect) all the rows in a group by selecting the check box in the corresponding group header row. The check box in the group header row will always reflect the current selection state of rows in that group, regardless of whether all rows are selected, no rows are selected, or only some rows are selected.

### Hiding column names

When grouping data, the default behavior is to show the column name in the group header row. Starting in version 10.0.14/Platform update 38, you can choose to suppress the column name in group header rows by selecting **Grid options > Hide group column name**.

## Freezing columns

Some columns in a grid might be important enough for context that you don't want them to scroll out of view. Instead, you want the values in those columns always to be visible. In version 10.0.17, the **Freeze columns in grid** feature provides this flexibility to users.

To freeze a column, right-click in the column's header, and then select **Freeze column**. The first time that you complete this step, the selected column becomes the first column and will no longer scroll out of view. Any subsequent column that you freeze will be added to the right of the last frozen column. You can use the standard Move functionality to reorder frozen columns as you require. However, frozen columns can't be moved so that they appear among the set of unfrozen columns. Likewise, unfrozen columns can't be moved so that they appear among the set of frozen columns.

To unfreeze a column, right-click in the frozen column's header, and then select **Unfreeze column**.

Note that the row selection and row status columns in the new grid are always frozen as the first two columns. Therefore, when these columns are included in a grid, they will always be visible to users, regardless of the horizontal scroll position in the grid. These two columns can't be reordered.

## Frequently asked questions

## How do I enable the new grid control in my environment?

### 10.0.9 / Platform update 33 and later

The **New grid control** feature is available directly in Feature management in any environment. Like other public preview features, enabling this feature in production is subject to the [Supplemental Terms of Use Agreement](#).

### 10.0.8 / Platform update 32 and 10.0.7 / Platform update 31

The **New grid control** feature can be enabled in Tier 1 (Dev/Test) and Tier 2 (Sandbox) environments in order to provide additional testing and design changes by following the steps below.

1. **Enable the flight:** Execute the following SQL statement:

```
INSERT INTO SYSFLIGHTING (FLIGHTNAME, enabled, FLIGHTSERVICEID, PARTITION)
VALUES('CLIRectGridEnableFeature', 1, 0, 5637144576);
```

2. **Reset IIS** to flush the static flighting cache.

3. **Find the feature:** Go to the **Feature management** workspace. If **New grid control** does not appear in the list of all features, select **Check for updates**.

4. **Enable the feature:** Find the **New grid control** feature in the list of features, and select **Enable now** on the details pane. Note that a browser refresh is required.

All subsequent user sessions will start with the new grid control enabled.

## [Developer] Opting out individual pages from using the new grid

If your organization discovers a page that has some issues utilizing the new grid, an API is available starting in version 10.0.13/Platform update 37 to allow an individual form to use the legacy grid control while still permitting the rest of the system to utilize the new grid control. To opt out an individual page from the new grid, add the following call post `super()` in the `run()` method for the form.

```
this.forceLegacyGrid();
```

This API will be honored until the October 2021 release, when the new grid control becomes mandatory. If any issues require that this API be used, report them to Microsoft.

## [Developer] Size-to-available-width columns

If a developer sets the **WidthMode** property to **SizeToAvailable** for columns inside the new grid, those columns initially have the same width that they would have if the property were set to **SizeToContent**. However, they stretch to use any extra available width inside the grid. If the property is set to **SizeToAvailable** for multiple columns, all those columns share any extra available width inside the grid. However, if a user manually resizes one of those columns, the column becomes static. It will remain at that width and will no longer stretch to take up extra available grid width.

## Known issues

This section maintains a list of known issues for the new grid control.

### Open issues

- After enabling the **New grid control** feature, some pages will continue to utilize the existing grid control. This will happen in the following situations:
  - A card list exists on the page that is rendered in multiple columns.
  - A grouped card list exists on the page.



- A grid column with a non-react extensible control.

When a user first encounters one of these situations, a message will display about refreshing the page. After this message appears, the page will continue to utilize the existing grid for all users until the next product version update. Better handling of these scenarios, so that the new grid can be utilized, will be considered for a future update.

- [KB 4582758] Records are blurry when you change zoom from 100 to any other percentage
- [KB 4592012] Unexpected client error in IE11 when pasting multiple lines from Excel
  - Microsoft is not pursuing a fix for this issue

#### **Fixed as part of 10.0.16**

- [KB 4598335] Multi-line string controls do not respect their DisplayHeights in lists/cards
- [KB 4591891] Invoice proposal lines disappear when unmarking lines
- [KB 4592104] Unable to edit records after clicking "Fix issue" and moving to a different row without fixing the validation issue
- [KB 4594449] "Never" and "Clear" buttons missing inside the date picker
- [KB 4594448] Entering time is treated differently with the new grid
- [KB 4600059] Unexpected client error with email throttling
- [KB 4574584] Expense attachment preview not available when hovering over the receipt icon

#### **Fixed as part of 10.0.15**

- (Quality update) [KB 4594444] Unexpected client error with preview for segmented entry control
- [KB 4582723] Display options not showing when done later in the form life cycle
- [KB 4591988] Issues using the keyboard to select a value from a ReferenceGroup lookup
- [KB 4588958] Regression Suite Automation Tool (RSAT) test fails with error: TypeError: Cannot read property 'text' of undefined
- [KB 4591970] Unexpected client error when pasting from Excel was done immediately after clicking into the grid
- [KB 4591904] Data changes are not saved if after editing a control the user immediately clicked and opened the lookup of a different control
- [KB 4584752] Unexpected client error with Project invoice proposals page
- [KB 4584540] Unable to leave the grid after pasting a single row into a journal line
- [KB 4591908] When creating a new row, focusing is staying in the column you were in

#### **Fixed as part of 10.0.14**

- (Quality update) [KB 4584752] Unexpected client error with Project invoice proposals page
- [KB 4583880] Regression Suite Automation Tool (RSAT) tests fail on OpenLookup action with "Cannot read property RowIndex of undefined"
- [KB 4583847] Unexpected client error when navigating through lookups

#### **Fixed as part of 10.0.13**

- (Quality update) [KB 4584752] Unexpected client error with Project invoice proposals page
- (Quality update) [KB 4583880] Regression Suite Automation Tool (RSAT) tests fail on OpenLookup action with "Cannot read property RowIndex of undefined"
- (Quality update) [KB 4583847] Unexpected client error when navigating through lookups
- (Quality update) [Bug 471777] Cannot select fields in a grid to edit or create a mobile app
- [KB 4582727] Grid freezes after user gets dialog for items with multiple quantities
- [Bug 474851] Hyperlinks in reference group controls don't work
- [Bug 474848] Enhanced previews with grids do not display

- [KB 4582726] The RotateSign property isn't being respected
- [Bug 470173] Check boxes in inactive rows toggle when the whitespace in the cell is clicked
- [Bug 474848] Enhanced previews with grids do not display
- [Bug 474851] Hyperlinks in reference group controls don't work
- [Bug 471777] Cannot select fields in a grid to edit or create a mobile app
- [KB 4569441] Issues with rendering multi-column card lists, tooltips on images, and display options on some fields
- [KB 4575279] Not all marked rows are deleted in General Journal
- [KB 4575233] Display options are not restored after moving to another row
- [Bug 477884] Lookups return wrong value/record if new grid control is activated
- [KB 4571095] Product receipt posting occurs when accidentally pressing Enter (correct handling of a page's default action)
- [KB 4575437] Lookups with editable controls close unexpectedly
- [KB 4569418] Duplicate line created in delivery schedule form
- [KB 4575435] Enhanced preview sometimes persists even when the mouse pointer isn't near the field
- [KB 4575434] Lookup isn't filtering when the field has been modified
- [KB 4575430] Values in password fields aren't masked in the grid
- [KB 4569438] "Processing has stopped because of a validation issue" displays after marking lines while settling supplier transactions
- [KB 4569434] Refreshing the Legal entities form results in fewer records
- [KB 4575297] Focus keeps moving to the task recorder pane when editing and tabbing through a grid
- [KB 4566773] Correction transactions not showing as negative on voucher transactions inquiry
- [KB 4575288] Focus resets to the active row when selecting the border between rows in a simple list
- [KB 4575287] Focus doesn't return to the first column when using the down arrow to create a new row in journals
- [KB 4564819] Cannot delete lines in a free text invoice (because the datasource ChangeGroupMode=ImplicitInnerOuter)
- [KB 4563317] Tooltips/enhanced previews aren't shown for images

#### Fixed as part of 10.0.12

- [KB 4558545] Table controls don't update the contents of displayed items.
- [KB 4558570] Items are still shown on the page after the record has been deleted.
- [KB 4558572] Styling that is associated with the List Panel **ExtendedStyle** isn't applied.
- [KB 4558573] Validation errors can't be fixed when the required change is outside the grid.
- [KB 4558584] Negative numbers aren't rendered correctly.
- [KB 4560726] An "unexpected client error" occurs after swapping between lists is done by using a List View control.
- [KB 4562141] Grid indices are off after a new record is added.
- [KB 4562151] The **Validate** and **Copy** task recorder options aren't available for date/number controls.
- [KB 4562153] Multi-select check boxes aren't visible on list/card grids.
- [KB 4562646] You sometimes can't click outside the grid after you multi-select rows in the grid.
- [KB 4562647] Focus is reset to the first control in the **Publish** dialog box after a new row is added in the security roles grid.
- [KB 4563310] The enhanced preview isn't closed after a row is changed.
- [KB 4563313] An "unexpected client error" occurs in Internet Explorer when a value is selected in a lookup.
- [KB 4564557] Lookups and drop-down menus won't open in Internet Explorer
- [KB 4563324] Navigation doesn't work after the **Personnel management** workspace is opened.

### Fixed as part of 10.0.11

- [Issue 432458] Empty or duplicated lines are shown at the beginning of some child collections.
- [KB 4549711] Lines in a payment proposal can't be removed correctly after the new grid control is enabled.
- [KB 4558374] Records that require a polymorphic selector dialog box can't be created.
- [KB 4558375] Help text isn't shown on columns in the new grid.
- [KB 4558376] List Panel grids aren't rendered at the correct height in Internet Explorer.
- [KB 4558377] Combo box columns that have **SizeToAvailable** width aren't rendered on some pages.
- [KB 4558378] Drill-through sometimes opens the wrong record.
- [KB 4558379] An error occurs when lookups are opened where **ReplaceOnLookup=No**.
- [KB 4558380] The available space in the grid isn't filled immediately after part of the page is collapsed.
- [KB 4558381] Negative numbers aren't rendered correctly / Users sometimes become stuck after validation issues are encountered.
- [KB 4558382] Unexpected client errors occur.
- [KB 4558383] Controls outside the grid aren't updated after the last record is deleted.
- [KB 4558587] Reference groups that have combo boxes for replacement fields don't show values.
- [KB 4562143] Fields aren't updated after a row change / Grid processing becomes stuck after row deletion.
- [KB 4562645] An exception occurs when a lookup is opened while Regression Suite Automation Tool (RSAT) tests are running.

### Fixed as part of 10.0.10

- [Issue 414301] Some data from previous lines disappears when new lines are created.
- [Bug 417044] There is no empty grid message for list-style grids.
- [KB 4539058] Some grids (typically on FastTabs) sometimes aren't rendered (but they will be rendered if you zoom out).
- [KB 4549734] Active rows aren't treated as marked if the marking column is hidden.
- [KB 4549796] Values can't be edited in a grid when it's in view mode.
- [KB 4558367] Text selection is inconsistent when rows are changed.
- [KB 4558368] Multi-select via the keyboard is allowed in single-select scenarios.
- [KB 4558369] Status images disappear in the hierarchical grid.
- [KB 4558370] A new row isn't scrolled into view.
- [KB 4558372] The new grid becomes stuck in processing mode if the number of columns in content that is pasted in exceeds the number of remaining columns in the grid.
- [KB 4562631] Time values aren't formatted correctly.

### Quality update for 10.0.9/Platform update 33

- [KB 4550367] Time values aren't formatted correctly.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create and work with custom fields

2/18/2021 • 7 minutes to read • [Edit Online](#)

While there is an extensive set of fields out-of-the-box for managing a broad range of business processes, sometimes there is a need for a company to track additional information in the system. While programmers can be used to add those fields as extensions in the developer tools, the custom fields feature allows fields to be added directly from the user interface, thereby allowing you to tailor the application to fit your business using your web browser.

The ability to add custom fields is available in platform update 13 and later. Only users with special permissions have access to this feature.

This video shows how easy it is to add a custom field to a page: [Adding custom fields](#).

## Creating custom fields

After you've identified additional information that you would like to track in the application, you can create the custom field on the appropriate table and expose that new field on a page.

The following steps describe the process for creating a custom field and placing that field on a form.

1. Navigate to the form where the new field is needed.
2. Because the end goal is to expose the custom field on a form, the entry point for creating custom fields exists inside the personalization experience. Open the personalization toolbar by selecting **Options**, and then **Personalize this form**.
3. Click **Insert** and then **Field**.
4. Select the region of the form where you want to expose the new field. After selection, the **Insert fields** dialog box will display a list of existing fields that can be inserted into the selected region of the form.
5. Ensure that the field you are interested in does not already exist in the list. If it does, you can simply select that field in the list and click **Insert**.
6. Click the **Create new field** button above the list to initiate the process of creating a custom field. This will open the **Create new field** dialog box.

If you do not see the **Create new field** button, you do not have the necessary permissions to use this feature.

7. In the **Create new field** dialog box, enter the following information.
  - a. Select the database table where this field should be added. Note that only tables that support custom fields will appear in the drop-down list. See the section below for technical details on supported tables.
  - b. Select the data type for the new field. The available data types are checkbox, date, date time, decimal, number, picklist, and text.
    - If you choose the text data type, you can also specify the maximum length of the text that can be entered in this field.
    - If you choose the picklist data type, you can also select the set of valid values for the field.
  - c. Provide a name, label, and help text for the field. The name corresponds to the physical field name

in the database, whereas the label and help text are the text used to represent this field in the user interface.

8. If this is the only field that you need to create for this form, click **Save**. If you need to create additional fields, click **Save and new** and go back to step 7. Note that there is currently a limit of **20 custom fields per table**.
9. Leaving the **Create new field** dialog box will return you to the **Insert fields** dialog box. Any custom fields that were just added will be automatically marked in the field list to be inserted into the form.
10. Click **Insert** to insert the marked fields into the selected region of the form.
11. **Optional:** Enable **Move** mode from the personalization toolbar to move the new fields to their desired location in the selected region. See [Personalize the user experience](#) for more information about how to use the various personalization capabilities to optimize a form for your personal usage.

## Sharing custom fields with other users

After you have created a custom field and exposed it on a form, you might want to provide this updated page view that includes the new field to other users in the system. This can be accomplished in two different ways using the personalization capabilities of the product:

- The recommended route is through the system administrator, who can push a personalization to all users or a subset of users. See [Personalize the user experience](#) for more details.
- Alternatively, you can export your changes (called *personalizations*), send them to one or more users, and have each of those users import your changes. The **Manage** option on the personalization toolbar enables you to export and import personalizations.

## Managing custom fields

Management of all the custom fields in the system can be accomplished through the **Custom fields** page in the System administration module. This page allows users access to many capabilities, including:

- Viewing a list of all custom fields in the system.
- Limited editing of existing custom fields.
- Deleting custom fields.
- Exposing custom fields on data entities.
- Providing translations of custom field labels and help text.

### Viewing all custom fields

The **Custom fields** page provides visibility to all the custom fields that have been defined in the system. Simply select the table that you are interested in, and the page will update to show a list of the custom fields associated with that table. Choosing a custom field from the list will allow you to view all the details about the field.

### Editing custom fields

After a custom field has been created, only certain pieces of information about the custom field can be modified on the **Custom fields** page.

You *can* modify these attributes:

- Label
- Help text
- Length, for Text fields

You *cannot* edit the following attributes:

- Field name
- Data type

Additionally, for picklist fields, the set of valid values for the custom field can be reordered, and new values can be added; however, existing values for the picklist field cannot be removed. Remember to click **Apply changes** when you are done editing fields for a particular table so the changes are saved.

### Exposing custom fields on data entities

It may also be important to allow custom fields to be visible on data entities. Data entities are utilized in the [Office integration overview](#) feature, as well as for data import/export scenarios.

Follow these steps to expose a custom field on a data entity:

1. Select the custom field on the **Custom fields** form.
2. Expand the **Entities** section to view the set of relevant entities.
3. Click the **Edit** button.
4. Modify the **Enabled** field to be selected for each entity that should expose this field.
5. Click **Apply changes** to save your selections.

### Allowing custom fields to be displayed in other languages

Because custom fields may need to be accessed by users in a variety of languages, the **Custom fields** page provides a mechanism to allow the label and help text for a custom field to be translated into other languages.

The following steps describe the process for translating custom fields in other languages:

1. Select the custom field on the **Custom fields** page.
2. Select the **Translations** button in the Action Pane. This will open a drop-down menu with existing translations for this field.
3. The **Language** drop-down menu shows the set of languages for which translations have already been provided.

If you want to edit an existing translation, select the desired language from the menu and modify the values for the label and help text.

Otherwise, click the **Add language** button, select the desired language from the menu, and then provide translated values for the label and help text.

4. Click **OK** when you are finished.

### Deleting custom fields

In some rare cases, you may decide that a custom field is no longer needed. When this occurs, a system administrator can choose to delete the field from the **Custom fields** page. To do this, ensure the correct field is selected, click **Delete**, click **Yes** to confirm the deletion, and finally click **Apply changes**.

#### NOTE

This action cannot be undone, and will result in the data associated with the field being permanently deleted from the database.

## Appendix

### Who can create custom fields?

As a safeguard to the system, only system administrators are able to create custom fields by default. However, those power users whom the organization deems necessary can be given rights to create custom fields by a

system administrator using the **Runtime customization power user** security role. Users without this security role will not be able to create custom fields, but will still be able to see and interact with custom fields added by other users in the system.

### **What tables support custom fields?**

For performance and technical reasons, only tables that meet the following conditions currently allow custom fields to be added.

- The table must be tagged as one of these groups:
  - Group
  - WorksheetHeader
  - Main
  - Miscellaneous
  - Parameter
  - Reference
  - TransactionHeader
- The table cannot extend another table.
- The table cannot be marked as a system table.
- The table cannot be a temporary table.

### **Can I reference custom fields from the developer tools?**

Custom fields can only be managed through the user interface and cannot be referenced by code.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Embed canvas apps from Power Apps

2/18/2021 • 8 minutes to read • [Edit Online](#)

Microsoft Power Apps is a service that lets developers and non-technical users build custom business apps for mobile devices, tablets, and the web without writing code. Finance and Operations apps support integration with Power Apps. Canvas apps that you, your organization, or the broader ecosystem develop can be embedded into Finance and Operations apps to augment the product's functionality. For example, you might build a canvas app from Power Apps to supplement a Finance and Operations app with information that is retrieved from another system.

To learn more about embedding Power Apps, watch the short [How to embed Power Apps](#) video.

## Adding an embedded canvas app from Power Apps to a page

### Overview

Before you embed a canvas app from Power Apps into the client, you must find or build an app that has the desired visuals or functionality. This topic doesn't include a detailed description of the process for building apps. If you're new to Power Apps, see the [Power Apps documentation](#).

There are two ways to access a specific canvas app on a page when you're ready to embed the app. You can choose whichever approach fits your scenario better. The first approach uses the **Power Apps** button that has been added to the standard Action Pane. Apps that you add by using this approach appear as items on the **Power Apps** menu button. When you select one of these items, a side pane that contains the embedded app appears. Alternatively, you can embed an app directly on a page as a new tab, FastTab, or blade, or as a new section in a workspace.

When you configure your embedded canvas app, you can select a single field that you want to send as context to the app. This step enables the app to be responsive, based on the data that you're currently viewing.

#### NOTE

You can't currently use this mechanism to embed modeled apps.

### Details

The following procedure shows how to embed a canvas app from Power Apps into the web client.

1. Go to the page where you want to embed the canvas app. This page will be the page that contains data that must be passed to the app as input.
2. Open the **Add an app from Power Apps** pane:
  - Click **Options**, and then select **Personalize this page**. Under the **Insert** menu, choose **Power Apps**. Finally, select the region where you would like to add the app. If you want to embed the app under the Power Apps menu button, choose the Action Pane. If you want to embed the app directly onto the page, choose the appropriate tab, FastTab, blade, or section (if you're on a workspace).
  - If the app will be accessed using the Power Apps menu button, you can alternatively click the **Power Apps** menu button in the standard Action Pane, and then select **Add an app**.
3. Configure the embedded app:
  - The **Name** field indicates the text shown for the button or tab that will contain the embedded app. Oftentimes, you may want to repeat the name of the app in this field.



- The **App ID** field indicates the globally unique identifier (GUID) for the canvas app that you want to embed. To retrieve this value, find the app on [make.powerapps.com](https://make.powerapps.com), and then look in the **App ID** field under **Details**.
  - For **Input context for the app**, you can optionally select the field that contains the data that you want to pass to the app as input. See the section later in this topic titled [Building an app that leverages data sent from Finance and Operations apps](#) for details on how the app can access the data sent from Finance and Operations apps.
  - Choose the **Application size** that matches the type of app that you're embedding. Select **Thin** for apps built for mobile devices, and **Wide** for apps built for tablets. This ensures a sufficient amount of space is allotted for the embedded app.
  - The **Legal entities** FastTab provides the ability to choose which legal entities the app is available for. The default is to make the app accessible to all legal entities. This option is only available when the [Saved views](#) feature is disabled.
4. After confirming that the configuration is correct, click **Insert** to embed the Power App on the page. You will be prompted to refresh the browser in order to see the embedded app.

## Sharing an embedded app

After you've embedded a canvas app on a page and confirmed that it's working correctly with any data context that is passed from that page, you might want to share the app with other users in the system. To share an embedded canvas app, follow these steps.

1. [Share the canvas app](#) with the appropriate users, so that they can access the app in Power Apps.
2. Make sure that the targeted users have the appropriate personalizations, so that the embedded app appears when those users view the page. You can use either of the following approaches:
  - Recommended: Use the [Saved views](#) feature to create and publish a view that includes the embedded app. This approach ensures that all users who have the security roles that are targeted by the published view will see the app in Finance and Operations apps.
  - If you don't have the Saved views feature turned on, you can have the system admin push a personalization that includes the embedded app to all users or a subset of users. Alternatively, you can export your page's personalizations, and send them to one or more users. Each of those users can then import the personalizations. The personalization toolbar has actions that let you export and import personalizations.

### NOTE

If the canvas app has been shared with external users, those users can't use the embedded app inside Finance and Operations apps. However, they can access the app directly inside Power Apps. External users include guests and users who don't belong to the Microsoft 365 Azure Directory where the Finance and Operations app is deployed.

See [Personalize the user experience](#) for more details about the personalization capabilities in the product and how to use them.

## Building a canvas app that uses data that is sent from Finance and Operations apps

When you build a canvas app that will be embedded in a Finance and Operations app, one important part of the process is to use the input data from that Finance and Operations app. From the Power Apps development experience, the input data that is passed from a Finance and Operations app can be accessed by using the `Param("EntityId")` variable.

For example, in the OnStart function of the app, you could set the input data from Finance and Operations apps to a variable like this:

```
If(!IsBlank(Param("EntityId")), Set(FinOpsInput, Param("EntityId")), Set(FinOpsInput, ""));
```

## Viewing a canvas app

To view an embedded canvas app on a page in Finance and Operations apps, just go to a page that has an embedded app. Remember that apps can be accessed by using the **Power Apps** button on the standard Action Pane. Alternatively, they can appear directly on the page as a new tab, or FastTab, or blade, or as a new section in a workspace. When users first try to load an app on a page, they will be prompted to sign in. This step ensures that the users have the appropriate permissions to use the app.

## Editing an embedded app

After an app has been embedded onto a page, you may need to make some changes to the configuration of the app. For example, perhaps you want to modify the label associated with the embedded app or a new version of the app has been created and you need to update the App ID to point at the latest.

Follow these steps to edit the configuration of an embedded app:

1. Go to the **Edit the app** pane.
  - If the embedded app is accessed through the Power Apps menu button, right-click on the Power Apps menu button and select **Personalize**. Select the app that you want to configure from the **Select an app** drop-down menu.
  - If the embedded app appears directly on the page, select **Options**, and then select **Personalize this page**. Using the **Select** tool, click the embedded app.
2. Make the needed modifications to the app configuration, and then click **Save**.

## Removing an app

After an app has been embedded onto a page, there are two ways to remove it if needed:

- Go to the **Edit an app** pane using the instructions from the [Editing an embedded app](#) section earlier in this topic. Confirm that the pane displays information for the embedded app that you would like to remove, and then click the **Delete** button.
- Because the embedded app is saved as personalization data, clearing your page's personalization will also remove any embedded apps on that page. Note that clearing the page's personalization is permanent and cannot be undone. To remove your personalizations on a page, select **Options**, and then **Personalize this page**, and finally the **Clear** button. After refreshing your browser, all the previous personalizations for this page will be removed. See [Personalize the user experience](#) for more information about how to optimize pages using personalization.

## Appendix

### [Developer] Specifying where an app can be embedded

By default, users can embed apps on any page, either under the Power Apps menu button or directly on the page as a tab, FastTab, blade or as a new section in a workspace. However, if required, developers can also configure this feature to only allow embedding of apps on certain pages by implementing the following methods:

- **isPowerAppPersonalizationEnabled** – If this method returns false for a specific page, then the Power

Apps menu button will not be shown, and users will not be able to embed apps anywhere on this page, including as a tab.

- **isPowerAppTabPersonalizationEnabled** – If this method returns false for a specific page, then users will not be able to embed apps directly on the page as a tab, FastTab, or panorama section. Users will still be able to embed apps through the Power Apps menu button if embedding is allowed on the page.

The following example shows a new class with the two methods needed to configure where apps can be embedded.

```
[ExtensionOf(classStr(FormRunConfigurationPowerAppsConfiguration))]  
  
public final class ClassTest_Extension  
{  
    public static boolean isPowerAppPersonalizationEnabled(str pageName)  
    {  
        var result = next isPowerAppPersonalizationEnabled(pageName);  
        return result;  
    }  
  
    public static boolean isPowerAppTabPersonalizationEnabled(str pageName)  
    {  
        var result = next isPowerAppTabPersonalizationEnabled(pageName);  
        return result;  
    }  
}
```

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Find information by using lookups

2/18/2021 • 4 minutes to read • [Edit Online](#)

Many fields have lookups that can help you easily find the correct or desired value. Several enhancements have been added to lookups that make these controls more usable and make users more productive. In this topic, you will learn about these new lookup features and will receive some helpful tips to get the optimal use out of lookups in the system.

## Responsive lookups

In previous versions, when interacting with a lookup control, a user would have to take an explicit action to open the drop-down menu. This may have been by typing an asterisk (\*) in the control to filter the lookup based on the current value of the control, clicking the drop-down button, or by using the **Alt+Down arrow** keyboard shortcut. Lookup controls have been modified in the following ways to better align with current web practices:

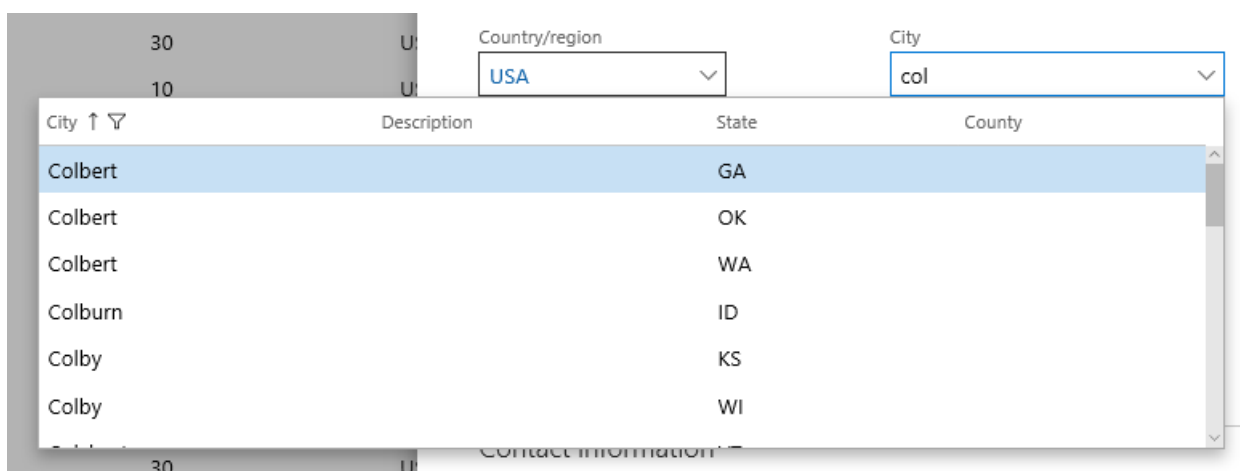
- Lookup drop-down menus will now open automatically after a slight pause in typing, with the drop-down menu contents filtered based on the lookup control's value.

Note that the old behavior of automatic opening of the dropdown after typing an asterisk (\*) has been deprecated.

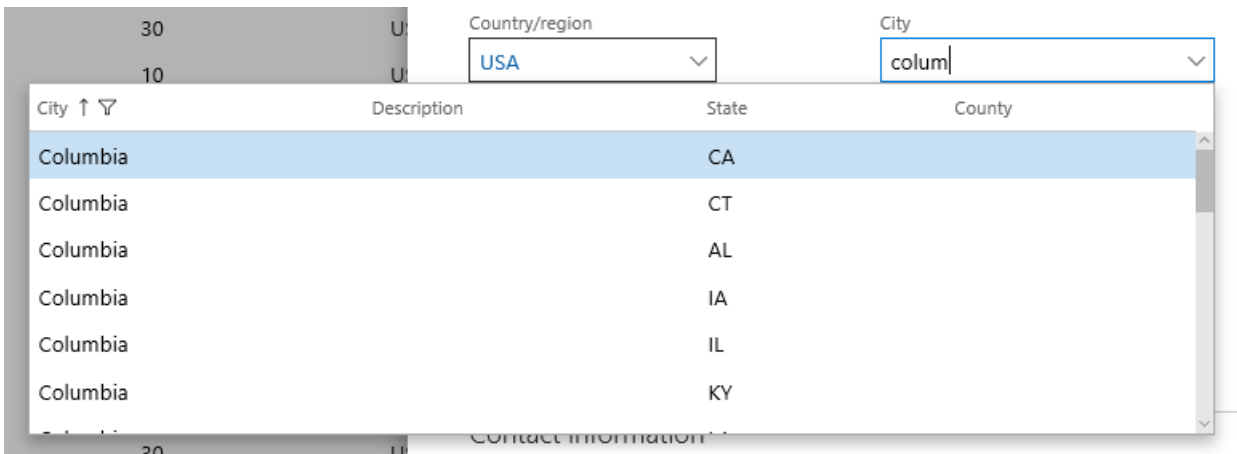
- After the lookup drop-down menu has opened, the following will occur:
  - The cursor will stay in the lookup control (instead of focus moving into the drop-down menu) so you can continue to make modifications to the control's value. However, the user can still use the **Up arrow** and **Down arrow** to change rows in the drop-down menu, and enter to select the current row in the drop-down menu.
  - The contents of the drop-down menu will adjust after any modifications are made to the lookup control's value.

For example, consider a lookup field called **City**.

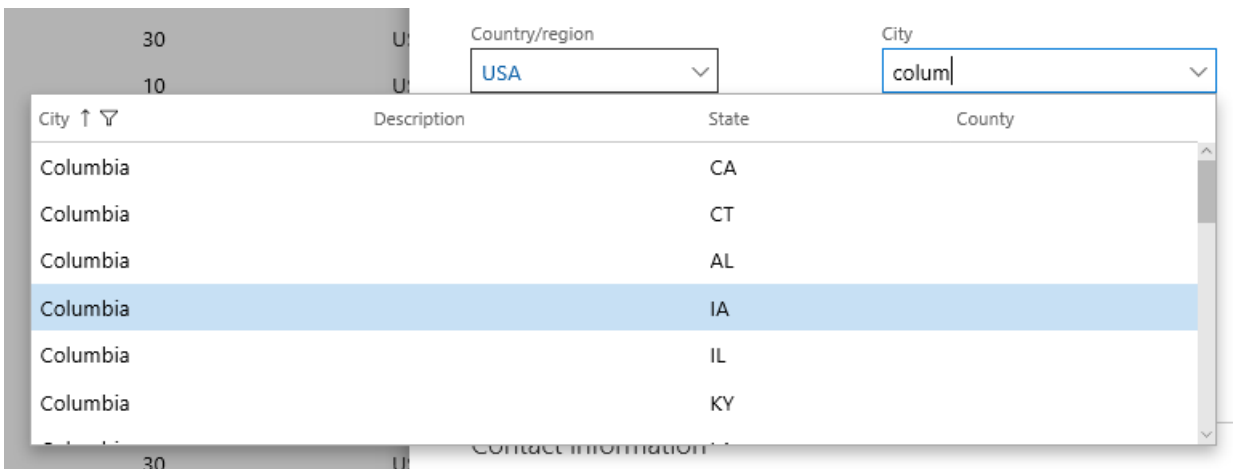
When focus is in the **City** field, you can start looking for the city that you want by typing a few letters, like "col." After you stop typing, the lookup will open automatically, filtered to those cities that begin with "col".



At this point, the cursor is still in the lookup field. If you continue typing so the value is "colum," the lookup contents adjust automatically to reflect the latest value in the control.



Even though focus is still in the lookup control, you can also use the **Up arrow** or **Down arrow** keys to highlight the row that you want to select. If you press **Enter** the highlighted row will be selected from the lookup and the control's value will be updated.

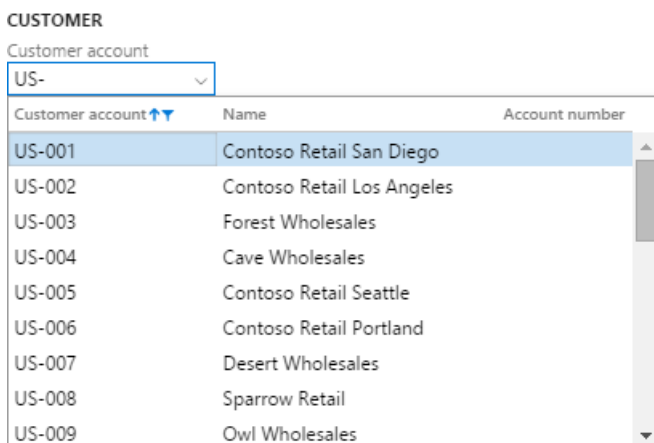


## Typing in more than IDs

When entering data, it's natural for users to attempt to identify an entity, such as a customer or vendor, in terms of the name rather than an identifier representing the entity. Many (but not all) lookups now allow contextual data entry. This powerful feature allows the user to type the ID or the corresponding name into the lookup control.

For example, consider the **Customer account** field when creating a sales order. This field shows the **Account ID** for the customer, but a user would typically prefer to enter an **Account name** instead of an **Account ID** for this field when creating a sales order, such as "Forest Wholesales" instead of "US-003."

If the user started to enter an **Account ID** into the lookup control, the drop-down menu would automatically open as described in the previous section and the user would see the lookup as shown below.



However, the user can also now enter the beginning of an **Account name** as well. If this is detected, then the user will see the following lookup. Notice how the **Name** column is moved to be the first column in the lookup, and how the lookup is sorted and filtered based on the **Name** column.

**CUSTOMER**

Customer account

Conto

Name ↑	Customer account	Account number
Contoso Europe	DE-001	
Contoso Retail Chicago	US-015	
Contoso Retail Dallas	US-011	
Contoso Retail Detroit	US-018	
Contoso Retail Los Angeles	US-002	
Contoso Retail Miami	US-028	
Contoso Retail New York	US-012	
Contoso Retail Portland	US-006	
Contoso Retail San Diego	US-001	

## Using grid column headers for more advanced filtering and sorting

The lookup enhancements discussed in the previous two sections greatly improve a user's ability to navigate the rows in a lookup based on a "begins with" search of the ID or **Name** field in the lookup. However, there are situations in which more advanced filtering (or sorting) is needed to find the correct row. In these situations, the user needs to use the filtering and sorting options in the grid column headers inside the lookup. For example, consider an employee entering a sales order line who needs to locate the right "cable" as the product. Typing "cable" into the **Item number** control isn't helpful, as there are no product names that begin with "cable."

Sales order lines

+ Add line + Add lines Add products Remove Sales order line Financials Inventory Product and supply Update line Warehouse

✓ T...	Variant number	Item number	Product name	Sales category	CW quantity	CW unit	Quan
✓		▼ cable		▼			

Search name Item number ↑ Product name Item group Item type

We didn't find anything to show here.

Line details

GENERAL SETUP ADDRESS

ORDER LINE Product name EXTERNAL REFERENCES INTERCOMPANY Preve

Instead, the user needs to clear the value of the lookup control, open the lookup drop-down menu, and filter the drop-down menu using the grid column header, as shown below. A mouse (or touch) user can simply click (or touch) any column header to access the filtering and sorting options for that column. For a keyboard user, the user simply needs to press **Alt+Down arrow** a second time to move focus into the drop-down menu, after which the user can tab to the correct column, and then press **Ctrl+G** to open the grid column header drop-down menu.

Sales order lines

+ Add line + Add lines Add products Remove Sales order line Financials Inventory Product and supply Update line Warehouse

✓ T...	Variant number	Item number	Product name	Sales category	CW quantity	CW unit	Quant
✓							

Item number ↑	Search name	Product name	Item group	Item type
1000	1001		consume	Item
4401	Proseware 50W Car Ra			Item
4402	Northwind Traders 50			Item
4403	A. Datum 50W Car Rad		dio	Item
A0001	HDMI 6' Cables		&Video	Item
A0002	HDMI 12' Cables		&Video	Item

Line details

GENERAL SETUP ADDRESS

ORDER LINE Product name EXTERNAL REFERENCES INTERCOMPANY

After the filter has been applied (see the image below), the user can find and select the row as usual.

Sales order lines

+ Add line + Add lines Add products Remove Sales order line Financials Inventory Product and supply Update line Warehouse

✓ T...	Variant number	Item number	Product name	Sales category	CW quantity	CW unit	Quant
✓							

Item number ↑	Search name	Product name	Item group	Item type
A0001	HDMI 6' Cables	HDMI 6' Cables	TV&Video	Item
A0002	HDMI 12' Cables	HDMI 12' Cables	TV&Video	Item
M0023	Banana Plugs 24K	Speaker Cable Banana Plugs 24...	AudioRM	Item
M0024	Speaker Cable In-wal	Speaker Cable In-wall 50 Ft	AudioRM	Item
T0001	SpeakerCable	SpeakerCable	Audio	Item

Line details

GENERAL SETUP ADDRESS

ORDER LINE Product name EXTERNAL REFERENCES INTERCOMPANY

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Change the date for a session

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to change the date for a session. By default, the current date is used when entering and posting journal entries or source documents. You can change the date that is used for your current session. Use this feature to back-date journal entries or source documents, as necessary.

1. In the navigation pane, go to **Modules > Common > Common > Session date and time**.
2. In the **Date** field, enter a date.
3. Select **OK**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Set a user's preferred time zone

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following topic explains how a user in the System Administrator role can set the time zone for a user.

1. Go to **Navigation pane > Modules > System administration > Users > Users**.
2. In the list, find and select the desired record.
3. Select **User options**.
4. Select the **Preferences** tab.
5. In the **Time zone** field, select an option from the drop-down list.
6. Select **Save**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Lifecycle Services (LCS) for Finance and Operations apps customers

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic is intended for customers who have signed up for the current versions of Finance and Operations apps. Partners who are working with customers to help them move through the lifecycle of their Lifecycle Services (LCS) project will also find this information useful.

## LCS workspace for the current versions of the Finance and Operations apps

When you sign up for the current versions of Finance and Operations apps, your subscription includes an Implementation project workspace. After you activate the service, the tenant administrator must sign in at <https://lcs.dynamics.com> by using the tenant account. The project workspace is automatically created for your organization. The workspace includes the following elements:

- Enabled features, based on the offer that you selected
- Environments that are deployed and managed by Microsoft
- Guidance that is provided through the Action center to help you complete required actions
- A new methodology experience that includes tasks that lock as you move through the implementation
- A more complete history that specifies who completed each methodology phase and task
- Milestones that you can use to track critical project dates
- Various services to help you with your implementation

## Methodologies

As a customer, you must complete the steps that are outlined in the methodology to gain access to the production environment. Before a phase can be marked as completed, you must complete the specified mandatory tasks. Locked tasks, such as tasks 1.6 and 1.9 in the following screenshot, are unlocked after you've completed the required actions. To learn which actions must be completed before a specific task can be unlocked, click the lock icon for that task.

## Dynamics Demo Project

### ACTION CENTER



Subscription estimate is not complete.  
The number of users estimated in the active subscription estimate does not match the number of enterprise seats purchased.

Subscription estimator

### METHODOLOGY



#### Phase history

Complete phase

- 1.1 Complete LCS project configuration \*
- 1.2 Invite your project team
- 1.3 Deploy demo environment
- 1.4 Publish Plan and Milestone Dates \*
- 1.5 Capture Business processes and requirements \*
- 1.6 Perform Fit/Gap analysis \*
- 1.9 Sign off requirements and business processes \*
- 1.10 Upload first iteration of setup and configurati...

This task cannot be completed until estimator  
1.5 is completed.

#### Task history

Action	User	Date
Reopened	administrator Dyna...	5/2/2016 2:51 PM
Closed	administrator Dyna...	5/2/2016 2:50 PM
Reopened	administrator Dyna...	5/2/2016 2:47 PM
Closed	TestCustomer1@Dy...	3/15/2016 6:35 AM
Reopened	TestCustomer1@Dy...	3/15/2016 6:35 AM
Closed	Shefy Manayil Kare...	2/18/2016 7:12 PM
Created	Shefy Manayil Kare...	2/18/2016 7:10 PM

#### Description

After you have captured the business processes and requirements, a thorough analysis of the each requirement is required to assess if it can be met by the standard product.

In the case of prerequisites, after you complete the required tasks, you can mark the dependent tasks as completed. For example, in the following screenshot, tasks 1.6 and 1.9 depend on task 1.5. Because task 1.5 has now been completed, the two dependent tasks can be marked as completed.

### METHODOLOGY



#### Phase history

Complete phase

- 1.1 Complete LCS project configuration \*
- 1.2 Invite your project team
- 1.3 Deploy demo environment
- 1.4 Publish Plan and Milestone Dates \*
- 1.5 Capture Business processes and requirements \*
- 1.6 Perform Fit/Gap analysis \*
- 1.7 Complete subscription estimator \*
- 1.8 Download templates
- 1.9 Sign off requirements and business processes \*
- 1.10 Upload first iteration of setup and configurati...

#### Task history

Action	User	Date
Closed	Kuntal Mehta	3/29/2016 2:56 PM
Reopened	Raji Ramesh	3/23/2016 10:06 AM
Closed	TestCustomer1@Dy...	3/15/2016 6:35 AM
Reopened	TestCustomer1@Dy...	3/15/2016 6:30 AM
Closed	Shefy Manayil Kare...	2/18/2016 7:11 PM
Created	Shefy Manayil Kare...	2/18/2016 7:10 PM

#### Description

Before kicking off, complete the required configuration for LCS to ensure the best experience. This includes to key areas, Sharepoint and Visual Studio Team Services.

Visual Studio Team Services :

## Milestones

High-level milestones must be defined for a project. Milestones can help you track the deliverables that must be completed and your progress toward the milestone goals. Color indicators help you quickly learn whether you're behind schedule. For example, in the following screenshot, the milestones are yellow. To enter or update the milestone dates, click the diamond shape in the methodology, and then click the Edit button (pencil icon). You can change milestone dates at any time.

# Dynamics Demo Project

**ACTION CENTER**



Subscription estimate is not complete.  
The number of users estimated in the active subscription estimate does not match the number of enterprise seats purchased.

Subscription estimator



Phase history

Complete phase

✓	1.1	Complete LCS project configuration	*
✓	1.2	Invite your project team	
✓	1.3	Deploy demo environment	
✓	1.4	Publish Plan and Milestone Dates	*
○	1.5	Capture Business processes and requirements	*
🔒	1.6	Perform Fit/Gap analysis	*
✓	1.7	Complete subscription estimator	*
✓	1.8	Download templates	
🔒	1.9	Sign off requirements and business processes	*
✓	1.10	Upload first iteration of setup and configurati...	

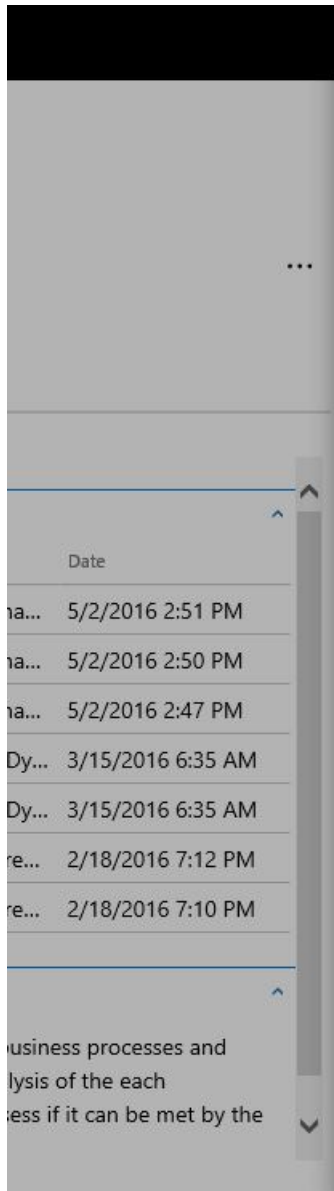
Task history

Action	User	Date
Reopened	administrator Dyna...	5/2/2016 2:51 PM
Closed	administrator Dyna...	5/2/2016 2:50 PM
Reopened	administrator Dyna...	5/2/2016 2:47 PM
Closed	TestCustomer1@Dy...	3/15/2016 6:35 AM
Reopened	TestCustomer1@Dy...	3/15/2016 6:35 AM
Closed	Shefy Manayil Kare...	2/18/2016 7:12 PM
Created	Shefy Manayil Kare...	2/18/2016 7:10 PM

Description

After you have captured the business processes and requirements, a thorough analysis of the each requirement is required to assess if it can be met by the standard product.

When you've finished entering milestones, the **Publish plan and milestone** task opens, and you can mark it as completed.



## Set up milestones

Select the date on which each milestone must be completed. These dates will enable and disable capabilities in Lifecycle Services.

Milestone	End date
Analysis	1/6/2016
Design	2/9/2016
Test	3/17/2016

Save Cancel

When you've completed all the required tasks in a phase, you can click **Complete phase** to mark the phase as completed. After you mark a phase as completed, next steps become available in Microsoft Dynamics Lifecycle Services (LCS).

**METHODOLOGY**

Phase history

**Complete phase**

- ✓ 1.1 Complete LCS project configuration \*
- ✓ 1.2 Invite your project team
- ✓ 1.3 Deploy demo environment
- ✓ 1.4 Publish Plan and Milestone Dates \*
- ✓ 1.5 Capture Business processes and requirements \*
- ✓ 1.6 Perform Fit/Gap analysis \*
- ✓ 1.7 Complete subscription estimator \*
- ✓ 1.8 Download templates
- ✓ 1.9 Sign off requirements and business processes \*
- ✓ 1.10 Upload first iteration of setup and configurati...

**Task history**

Action	User	Date
Closed	Kuntal Mehta	3/29/2016 2:56 PM
Reopened	Raji Ramesh	3/23/2016 10:06 AM
Closed	TestCustomer1@Dy...	3/15/2016 6:35 AM
Reopened	TestCustomer1@Dy...	3/15/2016 6:30 AM
Closed	Shefy Manayil Kare...	2/18/2016 7:11 PM
Created	Shefy Manayil Kare...	2/18/2016 7:10 PM

**Description**

Before kicking off, complete the required configuration for LCS to ensure the best experience. This includes to key areas, Sharepoint and Visual Studio Team Services.

Visual Studio Team Services :

### Methodology description and history

Descriptions can help you understand what is expected of you for a specific methodology task or phase. You can expand the methodology description to learn more about each task, and then collapse the description when you've finished. The task and phase history can tell you when a task or phase was completed or reopened. If you're a project manager, this information can help you stay on top of the high-level tasks that are required for your implementations.

## METHODOLOGY

Phase history

Complete phase

- 1.1 Complete LCS project configuration \*
- 1.2 Invite your project team
- 1.3 Deploy demo environment
- 1.4 Publish Plan and Milestone Dates \*
- 1.5 Capture Business processes and requirements \*
- 1.6 Perform Fit/Gap analysis \*
- 1.7 Complete subscription estimator \*
- 1.8 Download templates
- 1.9 Sign off requirements and business processes \*
- 1.10 Upload first iteration of setup and configurati...

Task history

Action	User	Date
Reopened	administrator Dyna...	5/2/2016 2:51 PM
Closed	administrator Dyna...	5/2/2016 2:50 PM
Reopened	administrator Dyna...	5/2/2016 2:47 PM
Closed	TestCustomer1@Dy...	3/15/2016 6:35 AM
Reopened	TestCustomer1@Dy...	3/15/2016 6:35 AM
Closed	Shefy Manayil Kare...	2/18/2016 7:12 PM
Created	Shefy Manayil Kare...	2/18/2016 7:10 PM

Description

After you have captured the business processes and requirements, a thorough analysis of the each requirement is required to assess if it can be met by the standard product.

## Subscription estimator

You can use the Subscription estimator tool to evaluate your subscription requirements for the current versions of the Finance and Operations apps. To use Subscription estimator, download the usage profile, which is a Microsoft Excel workbook. Then, in the workbook, complete the following worksheets:

- Deployment details
- Instance Characteristics
- Retail & Commerce

After you've completed the worksheets, enter the data from the summary sheet into Subscription estimator by clicking + **New estimate**. You must make one estimate the active estimate. Make sure that the estimate that you mark as active is same as the offer that you bought through the VL or CSP channel.

## New deployment experience

To provision your environment, you must to complete a configuration checklist. As you make progress through the methodology, environments become available to you. Click **Configure** to add deployment information.

The screenshot displays a project management interface. At the top, a sprint timeline shows Sprints 2, 3, and 4, with dates 1/11/2016 and 2/21/2016. Below the timeline is a 'Task history' section with a 'Description' of project planning sessions. A 'Task history' dropdown menu is also visible. On the right side, there is a list of environments under the heading 'ENVIRONMENTS'. The environments listed are:

- PRODUCTION:** Environment Prod-1 is deployed (status: green checkmark). A 'Full details' link is present.
- SANDBOX: PREMIER PERFORMANCE TEST (ADD-ON):** Configuration button.
- SANDBOX: STANDARD PERFORMANCE TEST (ADD-ON):** Configuration button.
- SANDBOX: PREMIER ACCEPTANCE TEST (ADD-ON):** Configuration button.
- SANDBOX: PREMIER ACCEPTANCE TEST (ADD-ON):** Configuration button.
- SANDBOX: STANDARD ACCEPTANCE TEST:** Configuration button.
- SANDBOX: STANDARD ACCEPTANCE TEST (ADD-ON):** Configuration button.
- SANDBOX: DEVELOP AND TEST:** Environment kranthi is deploying... (status: green checkmark). A 'Full details' link is present.
- SANDBOX: DEVELOP AND TEST (ADD-ON):** Configuration button.

Because the information that you enter determines your experience, carefully review your input. After you've entered all the required information, sign-off is required for the deployment request. The user who completes the sign-off becomes the system administrator on the instance. Verify that the correct user completes the sign-off for the deployment. After the sign-off is completed, the Microsoft site reliability engagement team reviews the request. After the team has reviewed the information that you entered, it initiates the provisioning. If the information isn't correct, the team will contact you. After the provisioning is completed, the status is updated to indicate that the environment has been deployed, as shown in the following screenshot. If the provisioning takes longer than expected, the Microsoft site reliability engagement team reviews the status and takes appropriate actions. These actions might include contacting you. After the environment is provisioned, click **Full details** to open the **Detailed environment** page, where you can sign in to the system, view the monitoring status, or view relevant updates.

**SANDBOX: DEVELOP AND TEST**

✓ Environment DynDemoDevTest is deployed

[Full details](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Implementation lifecycle management home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

These topics describe the programs, tools, and processes available related to the implementation lifecycle of your Finance and Operations project.

## Programs

[Microsoft FastTrack](#)

## Tools

[Microsoft Dynamics Lifecycle Services](#)

[Lifecycle Services \(LCS\) for Finance and Operations apps customers](#)

## Processes

[Onboard an implementation project](#)

[Prepare for go-live](#)

## Frequently asked questions

[Go-live for implementation projects FAQ](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

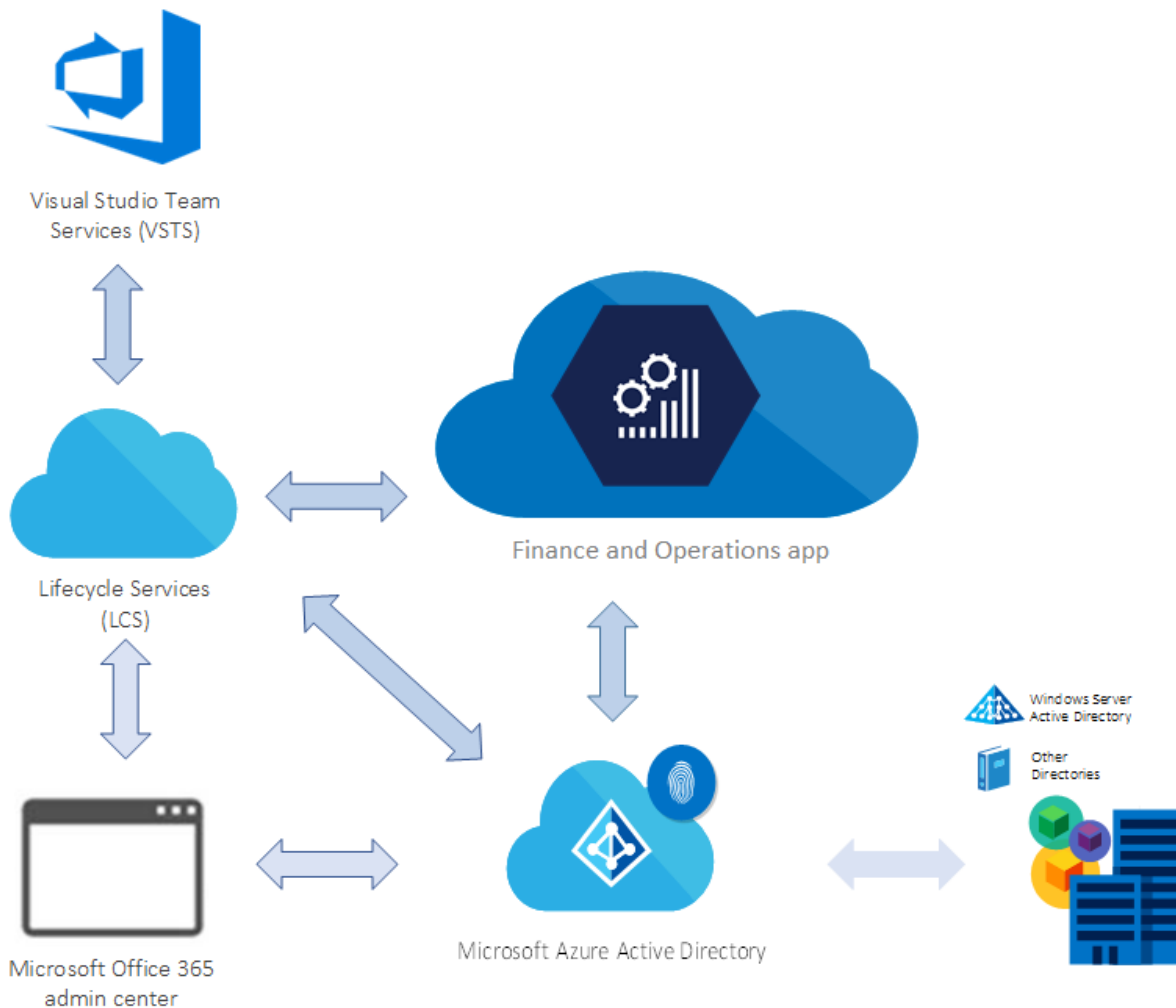
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Finance and Operations application architecture

2/18/2021 • 4 minutes to read • [Edit Online](#)

The Finance and Operations application cloud architecture contains all the elements that are common to all Microsoft cloud offerings, as described in [Subscriptions, licenses, accounts, and tenants for Microsoft's cloud offerings](#). Beyond this, it also includes services that automate software deployment and provisioning, operational monitoring and reporting, and seamless application lifecycle management.



The cloud architecture consists of these conceptual areas:

- **Subscription** – A subscription to Finance and Operations apps gives you an online cloud environment (or multiple environments) and experience.
- **Licenses** – Customers must purchase subscription licenses (SLs) for their organization, or for their affiliates' employees and on-site agents, vendors, or contractors who directly or indirectly access Finance and Operations apps. These apps are licensed through Microsoft Volume Licensing and the Microsoft Cloud Solution Provider (CSP) program. For more information, download the latest [Microsoft Dynamics 365 Licensing Guide from Dynamics 365 pricing](#).
- **Tenant** – In Microsoft Azure Active Directory (AAD), a tenant represents an organization. It's a dedicated instance of the AAD service that an organization receives and owns when it creates a relationship with Microsoft (for example, by signing up for a Microsoft cloud service, such as Azure, Microsoft Intune, or Microsoft 365). Every AAD tenant is distinct and separate from other AAD tenants. For more information

about AAD tenants, see [How to get an Azure Active Directory Tenant](#).

A tenant houses the company's user information. This information includes passwords, user profile data, permissions, and related information. The tenant also contains groups, applications, and other information that pertains to an organization and its security.

The tenant is created when customers sign up for their first subscription to any Microsoft online service, such as Microsoft 365, Microsoft Dynamics 365, or Azure. Any later subscriptions to the same online services or other online services can be grouped within the same tenant.

An organization can have multiple AAD tenants. If there are multiple tenants, make sure that any subscriptions for Finance and Operations apps are associated with the correct tenant.

- **Azure Active Directory (AAD)** – AAD is the multi-tenant, cloud-based directory and identity management service from Microsoft that combines core directory services, application access management, and identity protection in a single solution. For more information, see [Azure Active Directory](#). Finance and Operations apps use AAD as the store for identity. Access to AAD is provided as part of a subscription to Finance and Operations apps.
- **Microsoft 365 admin center** – Microsoft 365 admin center is the subscription management portal that Microsoft 365 provides for administrators. It's used to provide management functions for users (AAD) and subscriptions. As part of these management functions, it provides information about service health. For more information, see [About the Microsoft 365 admin center](#).

#### NOTE

You don't have to have an Microsoft 365 license to deploy Finance and Operations apps. However, you might require a license for specific Office integration scenarios. For more information, see [Office integration overview](#).

- **Microsoft Dynamics Lifecycle Services (LCS)** – LCS is a collaboration portal that provides an environment and a set of regularly updated services that can help you manage the application lifecycle of your implementations. For more information, see [Lifecycle Services resources](#). After you purchase and activate a subscription for a Finance and Operations app, an **Implementation project** workspace is provisioned in LCS when the tenant administrator signs in for the first time.

#### NOTE

An implementation project is an LCS project for the cloud service. As a Microsoft partner, you can also provision non-implementation LCS projects for your own purposes. For more information, see [Lifecycle Services \(LCS\) for Finance and Operations apps partners](#).

- **Finance and Operations apps** – Finance and Operations apps are deployed through LCS. Various topologies are available: development/test/build, acceptance test, performance test, and high-availability production. For more information about the various topologies, download the [latest Microsoft Dynamics 365 Licensing Guide from Dynamics 365 pricing](#).
- **Microsoft Azure DevOps** – Azure DevOps is used primarily for code version control, development, and to deploy a build environment. Azure DevOps is also used to track support incidents, such as work items in Azure DevOps that are submitted to Microsoft through Cloud-powered support, and to integrate the Business process modeler (BPM) library hierarchy into your Azure DevOps project as a hierarchy of work items. Azure DevOps is also used during code upgrade.

"Under the hood," Finance and Operations apps use many features of the Azure platform, such as Azure Storage, networking, monitoring, and Azure SQL Database, to name just a few. Shared services put into operation and orchestrate the application lifecycle of the environments for participants. Together, Azure functionality and LCS

offer a robust cloud service.

**NOTE**

Although many features of the Azure platform are used, you don't have to have an Azure subscription to deploy Finance and Operations apps in the Microsoft-managed cloud. You must have an Azure subscription only if you want to deploy Finance and Operations apps cloud-hosted environments in your own Azure subscription.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Onboard an implementation project

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to onboard a Finance and Operations project by using Microsoft Dynamics Lifecycle Services (LCS).

## Microsoft 365 Admin Center

After your organization has purchased a subscription to Finance and Operations, it must be activated on your organization's Azure Active Directory (Azure AD) tenant by your Tenant Administrator, who completes the following steps:

1. Open an InPrivate/Incognito browser session and go to the [Microsoft 365 Admin Center](#).
2. Sign in with the Tenant Administrator credentials.
3. Go to **Billing > Products & services** and confirm that there is an active subscription for the application that you want to deploy.

### NOTE

If you do not see an active subscription, consult with your Licensing Partner to confirm the status of the subscription transaction as well as the tenant for the subscription. By default, all Microsoft Online Services should be running on the same Azure AD tenant.

4. If the subscription in question is shown as active, proceed to the next step by signing in to LCS to trigger the Implementation Project creation flow.
5. Open another private browser tab and go to [Lifecycle Services](#). Select **Login** to access LCS with your current Tenant Admin credentials.
6. Accept and confirm any other prompts displayed to complete the Implementation Project provisioning.
7. The Tenant Administrator is assigned the Project Owner security role in the provisioned Implementation Project.

### NOTE

If the Tenant Administrator will not be a participant in the implementation, at least one additional Project Owner must be assigned to the implementation project.

For an overview of LCS user management, including the security roles that can be assigned to users, see [Configure Lifecycle Services \(LCS\) security](#).

## LCS implementation project workspace

After the Tenant Administrator has completed the Finance and Operations subscription activation and added additional project owners as appropriate, those team members can access the **Implementation project** workspace.

The first step to be completed in LCS is **Project onboarding**. This step is required for all LCS implementation projects that are created **on or after August 22, 2019, PST**, prior to deploying any of the Microsoft-managed

environments. You can access the **Project onboarding** feature using the action center notification or the LCS Implementation project menu. You must be assigned to the Project owner security role to access **Project onboarding** in LCS.

To get started with LCS, see [Lifecycle Services \(LCS\) for Finance and Operations apps customers](#).

## FastTrack onboarding services

After the LCS **Implementation project** workspace is provisioned, the Microsoft FastTrack team will monitor your onboarding progress. If project onboarding is not completed within a few weeks after creating an LCS **Implementation project**, a reminder will be sent to the project team.

For more information about the FastTrack program and the services provided, see [Microsoft FastTrack](#).

For more information about LCS project onboarding, see [LCS project onboarding](#).

### NOTE

All onboarding-related emails from the FastTrack team will originate from Dynamics 365 Onboarding ([ond365@microsoft.com](mailto:ond365@microsoft.com)), so please ensure that any spam blocker/filter allows email from this address.

## Key data to keep current in LCS

We recommend that you add key project members (such as project managers) from the customer and partner teams to the LCS implementation project. Be sure to include each person's work email address. In this way, you help us work best with you and help ensure that project members don't miss important communication from us.

Be sure to keep the milestone dates in your LCS project current. In this way, you help us connect with you at different project stages. When you're closer to your go-live date, we will contact you for a project Go-live assessment before we deploy your production environment.

Milestone dates are stored in the LCS implementation methodology. For more information, see the [Methodologies](#) section of the "LCS for Customers" topic.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Environment planning

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic provides an overview of various aspects that you must consider while you plan for your project's environment. To help guarantee a successful cloud implementation, it's important that you discuss and plan your environment early in the project.

## Environment planning overview

To begin, here are a few important concepts:

- **Environment purpose** – The reasons why the environment exists. Examples include development, system testing, user acceptance testing (UAT), and operations.
- **Environment topology** – The composition of the environment and the purpose. Examples include **Develop** and **Build and Test** for Tier-1 environments.
- **Environment tier** – The type or category of the environment. Examples include Tier-1 environments and Tier-2 environments.

For more information, about the various environments and tiers, see [Cloud deployment overview](#) and download the latest *Microsoft Dynamics 365 Licensing Guide* from [Dynamics 365 pricing](#).

### Environment types

You can use the following environment types for your project:

- **Standard** – This environment is included in the standard offer and is managed by Microsoft in a Microsoft subscription. Standard environments include the production environment and a Tier-2 Standard Acceptance Test environment.
- **Add-on** – The add-on environments are in a Microsoft-managed subscription that the customer has purchased in addition to the standard offer. For example, an add-on environment might be an additional Tier-4 environment for performance testing.
- **Cloud-hosted** – Cloud-hosted environments are additional environments that are managed by the customer or partner in a customer or partner Microsoft Azure subscription. A cloud-hosted environment can include a Tier-1 demo environment.
- **Environment image (VHD)** – These additional Tier-1 environments are hosted on-premises by using a virtual hard disk (VHD) that can be downloaded from [Microsoft Dynamics Lifecycle Services \(LCS\)](#).

#### IMPORTANT

In a *customer or partner Azure subscription*, the customer or partner brings its own Azure subscription, and deploys environments to that subscription for evaluation and development purposes only. The customer or partner pays for the resources that are deployed to its Azure subscription. The amount that the customer or partner pays is based on the Azure price list. By contrast, in a *Microsoft subscription*, the customer purchases licenses that allow the customer to deploy environments to an Azure subscription that is managed by Microsoft. Therefore, the customer has no separate Azure billing.

### Tier-1 vs. Tier-2 and higher

TIER-1	TIER-2 AND HIGHER
Single-box environment	Multi-box environment

TIER-1	TIER-2 AND HIGHER
All components are installed on the same server. These components include Application Object Server (AOS), the database, Dynamics 365 Commerce, and Management Reporter.	Components are installed on multiple servers.
Microsoft SQL Server is used.	<a href="#">Azure SQL Database</a> is used.
The architecture differs from the architecture of the production environment to maximize efficiency and cost of the development team.	The architecture is the same as the architecture of the production environment, even though this type of environment has a different sizing and isn't enabled for disaster recovery.
The environment can be cloud-hosted, or it can be deployed as an environment image (VHD).	The environment can be deployed only as a standard environment or an add-on environment. It can't be cloud-hosted.
The environment isn't suitable for UAT or performance testing.	The environment is suitable for UAT and performance testing.

## Standard cloud offer

The standard cloud offer includes two environments:

- **Tier-2 environment: Standard Acceptance Testing** – One Standard Acceptance Testing (UAT) instance is provided for the duration of the subscription. This instance is a non-production multi-box instance that customers can use for UAT, integration testing, and training. Additional sandbox/staging instances can be purchased separately as an optional add-on.
- **Production environment** – One production instance is provided per tenant. The production multi-box instance includes disaster recovery and high availability. It will be provisioned when the implementation approaches the Operate phase, after the required activities in the Microsoft Dynamics Lifecycle Services (LCS) methodology and a successful go-live assessment are completed. Additionally, some file storage and database storage are included in the offer:
  - **File storage:** Every customer receives a certain amount of file/Azure blob cloud storage for files and binary data. Additional file/blob storage can be purchased.
  - **Database storage:** Every subscription includes a certain amount of Azure SQL Database storage per customer at no additional charge. Additional storage capacity is provided at no charge as an organization increases the number of user and device service licenses. For more information about the various environments and the various types of storage, as well as the currently included free file and storage capacity, download the latest *Microsoft Dynamics 365 Licensing Guide* from [Dynamics 365 pricing](#).

### IMPORTANT

Microsoft promises service and data high availability as well as minimal servicing downtime guarantees as part of the Dynamics 365 software license agreement (SLA) for production environments. The SLA goals do not apply to non-production environments.

### Provisioning of standard environments

The standard environments are provisioned at different times. The following table shows the suggested timing for the environments in the standard cloud offer.

ENVIRONMENT	WHEN DOES PROVISIONING OCCUR?	IS IT SELF-SERVICE?
Tier-2 Standard Acceptance Test	Immediately after project onboarding has been completed in LCS	Yes
Production	At production system readiness	A Go-live assessment must be completed prior to configuring the production deployment request in LCS.

### IMPORTANT

Always deploy environments by using an **unnamed** account. This account must be from the customer domain, such as `dynadmin@customer.com` or `dynadmin@customer.onmicrosoft.com`. We strongly recommend using the same dedicated environment admin account on all environments. ISV licenses must be issued to the domain of the account used for environment deployment, therefore using both `dynadmin@customer.com` and `dynadmin@customer.onmicrosoft.com` on different environments can lead to problems with ISV licenses.

### Production system readiness

The production environment can be deployed when the project is ready for the initial go-live. For more information, see [Prepare for go-live](#).

Production system readiness includes, but isn't limited to, the following conditions:

- An up-to-date subscription estimate is activated, as described in [Subscription estimator in Lifecycle Services \(LCS\)](#).
- Code, configuration, and data are ready for cutover.
- An engineering process is in place to manage critical fixes.
- The customer has signed off on the solution and UAT.
- A cutover plan is in place.

Customers should use the production environment to **operate** the solution, not build it. The production environment is sized to run your business. The sizing is based on the subscription estimate and diagnostic data from performance testing. After deployment, customers can and should do a mock cutover and a final round of validation on the production environment. Before the final cutover, customers can request a Point in time restore to restore the production environment to a clean snapshot (maximum 28 days in the past).

To select the appropriate data center for the production environment, consider the latency from the geographic locations where the business operates. Use tools such as [PsPing](#) and [Azure Speed Test](#) to test latency to Azure data centers.

The following illustrations shows the environment planning process.



## Additional environments

Additional environments can be purchased as add-ons, or they can be deployed as cloud-hosted environments. The following illustration shows a *sample* overview of standard and additional environments, based on the complexity of the implementation.



Environment purpose	Tier	Standard project	Medium complexity project	Complex project
Build	1	X	X	X
Test (e.g. system integration testing or UAT)	2	X	X	X
Production	Based on the <b>sizing</b>	X	X	X
Development	1	X	X	X
Golden configuration	1 or 2		X	X
Pre-production / data migration	2		X	X
Performance testing	4 or 5			X
Training	2 or 3			X
System integration testing	2 or 3			X
Other / ad-hoc	...			X

### IMPORTANT

Always deploy environments by using an **unnamed** account, such as `dynadmin@customer.com`. Assign the environments an owner who will be responsible for their status and maintenance. We strongly recommend using the same dedicated environment admin account on all environments. After go-live, if you plan to work on new releases, get an additional Tier-2 or higher environment to support production.

### Deployment considerations for development environments

For development environments, there are two deployment options:

- **Cloud-hosted** – The environments are managed by the customer/partner in a customer/partner Azure subscription.
- **Environment image (downloadable VHD)** – The environments are hosted on-premises.

### NOTE

You must allocate one development environment per developer.

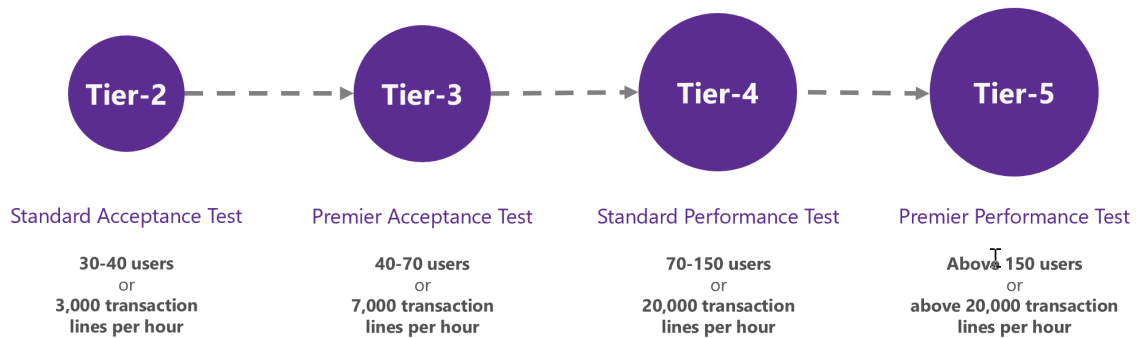
The following table compares the deployment options.

CAPABILITY	CLOUD-HOSTED	ENVIRONMENT IMAGE
Public URL	✓	Not supported
Integration development	✓	Extra setup is required. (For example, run the admin user provisioning tool.)
Azure DevOps	✓	Extra setup is required. (For example, rename the computer.)
Applying deployable packages from LCS	Automated	Command line runbooks (AxUpdateInstaller.exe tool)
Deploying data packages from LCS	✓	Not supported
Maintenance	Managed by the customer/partner	Managed by the customer/partner

CAPABILITY	CLOUD-HOSTED	ENVIRONMENT IMAGE
Cost model	Pay as you go (If the environment is on for eight hours, you pay for eight hours.). Cost is based on selected Virtual Machine size, disk size and settings, and premium storage settings	Hardware-related
Limitations	None. You have full control over VM specs, disk size and storage settings. You have administrator access to the VM.	None

### Selecting the correct Tier-2 or higher environment

It's important that you select the correct Tier-2 or higher environment, depending on the purpose of the environment. The guidance that is provided in the following illustration is a *baseline*. You must work with your implementation partner to adjust this guidance, based on your specific business scenarios and factors such as type of users, complexity, and volumes.



After a subscription estimate is activated, you can view transaction lines per hour in LCS, as shown in the following illustration.

Subscription estimator

What is subscription estimator?  
Subscription estimator provides an automated estimate of the subscription needed for your production instance. It uses the subscription licenses and transaction details to infer your subscription needs. While you can have multiple estimates, you need to mark one estimate as 'Active'.

The active estimate is currently locked because there is a production environment that is active or has been signed off for deployment. To mark a different estimate as active, please clear the signoff or contact support.

[+ New estimate](#) [Delete estimate](#) [Mark active](#) [Subscription\(s\)](#) [Sample usage profile](#)

Name	Estimated by	Created on
80	SA Solutions Architect	7/14/2018 1:25 AM

80

Specifications

The subscription license(s) are recorded at the time of an estimate's creation. If the subscription licenses have changed recently, please create a new estimate to refresh this data.

[Usage profile](#)

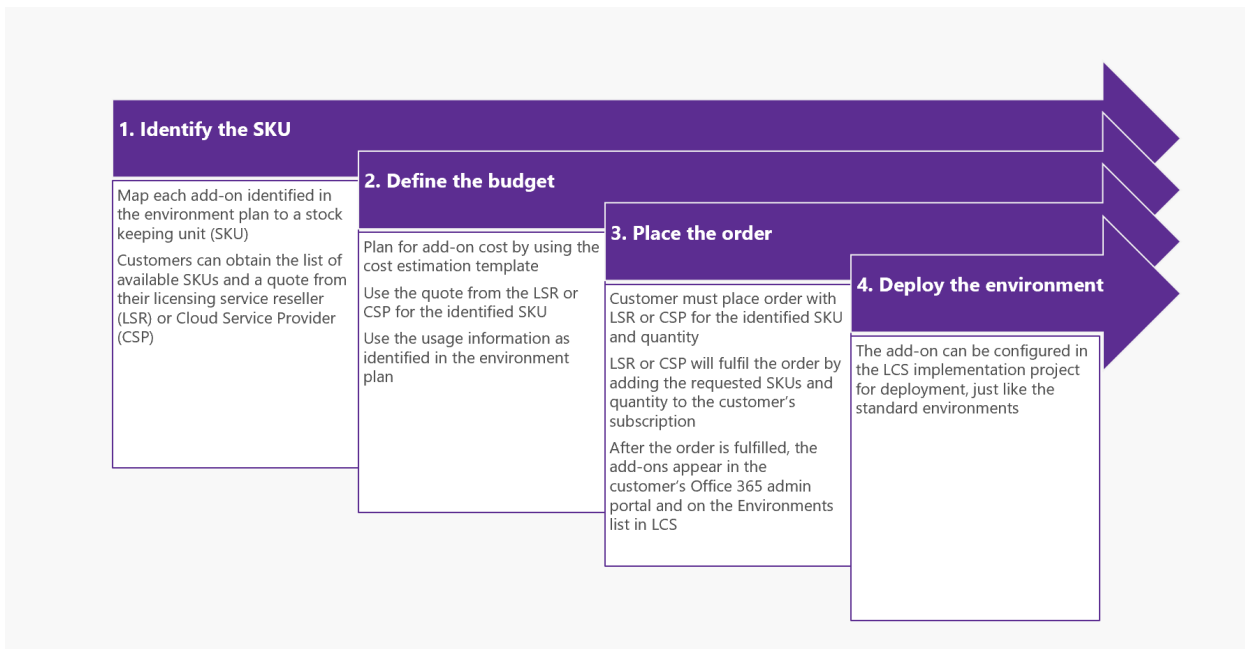
Operations user count: 32

Transaction lines per hour (Non-POS): 7327

### Purchasing add-on environments

If you want to purchase add-on environments, we recommend that you work closely with your Cloud Solution Provider or License Service Reseller. Consider the potential lead time that occurs between the time when the order is placed and the time when the environment is deployed.

The following illustration shows the process for purchasing add-on environments.



### IMPORTANT

If you have a Microsoft Volume Licensing agreement, you can subscribe to add-on environments on a monthly basis through the Microsoft Products and Services Agreement (MPSA) licensing program. Alternatively, you can subscribe to them through the Microsoft Cloud Solution Provider (CSP) program. For more information about the various environments and tiers, download the latest *Microsoft Dynamics 365 Licensing Guide* from [Dynamics 365 pricing](#).

## Environments plan

Create the environments plan early in your implementation.

1. Identify the project activities that require an environment. These activities include, but aren't limited to, development of customizations and maintenance of golden configuration data.
2. Determine the *activities lifecycle* to determine the *environments lifecycle*. Here are some examples of the questions that you should ask during this step:
  - When and for how long do you require the environment?
  - Do you require the environment before or after go-live?
3. Determine the type and topology of the required environments.
4. Summarize the list of required environments in a matrix.

After you've identified the environments, the environments plan can be used to structure the Application Lifecycle Management (ALM) flows. For example, after you finalize your environments plan, you can define the flows for building and moving the code and the data across environments.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

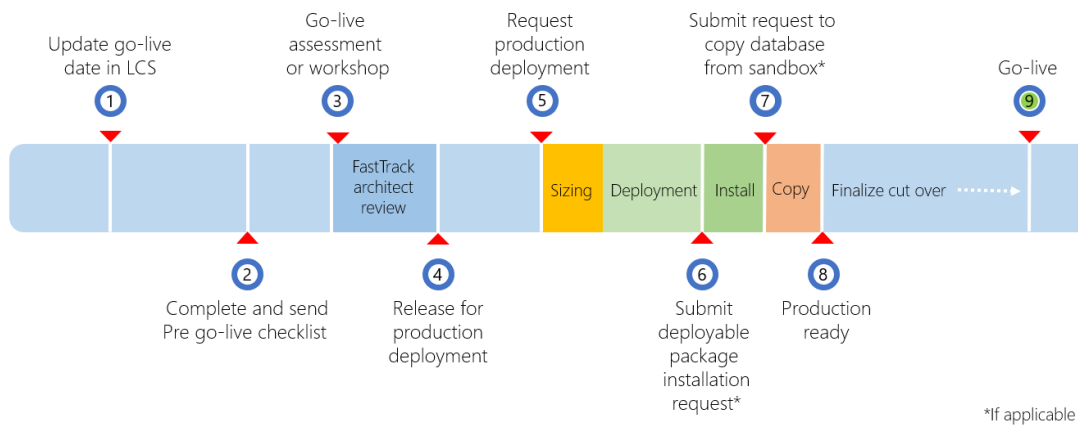
# Prepare for go-live

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic describes how to prepare to go live with a project by using Microsoft Dynamics Lifecycle Services (LCS).

Production and Sandbox can only be deployed in two different [types of environments](#): Microsoft Managed or Self-Service. Both follow the same preparation for go-live, but the service level agreements (SLA) and some of the process steps are different.

This graphic and the following table list the phases of the go-live process, the environment type to which each phase applies with the expected duration, and who is responsible to take the action.



PHASE	ACTION	ENVIRONMENT TYPE	DURATION/WHE N	WHO	NOTES
1	Update Go-live date in LCS	Both	At the latest 2-3 months in advance	Customer/Partner	The milestone dates should be kept up to date on an ongoing basis.
2	Complete and send pre go-live checklist	Both	After user acceptance testing (UAT) complete	Customer/Partner	Follow the instructions provided in the "FastTrack Go-live Assessment" section later in this topic.

PHASE	ACTION	ENVIRONMENT TYPE	DURATION/WHEE N	WHO	NOTES
3	Project Go-Live assessment (FastTrack Essentials)	Both	3-business days for initial report, plus additional time for mitigation, if required	Microsoft-FastTrack Solution Architect	Solution Architect delivers assessment after checklist is received and continues review until questions are clarified and mitigations are in place, if applicable.
	Go-live Assessment workshop (FastTrack)	Both	To coordinate with architect assigned	Microsoft-FastTrack Solution Architect	
4	Release for production deployment	Microsoft Managed	Immediate upon successfully completed assessment	Microsoft-FastTrack Solution Architect	Do not submit production request until the assessment is successfully completed.
		Self-Service	Immediate upon successfully completed assessment and <b>Configure</b> button is enabled	Microsoft-FastTrack Solution Architect	For Self-Service deployment, the <b>Configure</b> button remains disabled until the assessment is complete.
5	Production deployment request	Microsoft Managed	Self-service	Customer/Partner	The production deployment request should only be submitted after the FastTrack Architect has finished the assessment.
		Self-Service	Self-service	Customer/Partner	After the assessment is complete, the <b>Configure</b> button will be enabled and customer will be able to request the production deployment.

PHASE	ACTION	ENVIRONMENT TYPE	DURATION/WHE N	WHO	NOTES
	Sizing	Both	Immediate in case of automatic sizing. Could require further clarifications of the subscription estimate.	Microsoft-Dynamic Service Engineering (DSE)	Automatic sizing based on subscription estimate by default, manual sizing by exception.
	Deployment	Microsoft Managed	48 hours	Microsoft-Dynamic Service Engineering (DSE)	Status in LCS reflects the deployment progress. If there are any questions about your request, they will be posted as Comments on the service request.
		Self-Service	An average of 30 minutes	Microsoft-FastTrack Solution Architect	The deployment could take an average of 30 minutes after the assessment has completed and the production environment has been requested. For more information, see <a href="#">Deploy a new environment</a> .
6	Deployable package installation request	Both	Self-service	Customer/Partner	Follow the instructions in <a href="#">Apply updates</a> . The packages must contain all the models and binaries consolidated in an <a href="#">All-in-one</a> deployable package.

PHASE	ACTION	ENVIRONMENT TYPE	DURATION/WHE N	WHO	NOTES
	Package installation	Both	Minimum 5 hours lead time and 4 hours downtime	Microsoft-Dynamic Service Engineering (DSE)	Generally, 95% of updates are applied in less than one hour, however we still recommend that you provide a downtime window of four hours in case a rollback is required for any reason. When the package deployment succeeds, the environment will be available as soon as the package deployment has finished, which means that the longer downtime window does not have any negative effect on the availability of the system.
7	Database copy from Sandbox request (if applicable)	Both	Self-service	Customer/Partner	Follow the instructions <a href="#">Self-service database refresh</a> . If you have a golden configuration you can review <a href="#">Golden configuration promotion</a> .
	Copy database	Both	Five hours lead time and four hours downtime	Microsoft-Dynamic Service Engineering (DSE)	Generally, the database copy is completed in less than one hour. We still recommend that you provide a downtime window of four hours in case a rollback is required for any reason.

PHASE	ACTION	ENVIRONMENT TYPE	DURATION/WHE N	WHO	NOTES
8	Production ready	Both	After all previous steps have been completed	Customer/Partner	Customer can take control of the production environment.
	Cutover activities	Both	Depends on the project	Customer/Partner	
9	Go live	Both	Depends on the project	Customer/Partner	

## Completing the LCS methodology

A major milestone in each implementation project is the cutover to the production environment.

To ensure that the production environment is used for live operations, Microsoft will provision the production instance only when the implementation is approaching the **Operate** phase, after the required activities in the LCS methodology are complete. For more information about the environments in your subscription, see the [Licensing guide](#).

Customers must complete the **Analysis, Design and Develop**, and **Test** phases in the LCS methodology before the **Configure** button that is used to request the production environment becomes available.

### NOTE

For Self-Service environments, the **Configure** button will only become available after the Solution Architect has signed off on the assessment.

To complete a phase in LCS, you must first complete every required step in that phase. When all the steps in a phase are completed, you can complete the whole phase. You can always reopen a phase later if you must make changes. If you require more help, see [Lifecycle Services \(LCS\) for Finance and Operations apps customers](#).

The process of completing a step has two parts:

- Do the actual work, such as a fit-gap analysis or user acceptance testing (UAT).
- Mark the corresponding step in the LCS methodology as completed.

It's good practice to complete the steps in the methodology as you make progress with the implementation. Don't wait until the last minute. Don't just click through all the steps so that you can get a production environment. It's in the customer's best interest to have a solid implementation.

## UAT completion and solution sign off

During the UAT phase, you must test all the business processes that you've implemented, and any customizations that you've made, in a Sandbox, or Standard Acceptance Test, environment in the implementation project. To help ensure a successful go-live, you should consider the following as you complete the UAT phase:

- Test cases cover the entire scope of requirements.
- Test by using migrated data. This data should include master data and opening balances, even if they aren't yet final.
- Test by using the correct security roles (default roles and custom roles) that are assigned to users.



- Make sure that the solution complies with any company-specific and industry-specific regulatory requirements.
- Run the [Customization Analysis Report \(CAR\)](#) and resolve critical issues.
- Complete performance testing.
- Document all features, and obtain approval and sign-off from the customer.

Regardless of whether the environment is a cloud-hosted environment or a downloaded virtual hard disk (VHD), testing can't be considered complete when you test only in an environment that is a developer or demo topology. Here are the reasons:

- The topology of the Tier-1 environments differs from the topology of your production environment. It's important that you test all functionality on a Tier-2 or higher sandbox environment in the Microsoft-managed subscription. It's especially important that you test integrations, printing functionality, workflow functionality, and warehouse and commerce devices in the sandbox environment.
- System performance can't be measured when you do the UAT on local virtual machines (VMs) or VMs that are privately hosted.
- To prevent delays during the cutover process, it's important that the team experience the servicing in LCS during the implementation. This servicing includes the processes of applying deployable packages, creating service requests, and moving database between environments.

## FastTrack Go-live assessment

All customers must complete a go-live review with the Microsoft FastTrack team before their production environment can be deployed. This assessment should be successfully completed before you request your production environment. If you aren't familiar with Microsoft FastTrack, see [Microsoft FastTrack](#).

About eight weeks before go-live, the FastTrack team will ask you to fill in a go-live checklist.

You can download the checklist from [Dynamics 365 Community](#) on the [Go-live Planning TechTalk](#) page.

The project manager or a key project member must complete the go-live checklist during the pre-go-live phase of the project. Typically, the checklist is completed four to six weeks before the proposed go-live date, when UAT is completed or almost completed.

When you've completed the go-live checklist, email it to **Dynamics 365 FO Go-Live** [d365fogl@microsoft.com](mailto:d365fogl@microsoft.com). Always include a key stakeholder from the customer and the implementation partner on the email.

After the checklist is submitted, a Microsoft solution architect will review the project and provide an assessment that describes the potential risks, best practices, and recommendations for a successful go-live of the project. In some cases, the solution architect might highlight risk factors and ask for a mitigation plan. When the assessment is completed, the solution architect will indicate that you're ready to request the production environment in LCS.

For Microsoft Managed environments, if you request the production environment before the assessment is completed, the deployment will remain in the **Queued** state until the assessment is successfully completed. For Self-Service environments, the **Configure** button to request production will be only enabled after the assessment is completed.

You can cancel an environment deployment request while it is in a **Queued** state by following these steps:

1. Select **Queued**.
2. On the **Customer sign-off** tab, select **Clear sign-off**.

This will set the environment back into a state of **Configure** and allow you to make changes to the configuration, such as selecting a different data center or environment topology.

# Requesting the production environment

## NOTE

The production environment is used exclusively for running your business operations and shouldn't be used for testing. You will be able to perform the cutover, and if planned, to mock the cutover in production. To test the solution, you must use a UAT environment, which is designed with the necessary elements and services for testing.

After you've completed the analysis, design and develop, and test phases in the LCS methodology, and the go-live assessment has concluded that the project is ready, you can request your production environment.

We recommend that you select a service account, for example a generic user account, as the Admin user of the environments that you deploy. If you use a named user account, you might not be able to access an environment if that user isn't available. Here are some scenarios where the Admin user must access an environment:

- **First sign-in to any environment after initial deployment** – In this case, the Admin user is the only user who can access the environment.
- **First sign-in to a sandbox environment after a database refresh from the production environment** – In this case, all user accounts except the Admin account are unable to sign in.

Your production environment should be deployed to the same datacenter where your sandbox environments are deployed.

After you've signed off on the request for the production environment, Microsoft is responsible for deploying the production environment for you. For **Microsoft Managed** environments, the Microsoft service level agreement (SLA) for deployment of a production environment is 48 hours. The production environment can be deployed at any time within 48 hours after you submit the request, provided that your usage profile doesn't require additional information. For **Self-Service** environments, the deployment will take around 30 minutes after the production request has been submitted. You can view the progress of the deployment in LCS. Typically, the status of the production environment request remains **Queued** for a few hours before it's changed to **Deploying**.

When you submit the deployment request, a service request for the Microsoft Dynamics Service Engineering (DSE) team is automatically created. You can view this service request in the **Service requests** list in LCS. If the DSE team has questions that prevent them from deploying the production environment, they will add a comment to the service request. For example, the DSE team might ask that you update the subscription estimate or change the datacenter. In some cases, you might have to clear the sign-off from the production deployment request to make changes.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Go-live for implementation projects FAQ

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic lists frequently asked questions about how to go live with an implementation project.

## When can I configure and request my production environment?

Typically, a production environment is deployed after all customizations are code-complete, user acceptance testing (UAT) is completed, the customer has signed off on the solution, and there are no blocking issues for go-live.

When you're at this stage, the Microsoft FastTrack team will work with the project team to do a Go-live assessment/review.

## What are the prerequisites to deploy a production environment?

For a list of the prerequisites, see [Prepare for go-live](#).

## What is a Go-live assessment/review, and why is it required?

The Go-live assessment/review is part of the [Microsoft FastTrack program](#). During this review, a solution architect assesses whether an implementation project is ready for a successful cutover and go-live. This review is mandatory for every implementation project before you can request to go live in a production environment.

## I want to request my production environment. Who do I contact for a Go-live assessment/review?

If a FastTrack solution architect is assigned to your project, contact him or her directly. Otherwise, based on the go-live date that is specified in Microsoft Dynamics Lifecycle Services (LCS), you will receive an email that instructs you to fill out the Pre-go-live checklist and send it to [d365fogl@microsoft.com](mailto:d365fogl@microsoft.com) a few weeks before the go-live date. If you haven't received an email, and you're ready for go-live, you can download the checklist from [Dynamics 365 Community](#) on the [Go-live Planning TechTalk](#) page, complete it, and send it to [d365fogl@microsoft.com](mailto:d365fogl@microsoft.com).

## The Production button isn't available in LCS. How do I request my production environment?

The **Production** button in LCS is available only after you've completed the **Analysis, Design & develop**, and **Test** phases of the LCS implementation methodology. For more information about how to complete these phases, see [Lifecycle Services \(LCS\) for Finance and Operations apps customers](#).

### NOTE

Your production environment won't be deployed until the Go-live assessment/review has been completed.

My sandbox environment is currently on an update that is set to expire in two months. Can I request a production environment that has the latest update?

No. We will deny any request for a production environment that is on a different version than your sandbox environment. When you configure a production environment, the versions that you select must match the versions of the sandbox environment where you signed off on your solution. Therefore, you must first apply the latest update to your sandbox environment, test it, and sign off.

For more information, see [Software lifecycle policy and cloud releases](#).

## Our sandbox environments are deployed in the Central US datacenter, but we want our production environments to be deployed in the West US datacenter. Can I select West US as the datacenter in my production configuration?

No. We will deny any request for a production environment that is in a different datacenter than your sandbox environment. We require that all your environments reside in the same datacenter. If you want your production environment to reside in the West US datacenter, you must first redeploy your sandbox environments to the West US datacenter, test them, and sign off.

For information that can help you select the correct datacenter, see the [Network requirements](#) section of the "System requirements" topic.

## How will my production environment be sized?

Your production environment will be sized based on the current user license count and the information in the subscription estimate that is active when you request the production environment.

### NOTE

If you add additional users later, you must create a support ticket to activate a new subscription estimate. Your production environment might have to be resized, depending on the number of users, the type of user licenses, and the expected peak transaction volume. Downtime is required in order to resize a production environment.

## I submitted the request for a production environment, but I made a mistake. Can I still change it?

Yes. As long as the status of the production environment is **Queued**, you can clear the sign-off flag, make changes, and then sign off again.

## How long does it take to deploy my production environment?

After the Go-live assessment with the Microsoft FastTrack team is completed and the production request is submitted, deployment of the production environment should be completed within 48 hours.

## What level of access do I have in my production environment? Can I sign in to the VM?

No. Access to the production environment is limited. You can't access the virtual machine (VM) or Microsoft Internet Information Services (IIS). You also can't access the database through Microsoft SQL Server Management Studio.

## How often is my production database backed up?

Databases are protected by automatic backups. Full database backups are done weekly, differential database

backups are done hourly, and transaction log backups are done every five minutes. Automatic backups are retained for 35 days.

For more information, see [Learn about automatic SQL Database backups](#).

## Can I request a copy of the backup of my production database?

No. However, you can submit a database refresh service request to copy your production database to your Tier 2 and higher sandbox environment. After the copy request is completed, you can back up your sandbox environment.

## My golden configuration database is in a Tier 1 sandbox environment. How can I copy and restore it to my production environment?

To copy and restore your database, follow the instructions in the topic, [Golden configuration promotion](#).

### NOTE

If your golden configuration is in data packages, you must manually import the data packages to the production environment.

## After go-live, can I apply new code changes to the production environment?

Yes. In LCS, you can submit a service request to apply a deployable package to your production environment. Application of one deployable package to a production environment involves a lead time of five hours and downtime of approximately five hours.

For more information, see [Apply updates to cloud environments](#).

## What should I do if my production environment is down?

To report a production outage, follow the process described in the topic, [Report a production outage](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Submit service requests to the Dynamics Service Engineering team

2/18/2021 • 7 minutes to read • [Edit Online](#)

A service request is a ticket that you use to request that the Dynamics Service Engineering (DSE) team perform a predefined set of tasks on your environments.

## NOTE

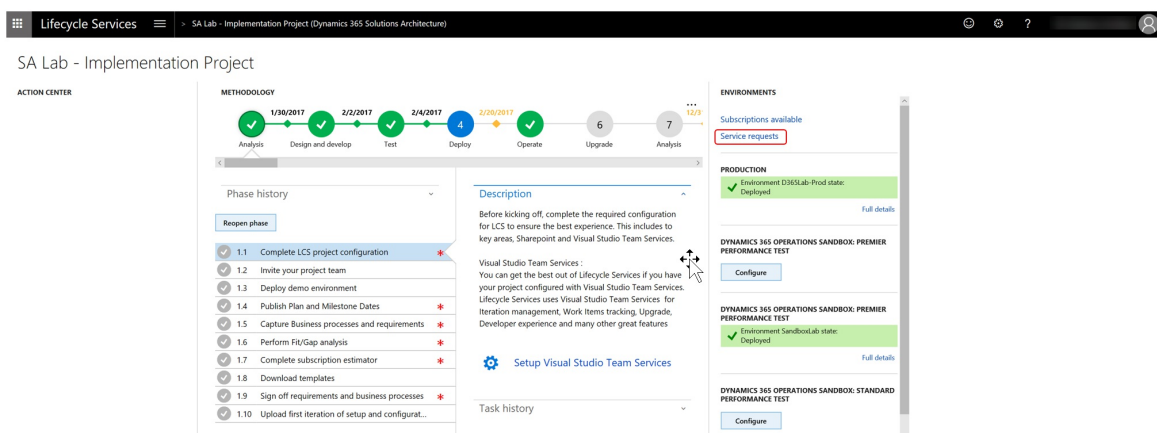
Don't use service requests for product issues. If you encounter a situation that doesn't fit into any of the tasks that are described in this topic, submit a support ticket instead. For more information about support tickets, see [Get support for Finance and Operations apps](#) or [Lifecycle Services \(LCS\)](#).

You can use Microsoft Dynamics Lifecycle Services (LCS) to submit service requests directly to the DSE team. You can also view which requests have been submitted, executed, and canceled for your environments.

## View service requests

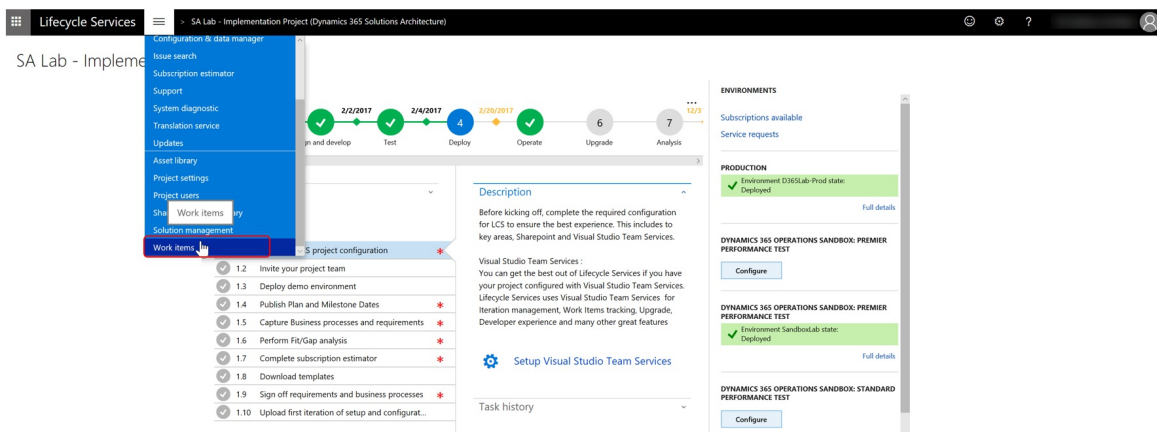
There are two ways to view service requests:

- On the project dashboard, in the **Environments** section, select **Service requests**.



The screenshot shows the Lifecycle Services interface for a project named 'SA Lab - Implementation Project (Dynamics 365 Solutions Architecture)'. The 'ENVIRONMENTS' section is visible on the right, with a 'Service requests' link highlighted in red. The main area shows a methodology timeline with steps: Analysis (1/30/2017), Design and develop (2/2/2017), Test (2/4/2017), Deploy (2/20/2017), Operate (4), Upgrade (6), and Analysis (7/12/23). Below the timeline is a 'Phase history' list with tasks like 'Complete LCS project configuration', 'Invite your project team', etc. A 'Description' box provides details about the configuration process.

- Select the **Menu** button and select **Work items**. On the **Work items** page select the **Service requests** tab.



This screenshot shows the same project dashboard as above, but with the 'Work items' menu open on the left. The 'Work items' option is highlighted in red. The 'ENVIRONMENTS' section on the right is still visible, showing the 'Service requests' link. The main area displays the methodology timeline and phase history.

By default, the **Service requests** tab on the **Work items** page lists all requests that are currently active and requests that have been denied. However, you can use the filter options to show canceled and finished requests too.

Work items

Open work items + Add View environment details Reschedule Cancel

Support issues Filter

Service requests

ID	Service request type	Environment name	Service request status	Actionable by	Downtime start date	Downtime end date	Created by	Modified by	Created at	Modified at
34679	New deployment	D36580PU15Lab	Requested	Microsoft	4/3/2018 4:37:41 AM (UTC +02:00)		SA Solutions Architect	SA Solutions Architect	4/3/2018 4:37 AM	4/4/2018 1:59 AM
27563	Other	D36572UAT	Request denied	Customer / Partner	4/1/2018 5:30:00 AM (UTC +02:00)	4/2/2018 8:00:00 AM (UTC +02:00)	SA Solutions Architect	Microsoft SRE Tier1 gr...	2/1/2018 12:16 AM	4/1/2018 5:45 AM
24209	Upgrade environment	D36572PUAT	Request denied	Microsoft	3/20/2018 2:00:00 AM (UTC +02:00)	3/21/2018 2:00:00 AM (UTC +02:00)	SA Solutions Architect	SA Solutions Architect	1/3/2018 1:33 PM	3/2/2018 9:12 PM

After you submit a request, it has a status of **Requested**. Before the DSE team acts on the request, it might ask for clarification by entering a comment in the **Comment** field. For example, you might receive a comment from the DSE team if you request deployment of a production environment, but the data center differs from the data center where your sandbox environments are deployed. Carefully review the comments, and provide any required clarification in your own comment. To view the details of a specific request, or to submit comments for a service request, select the request ID.

If you signed up for LCS notifications, you receive an email when the status of a service request changes or a comment is entered.

If you submit a service request to the DSE team, and the action is outside the team's scope, the service request will be denied. In this case, the reason for the denial and suggestions for further action are provided. For some typical examples of service requests that the DSE team will deny, see the "Denied service requests" section later in this topic.

## Create service requests

There are two ways to create a service request: automatically and on demand.


- **Automatically** – A service request is automatically created when you request deployment of an environment, or an application of a package.
- **On demand** – A service request is manually created when you enter a request for a database point-in-time restore, and some other services.

### Automatically create a service request

- **Environment deployment** – To set up deployment options and submit a request to the DSE team to deploy a new environment, in the **Environments** section, select **Configure**.
- **Package application** – To apply a package to the production environment, on the **Environment details** page, select **Maintain**, select the package to apply, and then select **Schedule**. For more information, see [Apply updates to cloud environments](#).

## IMPORTANT

If your scheduled time overlaps with a [planned maintenance window](#), you will receive the following warning message.

 **There is a high likelihood that maintenance activity may be scheduled for this environment during this time. Overlapping environment operations with maintenance activity will cause issues and possibly cause extended downtime. Would you like to proceed with this operation?**

[Show diagnostic information](#)

Yes

No

If you choose to continue deploying the package, the package deployment operation will be rolled-back in the event of conflict, as planned maintenance takes priority.

This restriction is applicable to **Microsoft-managed IAAS environments** only.

## Create a service request on demand

Service requests that are created on demand aren't explicitly accepted by the DSE team. They will be addressed during the specified downtime window unless the DSE team has entered a comment in the request or has had to deny the request. For details, review the comments in the service request.

Microsoft frequently reviews all incoming service requests. By selecting the correct type of service request for your scenario, you help the DSE team handle the request in a timely manner.

1. On the **Work items** page, on the **Service requests** tab, select **Add**.
2. In the **Create request** dialog box, select the type of service request to create. The options on the page then reflect the specific type of request that you selected.
  - **Sandbox point-in-time restore request** – Select this request type to restore a *non-production* database to a specific point in time. For more information, see [Database movement operations home page](#).

### NOTE

If you need to restore a *production* database to a previous point-in-time during the cutover phase, select the **Production point-in-time restore request** type. If you need to restore a production database when you're already live in operations, submit a support ticket through LCS.

- **Database refresh request** – Select this request type to refresh a database from a production environment to a sandbox environment, or from one sandbox environment to another. For more information, see [Refresh database](#). *This request type is being retired on January 31, 2019.*

### NOTE

If you need to refresh a database from a sandbox environment to a production environment during the cutover phase, select the **Sandbox to Production** type.

- **Sandbox to Production** - Perform a database refresh of your configuration data to a production



environment during the cutover phase. For more information, see [Database movement operations home page](#).

- **Other request** – You need to use the **Other request** type exactly as described here. If you word a request in a way that isn't clear to the DSE team, the team will enter a comment to ask for clarification, and your request will be delayed. If you use the **Other request** type for any request that isn't listed below, the request will be denied. Select this request type to request that the DSE team perform one of the following actions:
  - Turn on maintenance mode in a production environment. For more information, see [Maintenance mode](#).
  - Tenant move of a live Production environment. Request the Microsoft Service Engineering team to move the Production database and Azure Blob Storage from the old tenant to the new tenant if you are moving tenant on a live Production environment. Make sure that you only request this service when you are ready with all prerequisites. For more details, see [Move LCS implementation projects to different Azure AD tenants](#).
  - Define explicit Internet Protocol (IP) safe list rules in a production environment.

**NOTE**

Support for explicit safe list rules is deprecated for self-service environments. For more information, see [Removed or deprecated platform features](#).

- Request that Microsoft Power BI Embedded be activated in a sandbox environment, Standard Acceptance Test environment, or production environment if you receive the following message: "Power BI embedded isn't enabled. Please contact your system administrator."

### Commonly denied service requests

Here are some typical examples of service requests that will be denied:

- You submit a request of the **Other request** type for one of the following actions, but you should have submitted a support ticket instead:
  - You want to activate a new subscription estimate after you're live in production or after you've requested a production environment.
  - You want to reset the Financial reporting data mart in a release that is earlier than Microsoft Dynamics 365 for Finance and Operations Financial reporting release 7.2.6.0.
  - You want to restore a production database after go-live.
  - You encountered an issue after the DSE team did an application upgrade.
- You submit a request of the **Other request** type for an action that you should have requested through a different request type. Examples include a database refresh in a non-production environment.
- You submit a request of the **Other request** type for an action that you should perform yourself. Examples include a database upgrade in a development environment.

## Service request types and SLAs

SERVICE REQUEST TYPE	APPLICABLE ENVIRONMENTS	REQUESTED SERVICE	LEAD TIME	DOWNTIME
----------------------	-------------------------	-------------------	-----------	----------

SERVICE REQUEST TYPE	APPLICABLE ENVIRONMENTS	REQUESTED SERVICE	LEAD TIME	DOWNTIME
Environment deployment	Any	Environment deployment	Service level agreement (SLA): within two business days	
Package application	Production	Deployable package application	Five hours	Five hours
Sandbox point-in-time restore	Any Tier 2 or higher sandbox	Database point-in-time restore	Five hours	Four hours
Production point-in-time restore	Production	Database point-in-time restore	Based on data volume	Based on data volume
Sandbox to Production	Tier 2 or higher sandbox to Production	Sandbox to Production	Five hours	Four hours
Other	Production	Maintenance mode	Five hours	Not applicable, because the customer indicates in the service request when the environment should be taken out of maintenance mode again
	Production	IP safe list rules	Five hours	Two hours
	Production	Power BI Embedded	Five hours	Two hours

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Subscriptions, LCS projects, and Azure Active Directory tenants FAQ

2/18/2021 • 3 minutes to read • [Edit Online](#)

When customers subscribe through a Microsoft Volume Licensing agreement or a Microsoft Cloud Solution Provider (CSP) agreement, they usually have one Microsoft Azure Active Directory (Azure AD) tenant, one Microsoft Dynamics Lifecycle Services (LCS) Implementation project and any number of sandbox environments that are deployed to one data center of the customer's choice, and one production environment. For more information about these core concepts, see [Finance and Operations application architecture](#). Although this setup works well for most projects, more advanced scenarios are sometimes required, or changes during the implementation lifecycle must be accommodated.

This topic provides answers to frequently asked questions about subscriptions and licenses, Azure AD tenants, and LCS Implementation projects.

For more information, see the following topics:

- [Move environments between data centers](#)
- [Move licenses between agreement types](#)
- [Move LCS implementation projects to different Azure AD tenants](#)
- [Multiple LCS projects and production environments on one Azure AD tenant](#)

## Do I have to move Azure AD tenants when I move from a CSP agreement to a Volume Licensing agreement?

No. You can keep the existing Azure AD tenant, but you must make sure that the Volume Licensing subscriptions are purchased against the same Azure AD tenant as the CSP subscriptions.

## Do I get a new LCS Implementation project when I move from a CSP agreement to a Volume Licensing agreement?

No. The LCS project remains the same.

## Can I keep the existing LCS Implementation project when I move to different Azure AD tenant?

No. A new LCS project will be created.

## How long does it take to move from a CSP agreement to a Volume Licensing agreement?

For a Volume Licensing purchase, it can take a few days for the order to be processed and the subscriptions to be activated. Redeployment of add-on environments has a service level agreement (SLA) of two business days. It takes a few hours to deallocate and delete old add-on environments.

## What if I forget to delete the existing environments before I suspend the existing subscription?

If you don't deallocate and delete the existing environments before you suspend the subscriptions, the environments will remain in a **Deployed** state. However, if you try to access the full details of these environments, you will receive an error message.

## Can I have a CSP agreement and a Volume Licensing agreement in parallel?

Yes. However, you must maintain the minimum required number of licenses under each program.

## How can I find the Tenant name and Tenant ID within LCS?

1. Go to project home page in LCS.
2. In the **Environments** section, select **Subscriptions available**.
3. On the **Subscriptions available** page, you will find the **Tenant name** and the **Tenant ID**.

## How can I find the subscription status?

1. Go to the project home page in LCS.
2. In the **Environments** section, select **Subscriptions available**.
3. On the **Subscriptions available** page, you'll find all **Service plans** available to the tenant.
4. The **Assigned date** indicates the date that service plan status was changed.

## How would the subscription status impact the environment?

Some of the environment's operations may be impacted by the subscription status:

- **Active** - Your subscription is in an operative state. You should be able to perform all environment operations.
- **In Grace Period** - Your subscription has expired, but is within the grace period. You should renew your subscription soon. The subscription status won't impact your license quantity, ability to deploy a new environment, or to perform environment operations.
- **Suspended** - Your subscription has expired beyond the grace period. This subscription status may reduce the license quantity, impact your ability to deploy a new environment, or impact your ability to perform environment operations.
- **Deleted** - Your subscription has been deleted. This will impact your ability to deploy a new environment or perform environment operations.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Move environments between data centers

2/18/2021 • 3 minutes to read • [Edit Online](#)

Occasionally, you must move environments that are managed by Microsoft to a different Microsoft Azure data center. Here are some scenarios where this move might be required:

- The data center that you planned to use wasn't available when the environments were originally deployed.
- The project creators didn't do enough research to determine the best data center before the environments were originally deployed.
- The customer moves the physical location of its operations, and the wide area network (WAN) connection is now closer to a data center that provides lower latency.

Microsoft asks that you keep all your environments in the same data center. When you move environments to a different data center, you should plan to eventually have all environments deployed in the same data center.

You can verify the data center that an environment is deployed to on the **Manage environments** page in Microsoft Dynamics Lifecycle Services (LCS).

To change the data center, you must redeploy all environments. The process differs for sandbox environments (sandbox standard acceptance test environments, and sandbox develop and test environments) and production environments.

## Move sandbox environments

Because this move is a self-service action, the partner and/or customer must move the existing sandbox environments without Microsoft involvement. Although this action requires little effort on the part of the partner or customer resources, completion of the end-to-end process might require a few days. To streamline the data movement between environments, you should develop a plan to determine the best sequence before you begin the move.

### Save data

Before you begin the move, you must save your data.

- **Tier 1 environment database that is based on Microsoft SQL Server:** Make a backup of the database.
- **Tier 2 and higher environments that are based on Azure SQL Database:** Choose one of the following options:
  - **Option 1:** Review the processes that are listed in the [Database movement operations home page](#) topic.
  - **Option 2:** If you have an Azure subscription, save a copy of the Azure SQL database under that subscription.
  - **Option 3:** If you have multiple Azure SQL database environments, redeploy one environment, leave the remaining environments in the old data center, and then request a database refresh between the environments.
  - **Option 4:** Save data as data packages, and then import the packages after the redeployment is completed.

### Move the environments

After you've saved your data, follow these steps.

1. Verify that all code packages have been uploaded to the Asset library in LCS.
2. For each environment, follow these steps:
  - a. In LCS, select **Full details**.
  - b. Stop the environment, and then, when the environment has stopped, select **Deallocate**.
  - c. After the deallocation is completed, select **Delete**.
  - d. After the environment is deleted, select **Configure** to redeploy the environment.
  - e. In the **Geography/location** field, select the data center to use.
  - f. After the environment is deployed, apply the code packages.
  - g. If the redeployed environment is used as the build environment, complete the required configurations that are described in [Deploy and use an environment that supports continuous build and test automation](#).
  - h. Restore the data.

#### NOTE

- The movement of files that are stored in Azure Blob Storage isn't supported in sandbox environments.
- Commerce customers should be aware that extra steps are required for components to work correctly after a move. For more information, see [Data management overview](#).

## Move production environments

If you already have a production environment deployed, you must open a Support request to move the production environment to another data center after you've finished moving all the sandbox environments. This scenario is rare, and there is no automated/self-service action to complete the move. In this scenario, files that are stored in Azure Blob Storage will also be moved. For information about the maintenance window and downtime that are required in order to move a production environment to a different data center, see [Service Description](#) and the related service-level agreement (SLA) documents.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Move licenses between agreement types

2/18/2021 • 3 minutes to read • [Edit Online](#)

Sometimes, a customer who originally purchased subscriptions through a Microsoft Cloud Service Provider (CSP) agreement decides to change to a Microsoft Volume Licensing agreement with Microsoft after the Microsoft Dynamics Lifecycle Services (LCS) Implementation project has been created. The customer can make this change even after the project has gone live in production.

Less often, a customer who originally purchased the subscriptions through a Volume Licensing agreement with Microsoft decides to change to a CSP agreement. In this case, the change must align with the renewal date of the Volume Licensing agreement.

The process of moving subscriptions from one type of agreement to another is primarily a commercial process. The technical implications for the LCS Implementation project are minimal.

## NOTE

The movement of subscriptions between agreement types isn't the same as the movement of an Azure Active Directory (Azure AD) tenant. If the contractual changes in the agreements require that an Azure AD tenant be moved, you must also follow the process that is described in [Move LCS implementation projects to different Azure AD tenants](#).

Subscriptions come with two standard environments: a production environment and a Tier-2 Standard Acceptance Test environment. These environments aren't affected by the movement of subscriptions between agreement types. Action might be required in LCS only if the customer has additional add-on environments. In this case, action that is related to the add-on environments requires minimal effort on the part of partner or customer resources. To streamline the movement of data between environments, you should plan in advance to determine the best sequence.

## The customer has only default environments

If the customer has only the two standard environments that come with the Microsoft-managed subscription, and didn't purchase any add-on environments through the original CSP agreement or Volume Licensing agreement, the activities that are required are purely commercial.

1. The customer places the order for subscriptions under the new agreement with the Volume Licensing reseller or the CSP.

## IMPORTANT

Make sure that the subscriptions are purchased against the same Azure AD tenant that is used on the original agreement.

2. The customer activates the subscriptions.
3. In Microsoft 365 Admin center, the customer verifies that both the new and subscriptions and the existing subscriptions are active.
4. When the new subscriptions are active, the customer requests that the Volume Licensing reseller or the CSP suspend the existing subscriptions. Typically, there is an overlap to help guarantee continuity and avoid disruption of service.

# The customer has add-on environments

If the customer purchased add-on environments through the original CSP agreement or Volume Licensing agreement, those environments should be redeployed.

## Prerequisites

Before you begin the move, you must complete the following tasks:

- Save the data from your existing environments. Follow one of the options described in [Database movement operations home page](#).
- Verify that all code packages have been uploaded to the Shared asset library in LCS.

## Commercial activities

1. The customer places the order for the subscriptions under the new agreement with the Volume Licensing reseller or the CSP. These subscriptions include the subscriptions for the add-on environments.

### IMPORTANT

Make sure that the subscriptions are purchased against the existing Azure AD tenant.

2. The customer activates the subscriptions.
3. In Microsoft 365 Admin center, the customer verifies that both the new subscriptions and the existing subscriptions are active.

## Deploy new environments

1. When the new subscriptions are active, additional add-on environments that you can configure appear in LCS. Deploy the add-on environments, and configure them as appropriate.
2. Apply the deployable packages, and restore the data.

## Delete environments under the obsolete agreement

Follow these steps for every environment that was deployed under the old agreement. After you've deleted the environments, don't use or redeploy them again.

1. In LCS, on the **Environment details** page, select **Full details**.
2. Stop the environment, and when the environment has stopped, select **Deallocate**.
3. When the deallocation is completed, select **Delete**.
4. When the environment has been deleted, select **Configure**.

## Update environments

1. The Volume Licensing reseller or the CSP suspends the existing subscriptions.
2. Any original add-on environments no longer appear in LCS.

### NOTE

Until physical redeployment of the add-on environments is completed, both existing subscriptions and new subscriptions must be kept in an active state.

- The movement of files that are stored in Azure Blob storage isn't supported in sandbox environments.
- Commerce customers should be aware that extra steps are required in order for components to work correctly after the move. For more information, see [Data management overview](#).



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Move LCS implementation projects to different Azure AD tenants

2/18/2021 • 10 minutes to read • [Edit Online](#)

You can move your subscriptions and your Microsoft Dynamics Lifecycle Services (LCS) Implementation project to a different Microsoft Azure Active Directory (Azure AD) tenant. Here are some scenarios where this move might be required:

- Subscriptions were accidentally purchased against the incorrect Azure AD tenant.

## NOTE

If you're a cloud service provider, and you sell subscriptions for Finance and Operations apps to an existing customer, you must request a reseller relationship with that customer to put the subscriptions on the customer's existing Azure AD tenant. If you create a new customer record for the customer in Microsoft Partner Center, you create a new Azure AD tenant for the customer.

- The customer changes the structure of the Azure AD tenant after the subscription is purchased.

The process for moving your subscriptions and all related artifacts has four main steps, as shown in the following illustration.



## Activate subscriptions on the new tenant

Work with your cloud service provider or volume license reseller to activate the subscriptions against the new Azure AD tenant. All subscriptions for users, and for add-on environments, must be activated.

### Cloud service provider

If you're licensed through a Microsoft Cloud Solution Provider (CSP) agreement, purchase the required subscriptions against the new tenant from your cloud service provider. If the new tenant already exists, the cloud service provider must request a reseller relationship. Alternatively, in Partner Center, the cloud service provider must create a new customer that has the desired default domain name, \*.onmicrosoft.com (for example, contoso.onmicrosoft.com).

Ask the cloud service provider not to suspend the existing subscriptions at this time.

### Volume Licensing

If you're licensed through a Microsoft Volume Licensing agreement, you must call the [Volume Licensing support center](#) and ask that the subscriptions be remapped from the old tenant to the new tenant. You can contact Volume Licensing Support through Microsoft 365 Admin center. Request a grace period, when the subscriptions will be active on both tenants. Because of customer privacy concerns, this request must be made by the customer. You should have the following information available:

- Public customer number.
- Enrollment number.
- The current tenant domain that the subscriptions are currently provisioned on.
- The destination tenant domain that the customer wants the subscriptions provisioned under.
- A detailed explanation of why the customer must have its Volume Licensing subscriptions migrated to a different tenant.
- The total number of paid subscriptions that must be moved to the new tenant, together with the subscription type and seat count.

#### IMPORTANT

It's crucial that the subscriptions be active on both tenants in parallel for a few weeks, until you've finished decommissioning LCS on the old tenant.

## Configure LCS on the new tenant

On the new tenant, you will get a new LCS project that you must initiate and set up.

1. Complete the Project Onboarding wizard. For more information, see [LCS project onboarding](#). When completing the wizard, you must indicate on the **Project Overview** page that you are **Moving existing LCS project from another tenant** and provide the source LCS project ID.
2. Fully configure LCS. As part of this configuration, you must:
  - a. Upload and activate a subscription estimator. If you are already live in the source LCS project, you need to ensure that the estimates match.
  - b. Add your deployable package to the asset library.
  - c. Update your Business process modeler (BPM) library.

#### IMPORTANT

During this period, you will have two parallel LCS projects. You can verify the name and ID of the Azure AD tenant that is associated with an LCS project on the [Subscriptions available](#) page in LCS.

## Move your sandbox environments to the new tenant

1. Deploy the non-production environments in the new LCS project.
2. Apply the required code packages to the environments. Make sure that the target is running the same application version as the source. We recommend using [All-in-one deployable packages](#) and include any ISV licenses, if applicable.
3. Upload data to the environments. You can move the data through data packages or by restoring the database. If you restore the database, additional steps are required in order to remap some properties to the new tenant.
4. Update your user information.
  - a. Remove all user accounts except the admin user.
  - b. Fix the admin user record in USERINFO.

```
UPDATE USERINFO
SET SID='mysid', NETWORKALIAS='myalias/email', NETWORKDOMAIN='https://sts.windows.net'
WHERE ID = 'Admin'
...
```

5. Re-import all other users that have the correct security identifier (SID) and identity provider.
6. Run the following commands to update the tenant ID in the appropriate tables. You can verify the Azure AD tenant ID that is associated with an LCS project on the **Subscriptions available** page in LCS.

```
Update POWERBICONFIG set TENANTID = 'newtenantid' where TENANTID = 'oldtenantid'
Update PROVISIONINGMESSAGETABLE set TENANTID = 'newtenantid' where TENANTID = 'oldtenantid'
Update B2BINVITATIONCONFIG set TENANTID = 'newtenantid' where TENANTID = 'oldtenantid'
Update RETAILSHAREDPARAMETERS set TENANTID = 'newtenantid' where TENANTID = 'oldtenantid'
```

7. Fully configure the environments. As part of this step, configure the integration endpoints.
8. Perform smoke tests on the user acceptance testing (UAT) environment in the new LCS project. These tests should focus on user sign-in, integrations, workflows, printing, reporting, and similar processes that depend on configuration and user information.

Depending on your solution and scope, you might have to perform additional steps on the new Azure AD tenant. These steps might include registering applications (for recurring integrations and warehouse management), adding domains, and setting up directory synchronization to enable single sign-on (SSO).

Note that calls to web services are allowed only from the **home** tenant for the environment. For example, the original tenant was `companya.com`, and integration ran as `services\@companya.com`. In this case, when you switch tenants to `companyb.com`, you can no longer use `services\@companya.com` for web service calls, even if you update `userInfo.networkdomain` to `https://sts.windows.net/companyb.com`.

#### IMPORTANT

On your sandbox environments, you will lose any document handling attachments that are stored in Azure Blob storage. Blob storage will be moved by Microsoft only for production environments.

## Move your production environment to the new tenant

If you do not have a production environment deployed already on the old tenant, you can skip this section.

If you already had a production environment deployed on the old tenant, Microsoft will move your database and Azure Blob storage from your old production environment to the new one. As a pre-requisite, you must complete the additional steps below after you've finished moving all the sandbox environments and completed UAT. The process of moving a production environment to a new tenant requires a downtime.

Before requesting the production environment, ensure that all pre-requisites are completed:

1. Get all required licenses that are needed to correctly license all users on the production environment.
2. When the licenses are in place, upload a subscription estimator to the new LCS project. It should match the subscription estimator that is active in the source LCS project, and it must correctly reflect peak transaction volumes.
3. Send an email to Dynamics 365 FO Go-Live ([d365fogl@microsoft.com](mailto:d365fogl@microsoft.com)) stating that your new LCS project is ready for Microsoft to move your production database and Azure Blob Storage. To ensure that the process will run smoothly, provide the following details in the email. We suggest that copy the following list to your email, and then answer all of the information line by line.

## Lifecycle Services

- Provide the LCS IDs (number in the LCS project URL) for source and target LCS project.
- Confirm that the go-live date is set correctly in the target LCS project.
- Confirm that the update schedules are set in the target LCS project (**LCS > Menu > Project settings > Update settings**).
- Confirm if you are using Azure Blob Storage for document attachments.
- Confirm that your project is identified as a tenant move in the Project Onboarding wizard.

## Testing

- Confirm that the smoke testing is completed on the sandbox environment (Tier-2 or higher) in the target LCS project.

## Code Management

- Confirm that your deployable package is marked as a release candidate in the target LCS project.
- List the ISV solutions you are using.
- Confirm which version your old production environment is running on.
- Confirm that non-standard code to be applied in the new production environment will be exactly the same as the non-standard code present in the old production environment in order to prevent database copy issues.
- Confirm if there were any non-typical actions taken on your old production environment which need to be considered on the new production environment, like installation of a custom font or environment upscale.

## Environment

- Share which environment version you plan to deploy your new production environment.
  - Describe how you will conduct your cut over.
  - Confirm the dates when the source LCS environments and project will be deallocated and deleted.
4. The Dynamics 365 FO Go-Live team will reply to you within 2 business days and a FastTrack Solution Architect will work with you on the assessment of the project readiness for production deployment.
  5. When the tenant move assessment is successfully completed, the FastTrack Solution Architect will approve your production request for deployment.
  6. Create the production deployment request on the new LCS project.
    - It is not possible to select the same name for the new production environment, as it is in use for your old production environment. You will need to choose a new environment name so that a new URL will be generated.
    - Make sure you select the same application version that is used by your current production environment.
    - In the Production configuration wizard, select a generic user account, not a named user, as Environment Administrator.
  7. After the production environment has been deployed, verify that source and target environments have exactly the same code, otherwise migration will fail. If necessary, deployable packages must be installed on the target production environment.
  8. Request to copy database and blob storage from the old production environment to the new production environment.
    - **Cloud deployment to self-service deployment:** [Submit a service request](#) of type **Other** to request that the Microsoft Service Engineering team copy the database and blob storage, if applicable, from the old production environment to the new production environment. Be sure to include LCS IDs and environment IDs from source and target projects in the service request.

- **Both projects (old and new) are self-service deployments:** Submit a **support ticket** requesting a copy of the database and blob storage, if applicable, from the old production environment to the new production environment. Be sure to include LCS IDs and environment IDs from source and target projects in the support ticket.
  - a. This process will require interaction between Microsoft and the implementing project team. Ensure that you follow the email notifications or notifications directly in the service request.
  - b. After Microsoft has completed the activity and provided you with updated information, you will need to validate the new production environment.
  - c. If you encounter an issue after the migration, file a support ticket.

## Tear down the LCS project on the old tenant

After the new LCS project on the new Azure AD tenant is fully functional, you must stop, deallocate, and delete the environments on the old LCS project. When you've finished, the **Configure** button becomes available for each environment. If you already had a production environment on the old tenant, you must file a support ticket to have it deleted.

You should save any remaining artifacts from the Asset library that you might require later.

After all environments have been deleted and all artifacts saved, an Organization Administrator on the old tenant must delete the LCS project. Microsoft reserves the right to disable the customer's account and delete the customer data after the service has been suspended for an extended period.

## Suspend subscriptions on the old tenant

After all the environments have been deleted, and you've saved the LCS artifacts that you require, work with your cloud service provider or Volume Licensing Support to suspend all the licenses on the old Azure AD tenant.

- **Cloud service provider** - Suspend the existing subscriptions against the old tenant.
- **Volume Licensing Support** - Call Volume Licensing Support to confirm that you've completed the work and that the subscriptions can now be suspended against the old tenant.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Move LCS implementation projects from on-premises to the cloud

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic explains how to move your Microsoft Dynamics 365 Finance + Operations (on-premises) environments that are hosted on your own infrastructure to the Azure cloud.

## Cloud subscription licenses

If you don't already have cloud subscription licenses, work with your cloud service provider or volume license reseller to get and activate the required subscriptions on your Azure Active Directory (Azure AD) tenant. All subscriptions for users and add-on environments must be activated.

## Configure LCS cloud implementation project

If no Finance and Operations cloud-named user subscription licenses have previously been activated on the Azure AD tenant, a new Microsoft Dynamics Lifecycle Services (LCS) cloud implementation project is automatically provisioned. Otherwise, you must open a support request to have an LCS cloud implementation project created. For more information, see [Multiple LCS projects and production environments on one Azure AD tenant](#).

After your LCS cloud implementation project has been created, you must fully configure it. As part of this configuration, you must add users, an Azure DevOps association, and subscription estimates, fill in the Asset library and Business process modeler (BPM), and more.

### NOTE

While you're onboarding your project, you must select **AX 2012 Upgrade** as the source system, so that a singleton Azure SQL database will be used for your sandbox instead of an elastic pool. Eventually, a more appropriate option will be available, such as **On-premises Finance and Operations**.

## Complete development and testing of updated integrations

You will probably have to make some changes to the integration design patterns that you used for interfaces with your Finance + Operations (on-premises) environment. These changes can be substantial, and a detailed discussion of them is beyond the scope of this topic. Nevertheless, you must evaluate all your interfaces and make the appropriate changes to them.

You should consider developing your updated interfaces in such a way that they can coexist in the same code base as the original interfaces. This approach will simplify code lifecycle management during the period of your transition from on-premises to cloud. If this approach isn't possible, you must manage a new development branch through your cloud go-live. To simplify management of this new branch during the transition period, we recommend that you freeze other code changes as much as you can. Additionally, in your detailed cut-over plan, you should carefully document the steps for inactivating your old interfaces and activating the new interfaces.

## Do a trial migration and resolve issues

1. Deploy a tier-2 environment.

2. Apply the same code package that is applied in your on-premises production environment (or, as appropriate, in the current build from the cloud integration development branch that was discussed in the previous section). This code package should be a single, complete deployable package that includes any independent software vendor (ISV) solutions and licenses.
3. In SQL Server Management Studio (SSMS), run the following Transact-SQL (T-SQL) commands against the sandbox database to preserve the current Admin account, Azure AD tenant ID information, and Data management framework (DMF) shared working directory in that database. Save the results.

```
SELECT SID,NETWORKKALIAS,NETWORKDOMAIN,IDENTITYPROVIDER from USERINFO WHERE ID = 'Admin'  
SELECT VALUE from SYSSERVICECONFIGURATIONSETTING where name = 'TENANTID'  
SELECT TENANTID from POWERBICONFIG  
SELECT TENANTID from PROVISIONINGMESSAGETABLE  
SELECT TENANTID from B2BINVITATIONCONFIG  
SELECT TENANTID from RETAILSHAREDPARAMETER  
SELECT SHARED_FOLDERPATH from DMFPARAMETERS
```

4. Copy the database from on-premises to online. The export and import process that you use is the same process that is described in the [Golden configuration promotion](#) database movement tutorial. However, in this case, the source database is the existing on-premises production SQL database, and you must use the sqlpackage.exe approach that is described for importing into a developer environment. If you use the LCS self-service database import option instead, some data won't be imported, as noted in the warnings about data elements that are cleaned up. The target database information that is available in the LCS environment details must be used instead of the placeholders that are shown in the following code.

```
SqlPackage.exe /a:import /sf:D:\BacpacToImport\my.bacpac /tsn:<Azure SQL database server> /tdn:  
<target database name> /tu:<axdbadmin user from LCS> /tp:<axdbadmin password from LCS>  
/p:CommandTimeout=1200
```

5. Restore the Admin account, Azure AD tenant ID, and DMF shared directory values. Also remove the SF schema and its tables, if they are present.

```
UPDATE USERINFO SET SID='<preserved SID>', NETWORKKALIAS='<preserved NETWORKKALIAS>',  
NETWORKDOMAIN='<preserved NETWORKDOMAIN>', IDENTITYPROVIDER='<preserved IDENTITYPROVIDER>' WHERE ID =  
'Admin'  
UPDATE SYSSERVICECONFIGURATIONSETTING set VALUE='<preserved VALUE>' where name = 'TENANTID'  
UPDATE POWERBICONFIG SET TENANTID='<preserved TENANTID>'  
UPDATE PROVISIONINGMESSAGETABLE SET TENANTID='<preserved TENANTID>'  
UPDATE B2BINVITATIONCONFIG SET TENANTID='<preserved TENANTID>'  
UPDATE RETAILSHAREDPARAMETER SET TENANTID='<preserved TENANTID>'  
UPDATE DMFPARAMETERS SET SHARED_FOLDERPATH='<preserved SHARED_FOLDERPATH>'  
DROP TABLE IF EXISTS SYNCLOG  
DROP TABLE IF EXISTS SYNCLOCK  
DROP SCHEMA IF EXISTS SF
```

6. Reimport all other users, and assign the appropriate security roles.
7. Direct printing in a cloud environment is done via the Document Routing Agent (DRA). Set up sandbox DRAs as described in [Install the Document Routing Agent to enable network printing](#), so that regression testing can include your printing scenarios.
8. Copy document handling attachments to the cloud. Document handling attachments aren't stored in the database. If they must be preserved, you must move them separately. For instructions, see the [Migrate document handling attachments to your sandbox](#) section later in this topic.
9. Run a complete regression test cycle. This cycle should include testing of integrations.



10. Resolve any issues that are discovered during testing. For each issue, document and keep track of the correcting adjustments that you make in the sandbox, and repeat them in the on-premises source. If any change must not be made in the on-premises environment, because it's incompatible with the correct functioning of that environment, we recommend that you create a DMF data package for it instead of manually applying it for each iteration of the migration process.
11. Repeat steps 2 through 10 until all tests have been passed, and no further changes are being made to code or the configuration.

## Repeat the migration to production

1. Deploy the new production environment. Note that the regular prerequisites apply. For example, you must have an active subscription estimator, complete the LCS methodology phases before the operate phase, and complete the FastTrack readiness review. For more information, see [Prepare for go-live](#).
2. Apply the final version of the software deployable package to production.
3. Stop making data changes to the on-premises production environment.
4. Repeat steps 3 through 6 in the [Do a trial migration and resolve issues](#) section to copy the final/up-to-date on-premises production database to the cloud sandbox.
5. Repeat step 8 in the [Do a trial migration and resolve issues](#) section to copy the final/up-to-date document handling attachments to the cloud sandbox.
6. Request a database refresh from sandbox to production. (The process is the same as the process that is used to promote a golden configuration database to production.)
7. Open a support request to have Dynamics Support Engineering copy the document handling attachments from the sandbox storage account to the production storage account and update the references in the production database's DocuValue and DocuDeletedValue tables. After the request has been completed, validate that the attachments are available for a sample of document handling records.
8. Set up DRAs for production. If you're reusing any of the DRAs that were previously installed as part of your trial migration, remember to update their configuration so that they connect to the production URL instead of the sandbox URL.
9. Reconcile your cloud and on-premises production environments, as detailed in your cut-over plan.
10. Obtain sign-off for the go-live.
11. Activate cloud production interfaces, batch jobs, and so on.
12. Start to transact in your cloud production environment.

## Migrate document handling attachments to your sandbox

Document handling attachments for Finance + Operations (on-premises) environments are stored in a file share. However, the cloud version doesn't support this file share. You can use the following procedure to copy the attachments to the Azure storage account for your sandbox environment and update the corresponding metadata in the database. For subsequent promotion to production, you can request that Dynamics Support Engineering copy the attachments from your sandbox to production.

1. Upload a copy of the document handling attachment files from the on-premises production file share to a temporary folder on one of the sandbox instances of Application Object Server (AOS). For example, you can upload a zip file of the attachments and unpack it on the target. If you don't have remote desktop access (for example, for a self-service environment), you can use a different virtual machine (VM) instead. For reasonable conversion performance, this VM should be in the same Azure datacenter as the target sandbox. If you aren't using the AOS instance, you must add the VM to an allow list for access to the sandbox's SQL database instance.
2. Open a support request to get the name of the sandbox Azure storage account and a time-limited shared access signature token for the documents container. Update the corresponding placeholders in the Windows PowerShell script that is run in the next step. Also update the placeholders for your temporary

folder, and for your Finance and Operations transactional database, by using the environment details in LCS.

3. Run the following Windows PowerShell script on the sandbox AOS instance or other VM to upload the document handling files to the storage account and create the required metadata for each file.

```
#Upload F&O on-prem document handling attachments to Azure storage account
#
$filePath = "<TEMP_ATTACHMENTS_FOLDER_PATH>"
$dbHostName = "<DATABASE_SERVER>.database.windows.net"
$dbName = "<DATABASE_NAME>"
$dbUsername = "<DATABASE_USER>"
$dbPassword = "<DATABASE_PASSWORD>"
$storageAccountName = "<STORAGE_ACCOUNT_NAME>"
$sasToken = "<SAS_TOKEN>"

[Reflection.Assembly]::LoadWithPartialName("System.Security.Cryptography") #Load crypto
$cryptoObj = [System.Security.Cryptography.SHA256]::Create()
$storageContext = New-AzStorageContext -StorageAccountName $storageAccountName -SasToken $sasToken
foreach ($file in Get-ChildItem $filePath)
{
    try
    {
        $blob = (Set-AzStorageBlobContent -Context $storageContext -Container documents -File
$file.FullName -Blob "$($file.Name)" -Force).ICloudBlob
    }
    catch
    {
        Write-Host "Could not upload $($file.Fullname) to blob"
        Write-Host $_
    }
    if($blob)
    {
        #Write-Host "Processing $($file.Fullname)..."
        #FileHash:
        $fileBytes = [System.IO.File]::ReadAllBytes($file.FullName)
        $hashBytes = $cryptoObj.ComputeHash($fileBytes)
        $encodedHash = [System.Convert]::ToBase64String($hashBytes)
        #FullFileName:
        $origFileName = (Invoke-Sqlcmd -Query "SELECT ORIGINALFILENAME FROM DOCUVALUE WHERE FILEID =
'$($file.Name)'" -ServerInstance $dbHostName -Database $dbName -Username $dbUsername -Password
$dbPassword).ORIGINALFILENAME
        if ($origFileName.Length -eq 0)
        {
            $origFileName = (Invoke-Sqlcmd -Query "SELECT ORIGINALFILENAME FROM DOCUDELETEDVALUE
WHERE FILEID = '$($file.Name)'" -ServerInstance $dbHostName -Database $dbName -Username $dbUsername -
Password $dbPassword).ORIGINALFILENAME
        }
        if ($origFileName.Length -eq 0)
        {
            Write-Host "Missing DOCUVALUE $($file.Name)"
        }
        else
        {
            $nameBytes = [System.Text.Encoding]::UTF8.GetBytes($origFileName)
            $encodedName = [System.Convert]::ToBase64String($nameBytes)
            #Write-Host "Base64 encoded original filename $encodedName."
            $blob.Metadata["FileHash"] = $encodedHash
            $blob.Metadata["FileSize"] = $file.Length
            $blob.Metadata["FullFileName"] = $encodedName
            $blob.SetMetadata()
            Write-Host "Uploaded $($file.Fullname)"
        }
    }
}
}
```

4. In SSMS, run the following T-SQL commands to update the DocuValue and DocuDeletedValue records so that they reference the target storage location.

```
update DOCUVALUE
set ACCESSINFORMATION = replace(ACCESSINFORMATION, 'file://<SOURCE_PREFIX>/documents/',
'https://<STORAGE_ACCOUNT>.blob.core.windows.net/documents/'), STORAGEPROVIDERID = 1
where STORAGEPROVIDERID = 4 --4 for LBD filesystem, 1 for Azure blob
and ACCESSINFORMATION like 'file://<SOURCE_PREFIX>/documents/%'

update DOCUDELETEDVALUE
set ACCESSINFORMATION = replace(ACCESSINFORMATION, 'file://<SOURCE_PREFIX>/documents/',
'https://<STORAGE_ACCOUNT>.blob.core.windows.net/documents/'), STORAGEPROVIDERID = 1
where STORAGEPROVIDERID = 4 --4 for LBD filesystem, 1 for Azure blob
and ACCESSINFORMATION like 'file://<SOURCE_PREFIX>/documents/%'
```

5. Test a sample of the document handling attachments to make sure that they can now be accessed in the sandbox environment.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Multiple LCS projects and environments on one Azure AD tenant

2/18/2021 • 4 minutes to read • [Edit Online](#)

For any new cloud project, one Microsoft Dynamics Lifecycle Services (LCS) Implementation project is instantiated on a Microsoft Azure Active Directory (Azure AD) tenant that provides access to one production instance. In rare cases, to handle the requirements of a specific implementation, you might require multiple production instances that run in parallel. By creating multiple LCS projects against the same Azure AD tenant, you can have multiple production instances. Here are the most common scenarios where multiple production instances might be required:

- A global implementation's requirements for data residency, latency, or data volume can't be met by one instance.
- Different business units in an organization are implementing the product separately as independent applications.

Manual intervention by the Microsoft Dynamics Service Engineering (DSE) team is required in order to create additional LCS projects on a shared Azure AD tenant. This approach should be used only if a single-instance strategy truly isn't feasible. Before additional LCS projects can be created, customers must provide the business justification and confirm that they understand all the implications of the approach. This process should be started as early in the implementation lifecycle as possible. Customers who decide to proceed should inform the FastTrack solution architect who is assigned to their project that they require additional LCS projects. If no solution architect is assigned to their project, customers should open a support ticket.

## Licensing requirements

Every LCS Implementation project that runs on the same Azure AD tenant must satisfy the minimum licensing requirements. For example, if there are three LCS Implementation projects on the same Azure AD tenant, a customer must purchase no less than three times the minimum number of subscription licenses. Currently, the minimum license requirement is 20 full user licenses. Therefore, to run three LCS Implementation projects on the same Azure AD tenant, the customer must purchase at least 60 licenses.

Because the licenses are associated with the Azure AD tenant, the **Subscriptions available** page for every LCS project will show the total number of licenses, even though a given LCS project can use only the portion of licenses that has been allocated to it. This allocation of license to LCS projects must be documented outside the system.

Users who access multiple environments in parallel must be licensed separately for each environment. A user can only be assigned one license for each product for each Azure AD tenant. This allocation of licensing requirements for LCS projects for specific users must be documented outside the system. For additional information about licensing, download the [Licensing guide](#).

## Disadvantages of multiple LCS projects

There are some disadvantages to having multiple LCS projects. Here are some of them:

- Master data isn't shared.
- Intercompany transactions aren't supported.
- Integrations must be configured in each LCS project.
- Each LCS project requires a separate Bring your own database (BYOD) instance

- User acceptance testing (UAT) must be done on each instance, even if the code is the same. UAT is required on each instance, because differences can occur across the LCS projects, even if they share a code base. One source of differences can be the integration setup and BYOD configuration that must be done separately in each LCS project and therefore must be tested in each LCS project. Additionally, there might be data variations, different application configurations per region might affect functionality, and different data centers might support a different set of Azure services.
- Microsoft Azure DevOps must be configured in each LCS project. When customizations and code are shared, it makes sense to use the same Azure DevOps project.

## Advantages of multiple LCS projects

There are also advantages to having multiple LCS projects. Here are some of them:

- Data centers can be selected per LCS project to provide the best latency experience.
- Data centers can be selected per LCS project to satisfy statutory requirements for data residency.
- There is more flexibility to schedule servicing operations, such as code deployments and upgrades.

## Requesting multiple LCS projects on the same Azure AD tenant

If your solution requires multiple LCS projects on the same Azure AD tenant, all LCS projects except the original project must be provisioned on demand by the DSE team. You should inform the DSE team about this requirement as early as possible, ideally when the project is being onboarded. For more information, see [Onboard an implementation project](#). To request additional LCS Implementation projects, the customer must create a support request by using the Support portal in LCS. In this request, the customer must provide the following information:

- The business justification.
- The enterprise and project structure. This information includes the following details:
  - The name of the Implementation project
  - The breakdown of licenses per LCS project
- Confirmation that the customer understands the implications of multiple LCS projects on the same Azure AD tenant.

## Online deployments in China sovereign cloud

If your implementation includes China deployment/rollout, be informed that Dynamics 365 Finance online deployment became available in Mainland China starting in April 2019. For more information, see [Finance and Operations apps - operated by 21Vianet in China](#). This deployment is designed to comply with regulatory requirements in China and the services include a physically separated instance of a cloud service with a different tenant (Azure Active Directory) that is operated and transacted by 21Vianet.

This is a single organization in multiple clouds with different tenant (Azure Active Directory). The advantages and disadvantages of multi-Lifecycle Services projects or production environments described above are still applicable, but the licensing requirement and requesting procedure are different. Work with your Microsoft Account Executive or your implementation partners for any process assistance.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Implement Commerce projects

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides links to other topics that will help new implementers of Commerce projects with important aspects of the implementation process, so that they can avoid complications. The guidelines that are mentioned are best practices that have been collected during previous implementation projects. Therefore, implementers can focus on the actual feature work and not become bogged down by the procedures.

Some of the information in these topics can also be found on blogs and pages at other locations, such as [Development and administration for Finance and Operations](#) and <https://dynamicsnotes.com>. You can read those blogs and pages separately, or you can read this topic from beginning to end and review the linked documents within the subtopics as you require them during the implementation process.

The information focuses on implementations of Finance and Operations apps that include Commerce functionality.

[Set up new environments, Azure DevOps, and branches for Commerce projects](#)

[Update code and environments for Commerce projects](#)

[Testing and performance issues](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up new environments, Azure DevOps, and branches for projects

2/18/2021 • 18 minutes to read • [Edit Online](#)

Most environments for Microsoft Dynamics 365 Commerce projects are hosted in the cloud. They are either Microsoft-hosted on a Microsoft subscription or cloud-hosted on a customer subscription. By default, environments are Microsoft-hosted. You can use cloud-hosted environments to provide more control over a development or build environment. For more details, see the [Lifecycle Services \(LCS\) user guide](#).

## Development Tier 1 environments

Development environments are called Tier 1 environments. There are three options for hosting a development environment:

- The Commerce app comes with one Sandbox Tier 1 environment. (For details, see the [Microsoft Dynamics 365, Enterprise edition, Licensing Guide](#).)
- A cloud-hosted environment that you run on your own Microsoft Azure subscription. This type of environment is known as "cloud-hosted" in Microsoft Dynamics Lifecycle Services (LCS).
- A downloaded virtual machine (VM) that you host in a location of your choice.

If your implementation of Commerce includes code extensions, we recommend that you use a development environment where you have administrator privileges. If you don't have administrator privileges in your development environment, you won't be able to install programming tools or configure the operating system.

The hosting model that you choose has a financial impact. You can reduce some of the hosting cost by using a Tier 1 environment as a simple test environment or golden configuration environment. One Tier 1 environment is free with your Dynamics 365 subscription. Although this approach isn't ideal, it should work for most projects.

If you want to extend channel components, see [Prepare the development environment](#) to learn how to configure a development environment so that it's ready for development.

### NOTE

You can shut down cloud-hosted environments at any time. This capability helps reduce the hosting cost.

A hosting alternative is to download a virtual hard disk (VHD) from LCS and host it locally on a server. From a development perspective, VHD images have the same capabilities as a hosted VM. The only difference is that LCS deployments aren't supported on VHDs. However, command-line deployments are supported.

The following table shows the advantages and disadvantages of each hosting model. Use this information to evaluate the model that will work best for your project.

HOSTING MODEL	ADVANTAGES	DISADVANTAGES
---------------	------------	---------------

HOSTING MODEL	ADVANTAGES	DISADVANTAGES
Microsoft-hosted environment (in an LCS project, default or based on an add-on)	Your subscription includes one Tier 1 environment. We recommend that you use this environment as a build environment.  Telemetry data is collected and is available on the LCS diagnostics page.	Users can't perform administrative actions.  Users can't install any tools or certificates.
Cloud-hosted environment (in an LCS project, private subscription)	You have full administrative rights.  You can install tools and certificates.	There is additional cost. You can mitigate this cost by shutting down the environment.
Self-hosted downloaded VM	The experience depends on the host.  The experience can be much faster if the VM runs on a solid-state drive (SSD).	You can't deploy packages from LCS.

Tier 2 and higher machines are multi-box environments for multiple test and verification purposes. Production environments are hands-off, and the size of the environment is determined by the sizing process in LCS.

## Branches, build definitions, and environments

Branching is an important practice in software development. The [Branching and Merging Primer](#) topic describes the advantages of branching:

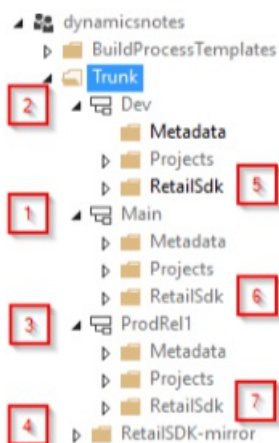
### NOTE

A branching and merging strategy involves a tradeoff between risk and productivity. You trade the safety of working in isolation for the increased productivity of working with other people. The productivity increases come with a cost—the additional effort required for merging software assets sometime in the future.

Using branches provides better isolation and control of individual software assets and increases productivity, because teams or individuals can work in parallel. However, using branches also requires an increase in merge activities and therefore risk, because you must later reassemble branches into a whole.

There is no single best strategy for creating branches. The strategy depends on the project and the size of the implementation.

The following illustration shows three code branches: Dev, Main, and ProdRel1. The numbers indicate the order of setup.





Here is an explanation of the setup. The numbers in brackets refer to the numbers in the preceding illustration:

- The **Dev** branch [2] is used for daily work that isn't ready for testing or might not be stable, but that must be shared with other developers. For larger teams, you might want to have multiple Dev branches for different features or purposes.
- The **Main** branch [1] is for changes that meet a certain quality bar and are ready for testing by other people. This testing might include user acceptance tests, performance tests, integration tests, and sanity tests after hotfixes. Deployable packages for this branch must be created by a build environment. As a best practice, you should not generate X++ packages in a Tier 1 environment and then deploy those packages into an official test or production environment. Otherwise, uncommitted source changes could be included. The correct approach is always to deploy packages that were built on official build environments.
- The **ProdRel1** branch [3] holds all source code exactly as it's deployed in a production environment at any given point. A build environment can be used but isn't required. If packages from the Main branch are deployed to a production environment, the code should be merged (from Main to ProdRel1) after a production deployment. By having a branch for production, you can generate official builds later if you require them.
- All three branches hold both X++ code (extensions and hotfixes in Metadata folders) and a copy of the Retail software development kit (SDK) in **RetailSdk** folders [5, 6, 7]. The Retail SDK includes base Microsoft code and code extensions. This base code and the code extensions can differ in each branch.
- The **RetailSdk-mirror** folder [4] is used to bring in Microsoft changes to the Retail SDK. It isn't used for development or build purposes. It should be updated only when a new version or hotfix is used. For a detailed description of the process, see this [cheat sheet](#).

For small projects, it's acceptable to have only two branches (Main = Dev branch). However, developers must be more disciplined, because any code submissions can immediately affect the quality of test builds.

You can build deployable packages out of multiple branches. In this case, you must have one build definition for each branch that can be built. The initial build definition is automatically created when a build environment is deployed (Main branch). You can make copies of the build for other branches. Note that you must make small additions to incorporate the Commerce code.

The following high-level steps are used to set up an environment so that development work can begin. For details about the numbers in brackets, see the previous illustration and the related information.

1. Deploy a build environment and an empty Main branch in Microsoft Azure DevOps [1].
2. Deploy a development environment.
3. Create the Dev branch and the release branch (for example, ProdRel1 in the previous illustration) [2, 3].
4. Add the Retail SDK [4–7].
5. Prepare the development environment.
6. Optional: Deploy a second build environment for a different release branch.
7. Prepare the build definitions.

After you've completed all these steps, your branches, environment, and builds will be ready.

The following sections explain each step in detail.

## Deploy a build environment and an empty Main branch in Azure DevOps

Use the LCS portal to deploy a new build environment. We recommend that you use a cloud-hosted environment, because you will have more options and capabilities if you have administrative rights. See the table about the various environment hosting models in the "Development Tier 1 environments" section, earlier in this topic.

Start by creating a new Azure DevOps project if you don't already have one. In your Azure DevOps account, select **New project**.

## Create new project

Projects contain your source code, work items, automated builds and more.

Project name \*

dynamicsnotes



Description

Version control

Team Foundation Version Control



Work item process

Agile



Create

Cancel

After you create the new Azure DevOps project, you must give Azure DevOps access to it. First, create a new personal access token on the Azure DevOps account. Then configure the LCS project with the correct URL and personal access token.

## Setup Visual Studio Team Services

**1** Enter the Visual Studio Team Services site  
Enter the Visual Studio Team Services site URL to allow Lifecycle Services to connect and manage resources.

**2** Select the Visual Studio Team Services project  
Choose the Visual Studio Team Services project in the selected site to link with this Lifecycle Services project.

**3** Review and save  
Review and save the Visual Studio Team Services settings for this Lifecycle Services project.

Enter a Visual Studio Team Services site URL to allow Lifecycle Services to connect and access resources.  
Example URL format accepted:  
<https://org.visualstudio.com/>

Create a personal access token for the Visual Studio Team Services site and allow all authorized scopes. Personal access tokens are used instead of a password to allow Lifecycle services access the resources stored in your account.  
To setup a personal access token, on the visualstudio.com site, click your name in the top right and select Security > Personal access tokens and create a new token for the Visual Studio Team Services account. You can get more information on how to create a personal access token at:  
<http://go.microsoft.com/fwlink/?LinkID=627398>

You will also need to choose a Visual Studio project. If you do not have a project, you can create one at  
<https://www.visualstudio.com>

Visual Studio Team Services site URL:

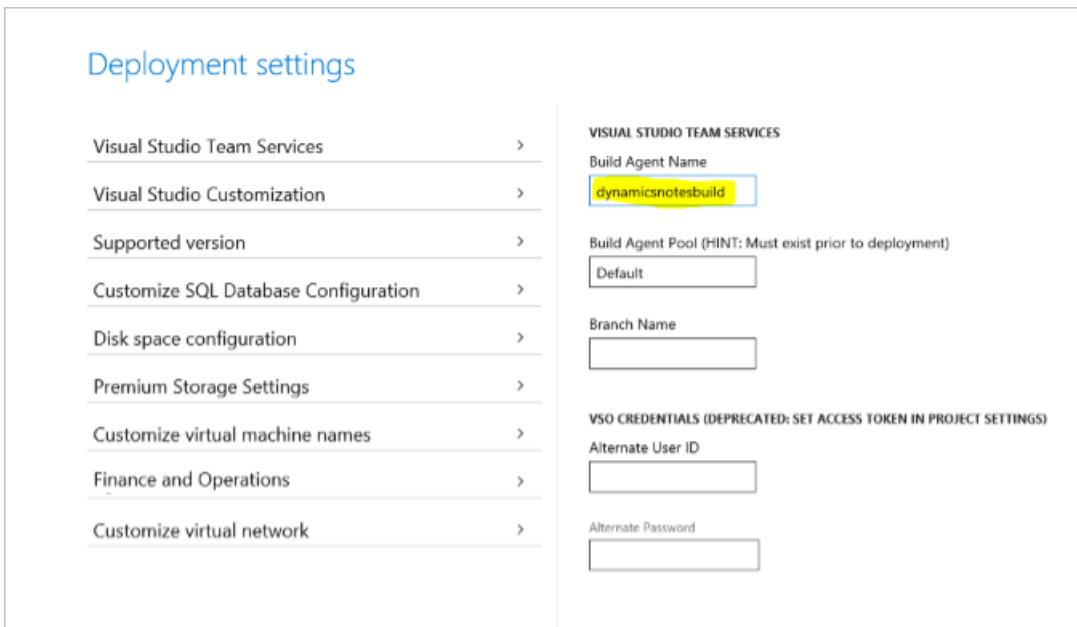
Personal access token:

[Privacy statement](#)

[Continue](#)

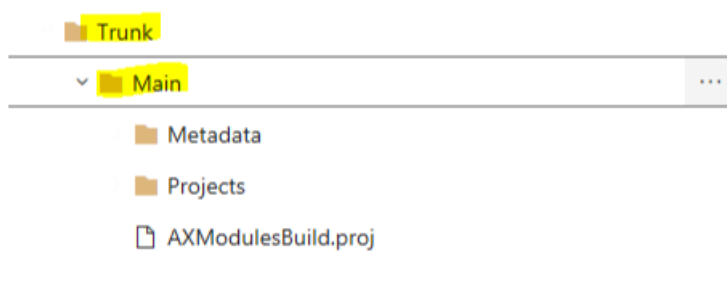
After the LCS project is linked to Azure DevOps, you're ready to deploy.

Add a new environment, select the version, select **DEVTEST** as the topology, and select a build environment. On the next page, enter a meaningful name for the environment. Then enter a similar name for the build agent.

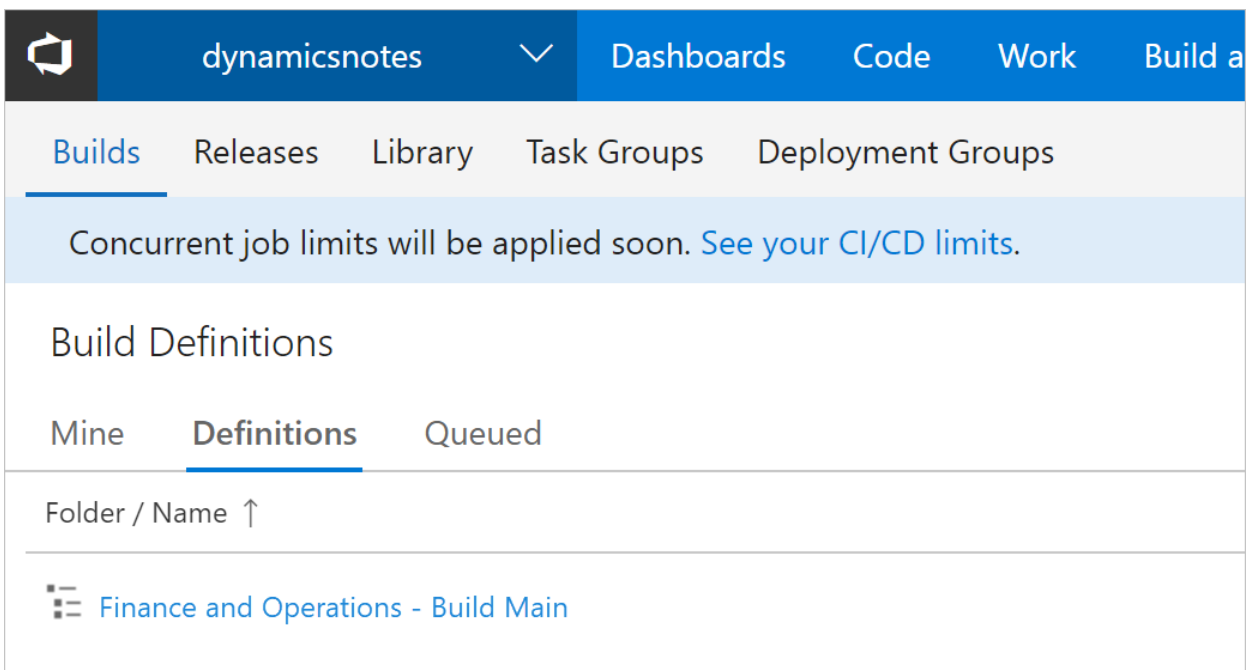


Next, under **Customize virtual machine names**, enter a unique name, and then deploy the VM.

The build box is deployed, and the build definition and Main branch are created, as shown in the following illustration. This process might take a couple of hours.



The build appears in the list of build definitions.



The build definition appears in the **Agents for pool Default** grid.

dynamicnotes

Dashboards Code Work Build and Release Test Wiki

Overview Work Security Version Control **Agent Queues** Notifications Service Hooks Services

New queue... Manage pools All Queues

- Default (Default)
- Hosted (Hosted)
- Hosted Linux Preview (Hosted Li...)
- Hosted macOS Preview (Hosted ...)
- Hosted VS2017 (Hosted VS2017)
- PRODREL1 (PRODREL1)

Agents for pool Default [Download agent](#)

Agents Roles Details

Enabled	Name	State	Current Status
<input type="checkbox"/>	VSTAgent_RetailTC...	Offline	Idle
<input type="checkbox"/>	VSTAgent_Transact...	Online	Idle
<input type="checkbox"/>	VSTSAgent-BLD17...	Offline	Idle
<input type="checkbox"/>	VSTSAgent-SLD72...	Offline	Idle
<input type="checkbox"/>	VSTSAgent-spring...	Offline	Idle
<input type="checkbox"/>	VSTSAgent_D365F...	Offline	Idle
<input checked="" type="checkbox"/>	dynamicsnotesbuild	Offline	Idle

## Deploy a development environment

Use the LCS portal of your implementation project to create a cloud-hosted development environment.

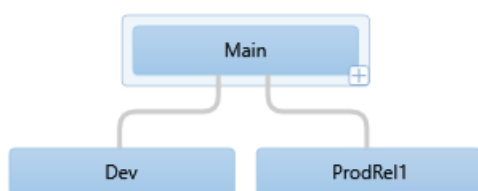
1. Make sure that you're signed in to the correct user account. This user account is used to create the tenant of the development machine. For example, if you're signed in to LCS as `lily@pad.com`, the environment is set up for the `@pad.com` tenant and expects users from that tenant. Although other users can be added, point of sale (POS) activation must be done by a user from that tenant. In some cases, user accounts from different domains can be used, such as when customers, partners, or other parties use email accounts from different domains. In these cases, coordination is required during POS activation, because only the tenant that was used during deployment can activate users.
2. Select the correct version, select **DEVTEST**, and then select **DEV**. Enter a meaningful and unique name, and make sure that the machine name is also unique in the advanced settings. The process of preparing the machine might take a couple of hours.

Because there is currently no Dev branch, you can skip the process of mapping Azure DevOps to the local directories. However, you will have to complete that process later.

## Create the Dev and release branches

As previously mentioned, you must have a branch that holds changes that are often made but less often tested. You must also have a branch that holds the source code for production. The following illustration shows the expected hierarchy.

### Main Branch Hierarchy



Follow these steps to create the branches.

1. Sign in to a development environment.
2. Start Microsoft Visual Studio as an administrator. Use an account that has access to the Azure DevOps project.
3. In Team Explorer, connect Visual Studio to the Azure DevOps project, if this connection doesn't already exist.
4. Map the **Trunk/Main** folder to a local folder (if this mapping doesn't already exist). This mapping is temporary.
5. In Source Control Explorer, right-click the **Main** folder, and then select **Branching and Merging > Convert to Branch**.
6. Right-click the **Main** branch, select **Branching and Merging > Branch**, and name the new branch **Dev**.
7. Use **Pending Changes**, and submit this change to Azure DevOps.
8. Right-click the **Main** branch, select **Branching and Merging > Branch**, and name the new branch **ProdRel1**.
9. Use **Pending Changes**, and submit this change to Azure DevOps.

At this point, Source Depot Explorer in Visual Studio resembles the following illustration.



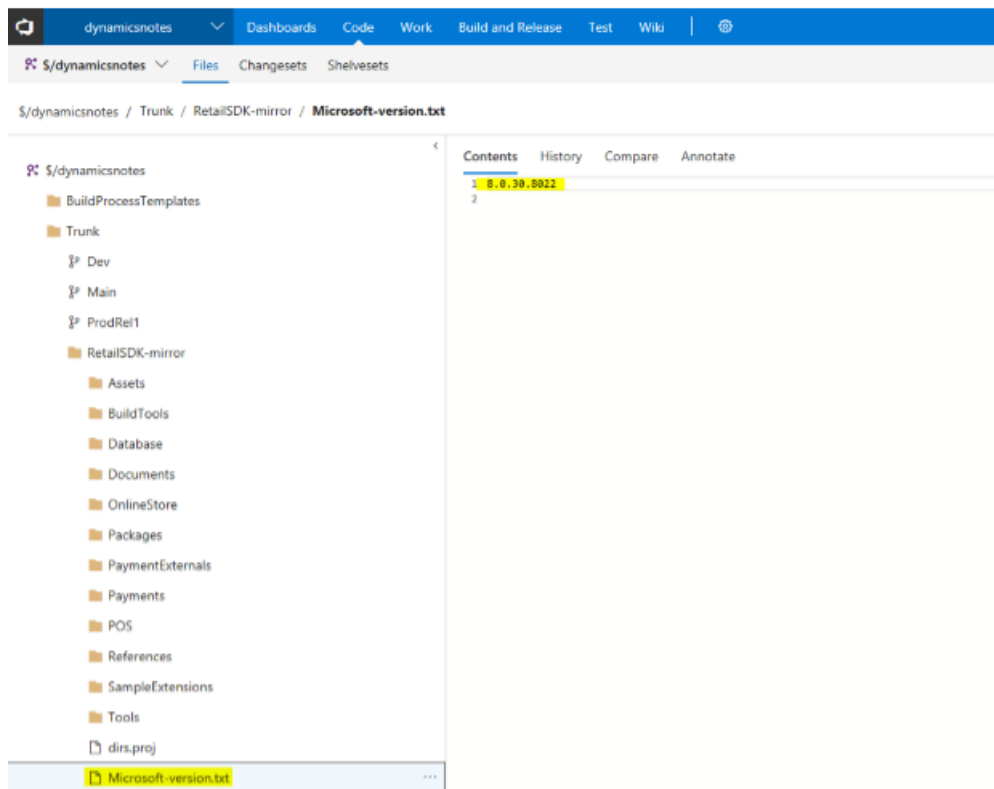
## Add the Retail SDK

Next, you must add the Retail SDK to each of the three code branches, so that code changes can be propagated from Dev to Main and eventually to ProdRel1. This step also enables separate changes between these branches, as for the X++ code. Therefore, we will have the Retail SDK in every branch, together with the X++ code.

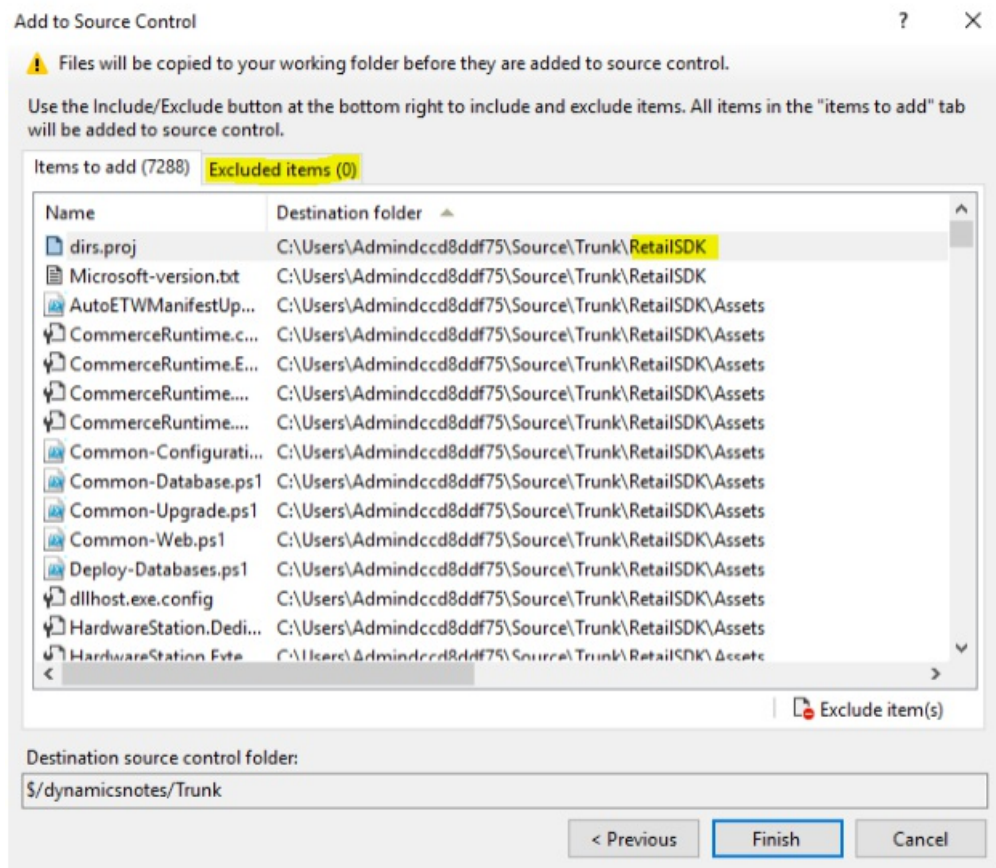
First, add the mirror branch. The Retail SDK mirror branch is required as a baseline for code merges when updates from Microsoft are imported. The process for taking updates will be explained later in this topic.

The mirror branch or folder is only required one time per project.

1. Find the unchanged Retail SDK that has the exact version that you want to start your development with. This Retail SDK can be found on every development machine on the service drive, or in every downloaded hotfix. You can uniquely identify a version of the Retail SDK by inspecting the Microsoft-version.txt file. This file should not be changed, except by an update to the Retail SDK mirror folder.



2. In Source Control Explorer, right-click the **Trunk** folder, and then select **Add Items to Folder**.
3. Select the top folder in the Retail SDK, and then select **Next**.
4. Visual Studio shows the number of files that will be added. Make sure that the **RetailSdk** folder is under the **Trunk** folder.
5. Make sure that there are 0 (zero) excluded items by selecting items and then selecting **Include items**.

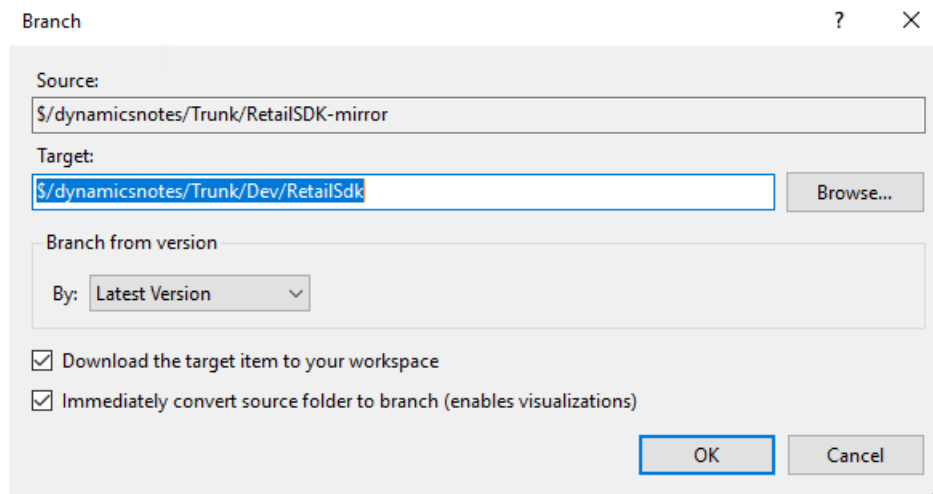


6. Select **Finish**. This process will take a few minutes.

7. When the process is completed, rename the folder **RetailSdk-mirror**.

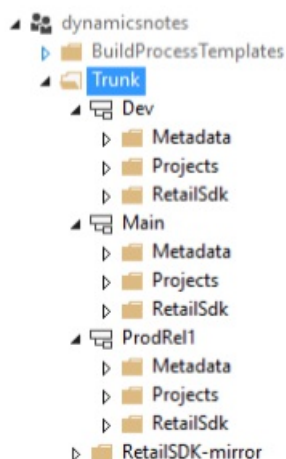
Next, you must branch to each branch. Follow the same path that the code changes will flow in: first to Dev, then to Main, and then to ProdRel1.

1. Select the folder for the mirror branch, right-click, and then select **Branching and Merging > Branch**.
2. Go to the **Dev** branch, append **/RetailSdk** to the name, and then select **OK**.



3. Use **Pending Changes**, and submit the changes.
4. Follow the same steps to branch the **RetailSdk** folder of the **Dev** branch to the **Main** branch.
5. Follow the same steps to branch the **RetailSdk** folder of the **Main** branch to the **ProdRel1** branch.

At this point, you have the code branches and code locations for the X++ and Commerce extensions setup. In Source Control Explorer, the file structure should resemble the following illustration.



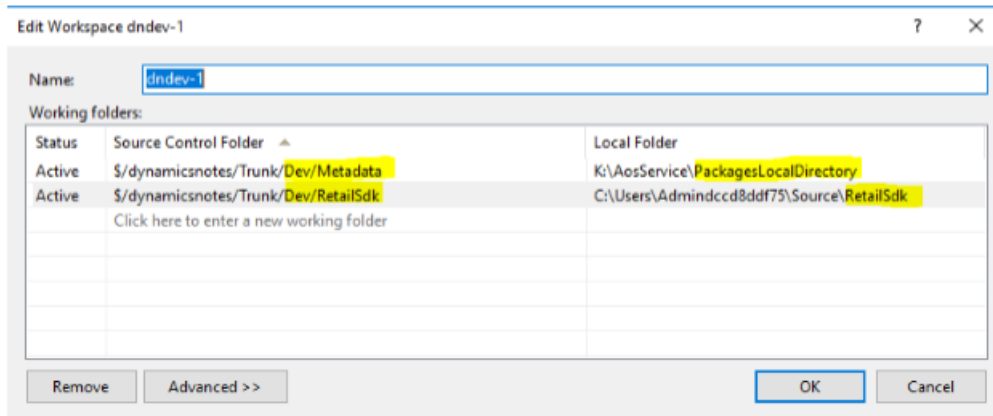
You should also change the version of the Commerce customization. This version should differ in the Dev, Main, and ProdRel1 branches. Either change the Customization.settings file, or add a new global.props file in the **RetailSdk\BuildTools** folder. For example, you can number Dev as 1.0.0.x, Main as 1.0.1.x, and ProdRel1 as 1.0.2.x.

## Prepare the development environment

You can now prepare the development environment for Commerce development tasks. The development environment will map the code locations for both X++ and the Retail SDK in the Dev branch to local folders. The Metadata folder (X++) must be always mapped to the PackagesLocalDirectory folder. The location of the RetailSdk folder must follow these guidelines:

- The location should be somewhere inside the local user's folder.
- The file path of any file is limited to 256 characters. Therefore, use a short path for the root of the Retail SDK. For example, you can use `c:\users\\Source\RetailSdk`.

To map X++ and the Retail SDK, you must edit the current workspace. Select **Pending Changes > Actions > Workspaces**, and update the current workspace so that it resembles the following illustration. As was previously mentioned, you should map the Metadata folder of the branch to the PackagesLocalDirectory folder and the RetailSdk folder to a short folder of your choice.



The download of the files can take a few minutes.

Regardless of whether there are customizations in the code branches, the following steps prepare your development box so that you can write and run code. Some steps are optional, depending on the customizations that are planned.

1. Install your favorite development tools. For information about one automated script, see [Auto-Installing most needed dev tools in 5 mins with Chocolatey](#).
2. To help reduce the compile time, exclude the code folders from Microsoft Windows Defender.
3. If there is already code in the **Dev/Metadata** folder, build all Commerce models. Select all the models, and then select **Database sync**.
4. To speed up the development experience, switch to Microsoft Internet Information Services (IIS). For instructions, see [MSDyn365FO. How to switch from IIS Express to IIS on development VM](#). This step can be done only on the Tier 1 VM where you have administrative privileges (cloud-hosted environment).
5. Optional: Restore a recent copy of a production database that has good data.
  - a. Rename the existing database **AxDB\_Orig**.
  - b. In Microsoft SQL Server Management Studio, restore the .bak file. (If a .bacpac file exists, follow the steps in [Copy a database from Azure SQL Database to a SQL Server environment](#).)
  - c. In Visual Studio, refresh the model store.
  - d. In Visual Studio, do a full build if the source and destination environments of the database are on different versions.
  - e. In Visual Studio, run a full database synchronization.
  - f. Make sure that the Batch service is running.
  - g. Run the Environment reprovisioning tool. (Find the latest version in the LCS Asset library, and deploy it by using the **Maintain** function.)
  - h. Verify that the tool succeeded. The following query should show the URLs of all local development



machines that were updated.

```
select * from dbo.RETAILCHANNELPROFILEPROPERTY where ISSYSTEMRECORD = 1
```

- i. In Commerce, run the **Initialize Commerce Scheduler** job to delete old data.
6. Make sure that you can sign in to Commerce by using your user account. If you aren't the Admin user in the production database, run the Admin provisioning tool to take ownership. (This tool is in the **PackagesLocalDirectory/bin** folder.)
7. Verify that Commerce Data Exchange (CDX) data synchronization works. In Commerce, go to **Download sessions**. You should see many applied sessions. If you don't see them, select job **9999**, and run it.
8. Install TypeScript version 2.2.2 from <https://www.microsoft.com/download/details.aspx?id=48593>.
9. Do a full build of the Retail SDK from a command prompt.
  - a. Open an MSbuild command prompt for Microsoft Visual Studio 2015 as an administrator.
  - b. Change the directory to the location of your Retail SDK on the local VM.
  - c. Type **msbuild**, and then press Enter. The build should succeed.
10. Add the development/sample Retail Modern POS (MPOS) certificate to the local machine's trusted root certificate store: ...\**RetailSDK\BuildTools\ModernPOSAppxSigningCert-Contoso.pfx**. Set the password to an empty string.
11. Install MPOS or MPOSOOffline by running the installer at ...\**RetailSDK\References\YourCompany\Contoso.ModernPOSSetupOffline.exe**. You must complete this step one time to deploy the ClientBroker files.
12. In Visual Studio, open **ModernPOS.sln** (as an administrator), and do a full rebuild.
13. Press F5 to start MPOS in the debugger.
14. In Commerce, open the **Channel profiles** page, and copy the Commerce Scale Unit URL for the default channel profile.
15. Open a browser window, and paste the URL into the address bar. You should be able to browse to your local Commerce Scale Unit.
16. In Commerce, add external user credentials to any worker (for activation), save the password, and don't allow a password reset on first sign-in.
17. In Commerce, run job **1060 (AX/Distribution schedule)**.
18. Activate MPOS by using the same Azure Active Directory (Azure AD) user that you added in step 16. Paste the Commerce Scale Unit URL, select a store and a register, and finish the activation.

You should now be able to run MPOS in the debugger from your local sources.

The process of preparing a development environment is now completed. At this point, any extension code (X++, Commerce runtime [CRT], Commerce Scale Unit, channel SQL, and POS) can be written, debugged, tested, and submitted to Azure DevOps.

## Optional: Deploy a second build environment for a different release branch

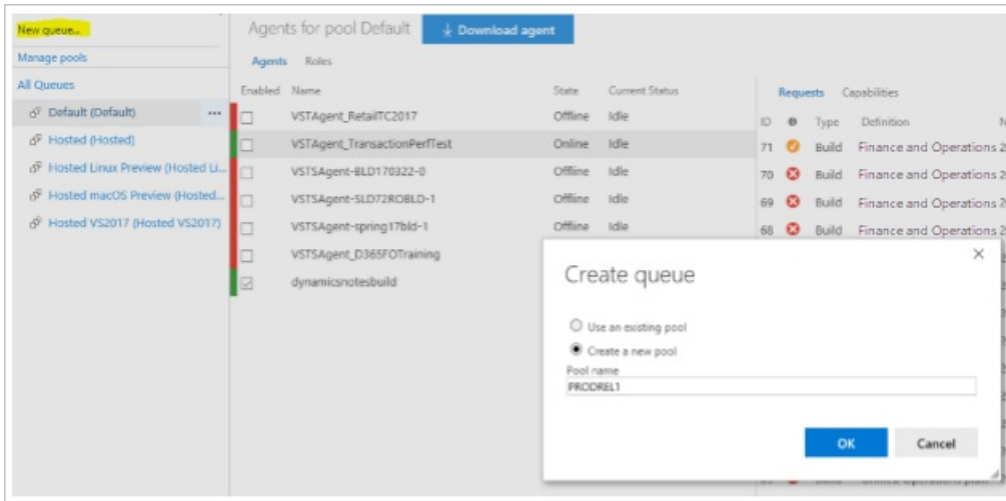
If you must maintain multiple releases at the same time, you must create deployable packages from different code branches (for example, Main2 or Main3, and/or ProdRel1 or ProdRel2).

The steps to set up a second build environment are the same as the steps for the first build environment. At this point, an Azure DevOps project, and the link between the LCS project and the Azure DevOps project, already exist.

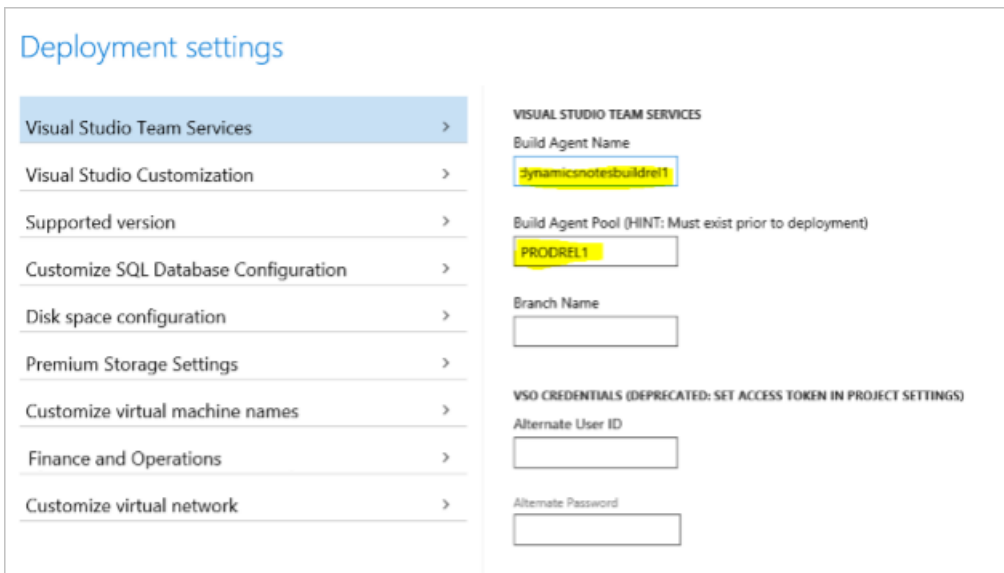
To separate the build environments, we recommend that you create a new Azure DevOps agent queue for the release branch. Although there are ways to share an agent queue (and its build environment) for multiple branches, this approach can be tricky.

Currently, the build environment must be on the same platform and binary hotfix version as the target environment during deployment. Otherwise, LCS might reject the deployable package because of version incompatibility.

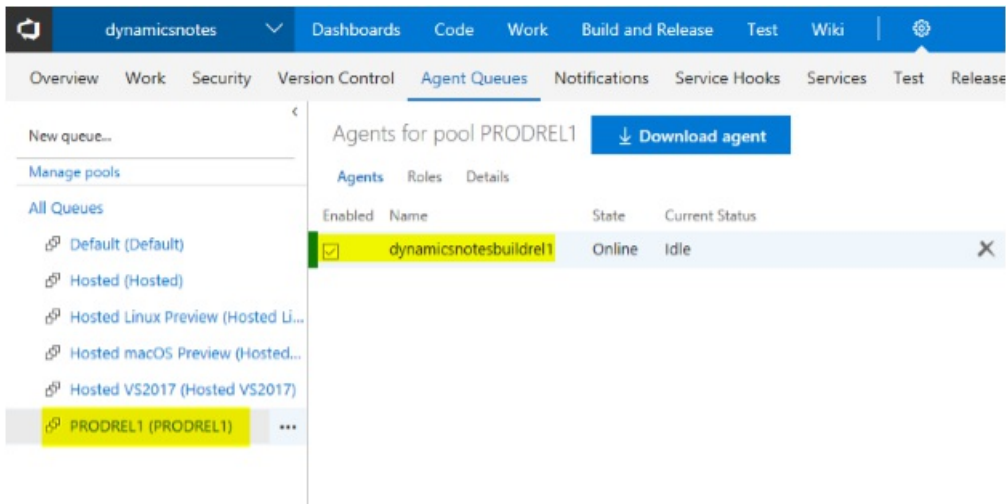
First, create a new Azure DevOps agent queue.



When you deploy from LCS, use **PRODREL1** as the name of the agent pool.



Next, on the **Customize virtual machine names** tab, enter a unique name, and then deploy the new build. The process of deploying a new build and creating a new agent queue can take a couple of hours.



## Prepare the build definitions

After you complete the steps earlier in this topic, you should have one build definition and two agent queues, and each agent queue should have one agent. To build different branches, you must configure the build definition differently. Therefore, you must clone the build definition.

However, before you clone the build definition, you must add the Retail SDK into the build, so that you don't have to complete this step twice. To edit the existing build definition, which is named **Unified Operations platform - Build Main**, follow the steps in [Integrate the Retail SDK with the continuous build system \(Azure DevOps\)](#) to integrate the Retail SDK into the metadata build of the Main branch.

If you had multiple build branches and environments, just clone the build definition, and name the new build definition so that it's clear which branch it's for. (The clone feature is available in the Azure DevOps portal). Select the new agent queue that you created, and change the following paths in any build steps or source mappings. (In the paths, change **Main** to **ProdRel1**.)

- Source mappings
- Retail SDK build step
- Retail SDK copy binaries step
- Build the solution step (X+ + build)
- Retail SDK copy packages step

## Tips

- You can speed up an official build by making these changes in the **Variables** section of the build definition:
  - Set **DeployReports** to 0.
  - Set **SkipSourcePackageGeneration** to 1.
- Change the version of the Commerce customization in each branch. The version should be different in the Dev, Main, and ProdRel1 branches. Either change the Customization.settings file, or add a new global.props file under the **RetailSdk\BuildTools** folder. You can use any kind of numbering for the file name. For example, you can number Dev as 1.0.0.x, Main as 1.0.1.x, and ProdRel1 as 1.0.2.x.
- For efficiency, shut down build or development environments when they aren't being used.
- If you're using cloud-hosted Tier 1 development environments (where you have administrative privileges), you can switch from IIS Express to IIS. Using IIS for running all web application is more robust, more performant, and avoids the switching. For details, see [MSDyn365FO. How to switch from IIS Express to IIS on development VM.](#)

- For prototyping purposes, a developer might want to change the Retail SDK on a development VM without using Azure DevOps source control. Always keep the original Retail SDK untouched, and make a copy that you can work in temporarily. In that way, you can take the unchanged Retail SDK into your mirror branch later, if it's required.
- Currently, a build environment must be on the same platform and binary hotfix version as the target environment.

## Additional resources

[Update code and environments for Retail projects](#)

[Testing and performance issues](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Update code and environments for Commerce projects

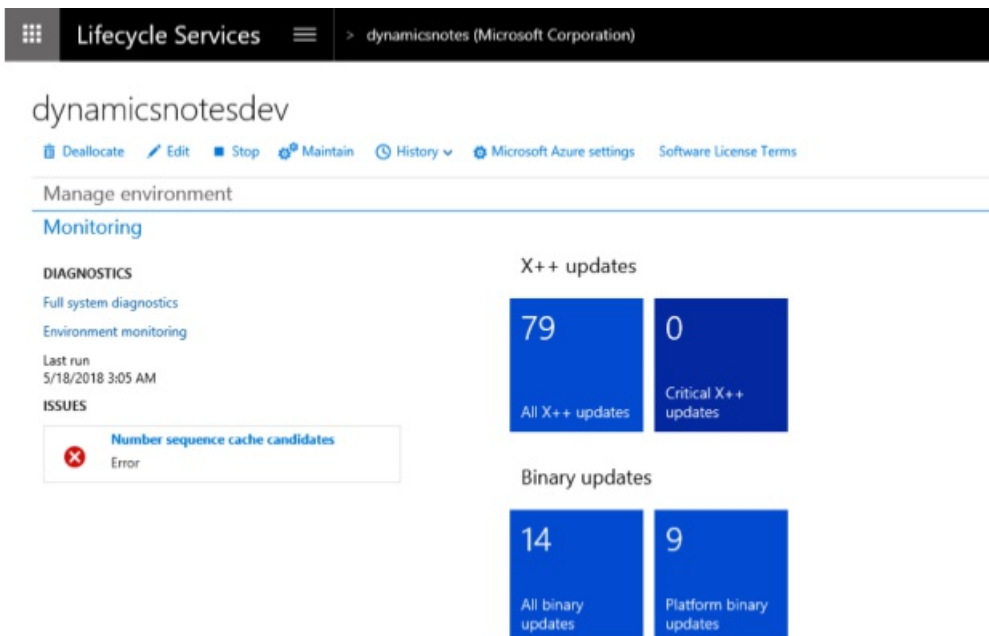
2/18/2021 • 12 minutes to read • [Edit Online](#)

An environment can be updated by updating either its data or its code.

There are multiple ways to update the data. For examples that show how to get data into an environment, see [Data entities and data packages](#).

When you update an environment, you should also consider moving the whole database. This approach lets you quickly and easily duplicate the data from one environment to another.

Other updates are code updates. The environment page in Microsoft Dynamics Lifecycle Services (LCS) tracks the updates that have been applied and the updates that must be applied. The following illustration shows an environment that has 79 outstanding X++ fixes, 14 outstanding binary updates, and nine outstanding platform binary updates.



Platform code is at a very low level, and no Microsoft Dynamics 365 Commerce features are implemented in the platform. Therefore, stand-alone platform binary updates don't require that you retest any Commerce-specific code. Examples of features that are implemented in the platform are the Data Import/Export Framework (DIXF) and the batch framework.

Binary updates or hotfixes include dynamic-link libraries (DLLs), scripts, and channel SQL schema changes. All channel-side hotfixes are released together as a binary update/hotfix. Because binary updates are DLLs, they are cumulative. For example, if you download a binary update on Friday, you automatically receive all binary hotfixes from Monday through Thursday.

If the code merge is done correctly, the version of a binary hotfix that you take matches the version of the Microsoft-version.txt file in the Retail software development kit (SDK). Typically, binary updates are also linked to the latest platform. Therefore, when you take binary updates, you must stay up to date with the platform. Platform updates help increase the stability of the platform, and they affect build environments and test efforts to some extent.

Application updates or hotfixes are delivered in X++ source code. Therefore, they aren't for the channel side but for the Microsoft Dynamics 365 side (they are either Commerce-related or not Commerce-related).

Note that some updates require both an application update and a binary update. For hotfix recommendations, see the next section.

Third-party packages resemble application packages, but they are developed by other people. For more information about how to use independent software vendor (ISV) packages, see [Manage Runtime Packages](#).

## Updating data by restoring the database

In one useful and typical operation, the whole database is moved from one environment to another. For example, you might move the production database to development environments when you're preparing to develop additional features. Alternatively, you might move the golden setup database to the production database as part of the go-live process.

For more details, see [Copy Database From Azure SQL to SQL Server](#). If source and destination environments don't have the same binary version, you should also do either a build and a database synchronization (for a development environment), or a deployment (for a sandbox or production environment).

Every time that a database that has been moved from a different environment is restored, specific links in the database can be broken. The Environment reprovisioning tool fixes all these broken links for the default database group, regardless of type of environment that is used. The general guideline is that if the database comes from a different environment, the Environment reprovisioning tool must be run.

In many cases, you should reset the Commerce scheduler after you update the database.

After you've restored the database, follow these steps.

1. Either do a build and a database synchronization, or deploy the deployable package.

### NOTE

If you have table extensions that include data, you must have the metadata for those extensions in the environment. Otherwise, you can lose data, because columns and tables might be dropped.

2. Make sure that the batch service is running.
3. Run the Environment reprovisioning tool. (Find the latest version in the global Shared asset library in LCS, and then deploy it by using the **Maintain** function.)
4. Verify that the tool succeeded, the Commerce channel profile is up to date with the correct URLs, and the data synchronization jobs for the Default data group succeeded.
5. In Commerce, run the **Initialize Commerce scheduler** job (select to delete old data). This step assumes that all Commerce Data Exchange (CDX) configuration changes are automated by using a resource file. If CDX configuration changes aren't automated, and if tables, subjobs, and jobs are manually created in the Commerce channel schema, don't select the option to delete the existing configuration. We recommend that you automate CDX configuration changes.

## Taking updates frequently

If your project is more than a few weeks from go-live or the final user acceptance testing (UAT), we recommend that you take all hotfixes (binary, X++, and platform) on a regular schedule. Specifically, we recommend that you take all hotfixes one time per month. The more often you perform this task, the fewer issues you should experience, because the code churn of the hotfixes is smaller. If you perform this task often, it will take significantly less than eight hours.

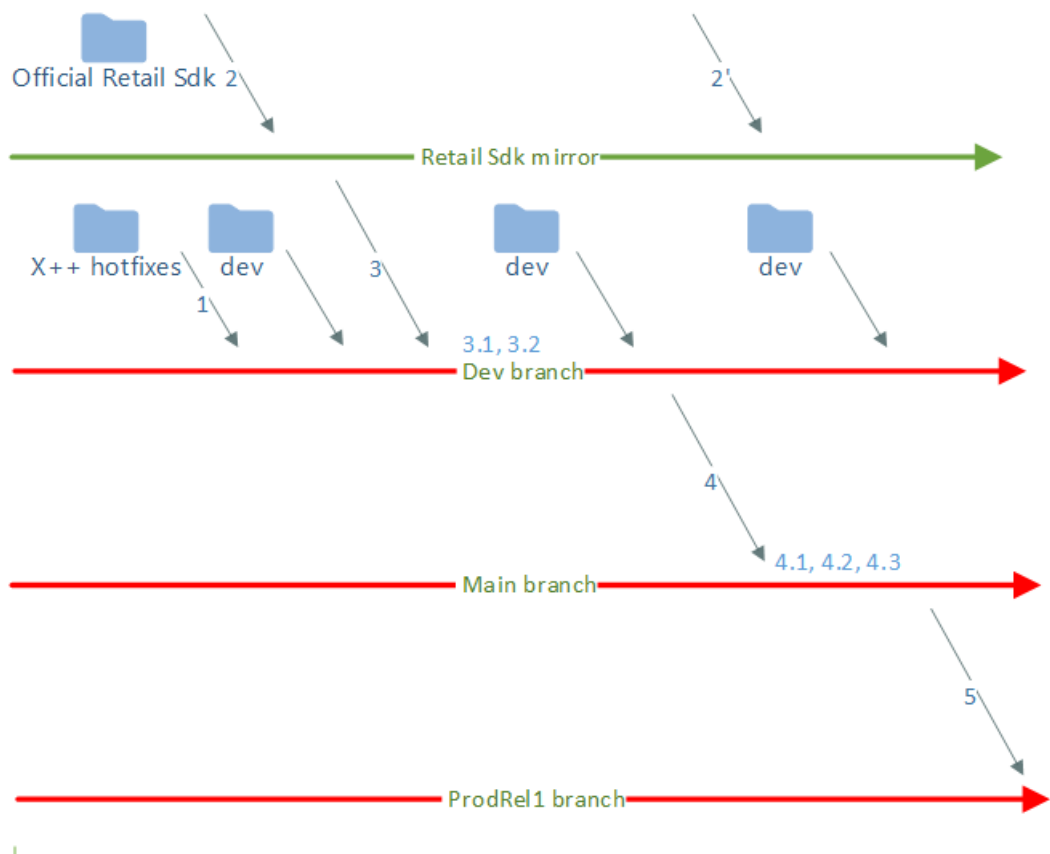
We recommend that you not pick and choose hotfixes, because this approach is more likely to cause errors and probably isn't worth your time. If you have 500 or 1,000 outstanding hotfixes, you should consider whether you're really ready to go-live. The quality of the product will be higher if the count on the update tiles in LCS is very low (fewer than 100 application fixes and fewer than ten binary fixes).

After you take new hotfixes, the results of a previous round of UAT become less meaningful. Therefore, it's crucial to retest again. The number of files that changed determines how extensive the testing must be. If hotfixes are frequently taken, especially during the implementation phase, the number of new files isn't very large, and the retesting effort is manageable.

Another approach is to take all hotfixes frequently and run only part of the UAT. Then, the next time that new hotfixes are taken, run a different part of the UAT. Run the different parts of the UAT in a circular manner. Before go-live, you should do a full UAT run.

## The flow of code changes through branches and environments

Just as the branching strategy is dictated by project, team, or other constraints, your project has flexibility about how the changes are propagated through the branches. The following illustration shows an example of the process. However, this example might be too simple for some projects and too complex for other projects. The important point is that a project should have a plan. Different persons in the team will have different responsibilities (development, deployment, code merges, sign-off, and so on), and the role ownership should be clearly defined.



- 1 ..... X++ developer gets and submits new X++ hotfixes
- 2 ..... Retail developer gets the Retail Sdk update from a binary update
- 3 ..... Retail developer code-merges the new Retail Sdk changes
- 3.1 ... Retail developer shares binary update and Retail package for other devs
- 3.2 ... other developers update their environments
- 4 ..... specific changes are merged from Dev to Main
- 4.1 ... deployable packages being build from Main branch, added to LCS
- 4.2 ... test environments are being deployed for testing
- 4.3 ... production is being deployed
- 5 ..... final code is merged to ProdRel1 branch

### Steps 1–3: Obtain and apply updates

For full details about steps 1 through 3 (taking updates), see the [hotfix and deployment cheat sheet](#). If the branches are set up in the same manner that is shown in the preceding illustration, you should do this work in the Dev branch.

### Steps 3.1–3.2: Keep development environments up to date

You don't have to have a build environment for the Dev branch. In fact, a build environment for the Dev branch isn't usually required. You just have to coordinate the packages that should be deployed to keep the version correct.

After you download binary updates and platform updates, you can deploy them via LCS package deployment.

For the X++ code, developers just synchronize the Metadata folder and do a full build and database synchronization.

If major new changes have been checked in by other members of the team (for example new files, configuration changes, or a new Retail SDK), it isn't enough to synchronize and build the new files. Remember that a few web applications that are installed on the developer machine won't be updated through a compilation. Those web applications must be deployed. Use the LCS package deployment to deploy the commerce package that can be produced at an MSBuild command prompt. For smaller code changes, new package deployments aren't



required in order to keep the dev environments in sync if the incremental changes are dropped to the install locations.

## Environment change history

Environment name `dynamicsnotesdev`

Organization name Microsoft Corporation

\* All times are displayed in your local time zone (currently UTC -07:00). Please be aware that other users will see the time localized into their time zones.

### ENVIRONMENT ACTIVITY LOG

Name	Type	Created Date	Completion Date	Status
RET_Dev_2018.6.21.1	Retail deployable package (combined)	6/21/2018 5:32 PM	6/21/2018 5:44 PM	Signed off
BIN_06202018	Platform and application binary package	6/20/2018 5:20 PM	6/20/2018 6:15 PM	Signed off

### Step 4: Move changes from the Dev branch to the Main branch

In this example, the Dev and Main branches have been separated to provide an opportunity to "leave some unwanted changes behind." Although this approach isn't required, it's a good option to have. Microsoft Visual Studio makes the process of moving the code from Dev to Main easy. You can select a range of changes, select all or individual changes, and merge those changes. To keep the process simple, you can have some type of a code freeze in the Dev branch. Then, when you're satisfied with the quality, you can merge all changes. There is no reason to treat X++ differently than the Retail SDK. They reside next together in each branch, because they are dependent on each other.

### Steps 4.1–4.2: Update test environments

Use your build environment to produce officially built packages from the code in the Main branch.

Build Definitions / Finance and Operations - Build ✓ passing [Queue](#)

[Summary](#) [History](#) [Deleted](#)

**Details**

Repository: [dynamicsnotes](#)  
Default queue: [Default | Manage](#)  
Queue status: [Enabled](#)  
Last updated by: [\[redacted\]](#) | Tuesday, June 19, 2018 4:26 AM

**Branches**

[\\$/dynamicsnotes/Trunk/Main](#)

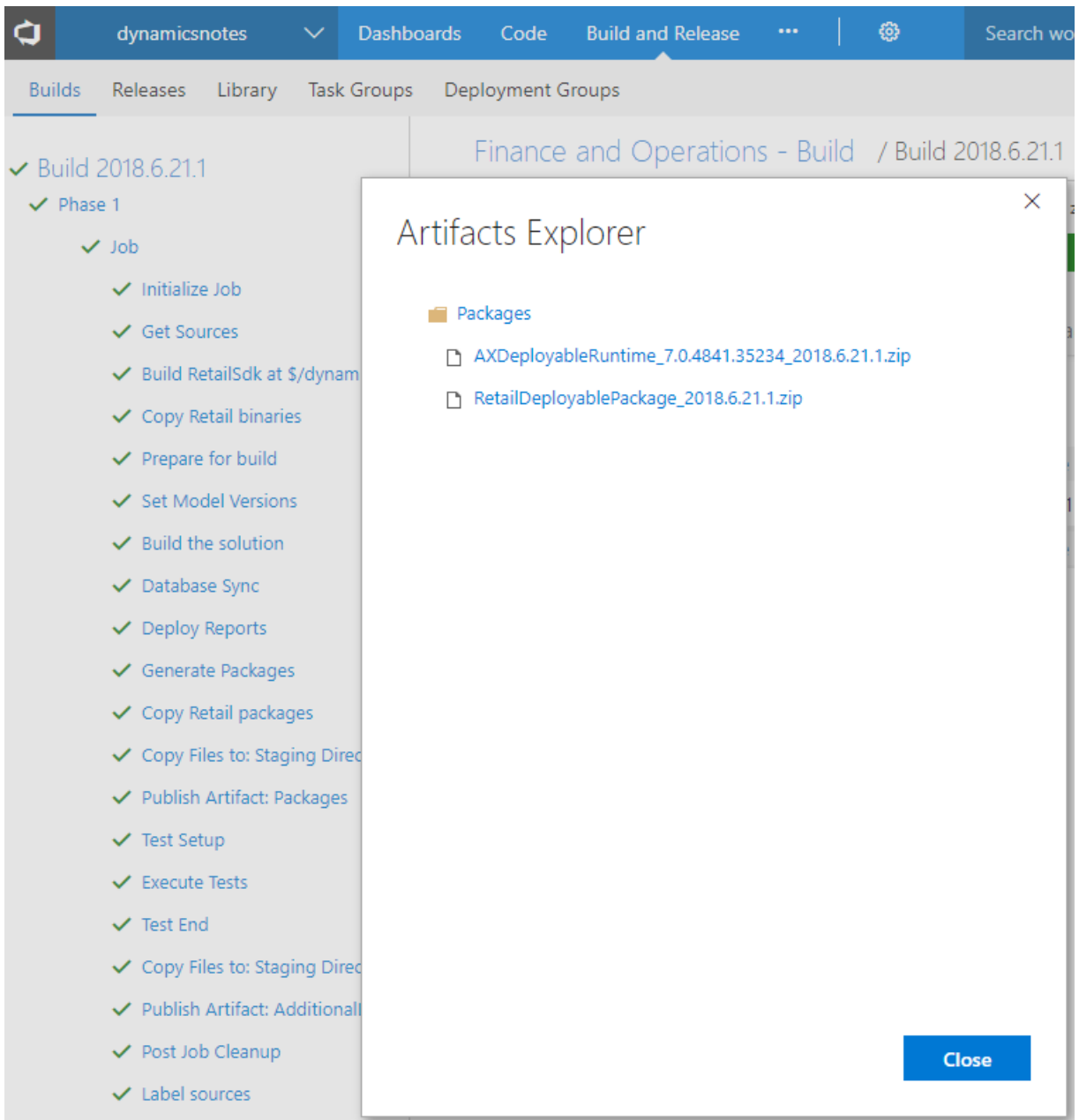
**Queued & running**

No builds queued or running at the moment

**Recently completed**

[#2018.6.21.1](#) ✓ succeeded [\\$/dynamicsnotes/Tru...](#) Andreas Hofmann

When the build is completed, find the packages that were built, download them, and rename them according to your naming conventions.



Then upload the packages to the LCS Asset library.



# Asset library

## Select asset type

- BPM artifact (0)
- Cortana intelligence application (0)
- Data package (0)
- Database backup (0)
- Downloadable VHD (0)
- Dynamics 365 Retail SDK (0)
- GER Configuration (0)
- Localized financial report (0)
- Marketing asset (0)
- Model (0)
- Power BI report model (0)
- Process data package (0)
- Software deployable package (3)
- Solution package (0)

## Software deployable package files

		IMPORT	Copy	SAVE TO MY LIBRARY	VERSIONS	MERGE
✓	Name	Valid	Version	Scope	Status	
	RET_Main_2018.6.21.1	✓	1	Project	Draft	
	AX_Main_2018.6.21.1	✓	1	Project	Draft	
	BIN_06202018	✓	1	Project	Draft	

Finally, deploy the packages to your test environments.

## Environment change history

Environment name dynamicsnotesUAT

Organization name Microsoft Corporation

\* All times are displayed in your local time zone (currently UTC -07:00). Please be aware that other users will see the time localized into

### ENVIRONMENT ACTIVITY LOG

Name	Type	Created Date	Completion Date	Status
RET_Main_2018.6.21.1	Retail deployabl...	6/21/2018 7:47 PM	6/21/2018 8:00 PM	Signed off
AX_Main_2018.6.21.1	Application depl...	6/21/2018 6:41 PM	6/21/2018 7:29 PM	Signed off
BIN_06202018	Platform and ap...	6/21/2018 5:33 PM	6/21/2018 6:26 PM	Signed off

### Step 4.3: Deploy packages to the production environment

When all the required tests are passed, you're ready to deploy the same packages to production. After the packages have been deployed and validated in a Tier 2 or higher environment, you must mark them as Release Candidates in the LCS Asset library. You must then plan the deployment and submit it via the LCS environment page.

There are many considerations when you update a production environment, such as downtime, downtime mitigation, data migration, store updates, and mass deployment. It's very important that you have a plan of all the steps that are required for an update, because Commerce projects usually require more than just

deployment. For some additional considerations, see the "Tips" section of this topic.

It's assumed that the planning for go-live was started much earlier. For more details, see [Implementation lifecycle](#).

### Step 5: Merge the code from the Main branch to the ProdRel1 branch

Immediately after deployment to production, and before any new feature work is added to the Main branch, you should take a snapshot and move it to the ProdRel1 branch. The steps are the same as in step 4. You don't have to select individual changes. Instead, just merge all changes up to the last code change set that was submitted to the Main branch.

## Update build environments

You should always deploy binary updates and platform updates by using LCS package deployment.

Finance and Commerce customization packages should not be deployed to a build environment.

## Environment change history

Environment name `dynamicsnotesbuild`

Organization name Microsoft Corporation

\* All times are displayed in your local time zone (currently UTC -07:00). Please be aware that other users will see the

### ENVIRONMENT ACTIVITY LOG

Name	Type	Created Date	Completion Date
BIN_06202018	Platform and ap...	6/20/2018 6:35 PM	6/20/2018 7:48 PM

## Compare LCS tile counts

Environments that are used for work of the same release should also have the same LCS tile counts. Here are some reasons why the tile counts might differ:

- The same deployable packages haven't been deployed and applied. You can troubleshoot this issue by inspecting and comparing the LCS deployment history.
- The scheduled task that collects the version information from an environment hasn't been run yet. For development environments, you can force the "LCSDiagnosticsCollector" schedule task to run.
- The counts for the build environment's application updates don't match because X++ packages aren't deployed on build environments. Binary and platform counts should be correct.
- The difference might be intentional. For example, a developer might work with the next version, whereas the rest of the team is still working with a different release. Alternatively, one development environment might be kept on an older version in case a production hotfix must be developed, and that production environment uses an older version than current development environments.

Notice that after you've finished updating an environment, the tile counts for the available updates are significantly lower than they were when you started.

# dynamicsnotesdev

Deallocate Start Maintain History Microsoft Azure settings Software License Terms

Sql server login	xrefdbuser	*****
Sql server login	axdwadmin	*****
Sql server login	axdwruntimeuser	*****

## AZURE CONNECTOR

Name	Azure subscription ID	Azure region
[REDACTED]	[REDACTED]	West US 2

## CLOUD SERVICE NAME

dynamicsnotesdev-DEV-D2-c7bfba39e7ddd50e

## Monitoring

### DIAGNOSTICS

Full system diagnostics  
Environment monitoring

Last run  
6/21/2018 3:15 AM

### ISSUES

✖ **Number sequence cache candidates**  
Error

### X++ updates

20	0
All X++ updates	Critical X++ updates

### Binary updates

4	3
All binary updates	Platform binary updates

## Move to a new version

To upgrade to a new version (such as 7.2 to 7.3 or 7.3 to 8.0), you must deploy a new environment. You must also run a code upgrade and a database upgrade, if these upgrades are applicable. For more details, see [Code migration home page](#).

## Tips

- Decide on a good package naming convention for names in the LCS Asset library and for the names of zip packages that are downloaded. In this way, you can more easily determine what package you've deployed and where it came from. Avoid spaces in package names. Here is an example of a naming convention:
  - **Platform update packages:** PUXX\_MMDDYY, where XX is the number of the platform update
  - **Binary update packages:** BIN\_MMDDYY
  - **X++ update packages:** APP\_MMDDYY
  - **Built X++ deployable packages:** AX\_BRANCH\_VERSION, where BRANCH is an appropriate branch name, and VERSION is the Microsoft Azure DevOps version string
  - **Built Retail combined package:** RET\_BRANCH\_VERSION, where BRANCH is an appropriate branch name, and VERSION is the Azure DevOps version string
- Whenever you start a new item of work, use the **Get latest** option in the Visual Studio source code

explorer.

- Any code submissions should use correct and detailed comments that describe the change sets.
- Production go-live procedures are important. You should consider including the following items on your Go-live checklist. Verify your Go-live checklist in a mock go-live or UAT environment. This list isn't exhaustive.
  - After deployment, does LCS show the expected deployment history together with the correct package names?
  - After deployment, do the LCS environment page and Commerce show the correct and expected version numbers?
  - Can Modern Point of Sale (MPOS) offline mode be used during downtime of Commerce? Package deployments will cause downtime. If MPOS offline mode can be used, have you tested the procedure? (To test the procedure, go offline, deploy, go online, synchronize offline transactions, and update MPOS.)
  - Does the Environment reprovisioning tool have to be run (if a database has been moved)?
  - Batch jobs for CDX synchronization must be reenabled by setting them to **Waiting**.
  - The "Initialize Commerce scheduler" job should be run.
  - Does other data have to be set up in addition to the deployable packages (for example, screens, buttons, receipt layouts, the Microsoft Azure Active Directory setup, Commerce shared parameters, the tax configuration, other batch processes, and DIXF recurring jobs)?
  - Is a synchronization of the CDX data jobs required?
  - Is a full synchronization of CDX data jobs required?
  - Does a deployment require that store components also be updated?
  - If the store components had to be updated, do they show the new version numbers?
  - Are the correct experts available during the deployment (for example, partners, ISVs, and customers)?

## Additional resources

[Set up new environments, Azure DevOps, and branches for Commerce projects](#)

[Testing and performance issues](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Testing and performance issues

2/18/2021 • 5 minutes to read • [Edit Online](#)

This document describes practices and tools that are related to functional testing, performance testing, and performance troubleshooting.

## User acceptance testing

The main purpose of user acceptance testing (UAT) is to verify that specific business scenarios work as you expect. The testing should include not only the customizations, but also out-of-box Microsoft functionality and non-happy-path testing. The goal is to catch anything that doesn't work correctly before you go to the production environment.

For the best results, the UAT environment should be a Tier 2 through Tier 5 environment, not a development environment (Tier 1).

If a development environment is used, there are scenarios where a developer who uses the same environment can cause errors (for example, uncommitted source changes or debugger attached errors). The switch between Microsoft Internet Information Services (IIS) Express and IIS can also cause issues. Additionally, because there is no way to know exactly what has happened on a development machine, Microsoft support for a Tier 1 environment is very limited.

A production environment can be used for UAT (for example, as a "dry run" for go-live). However, if you require access to the database for any reason, you must go through deployment support engineer (DSE) service requests. Therefore, this approach might not be efficient. Additionally, the production environment isn't available for a long period before the planned go-live date.

The UAT should be done after you deploy officially built deployable packages. It should not be done on packages that are manually built in Microsoft Visual Studio. The reason is that there is no way to prove what code changes were included in a manually built package. Only an official build system provides assurance and an audit trail of the exact changes that are in a specific build.

If you use Modern POS/Cloud POS, make sure that you use the correct user roles. You should test by signing in as both a manager and a cashier who has lower privileges.

## Performance

### Channel performance

In some cases, channel performance might not be as good as you expected. Poor performance is often caused by the following factors. This list is in order from highest to lowest impact.

- Additional custom calls to Commerce Scale Unit. If you extend the product with additional calls, performance often decreases significantly. Not only is there a possibility of additional processing, but the network latency must also be considered. We recommend that you try to avoid any additional Commerce Scale Unit calls. Often, you can accomplish the same tasks by using extension properties and extending existing Commerce runtime (CRT) handlers or triggers.
- Additional channel database extensions. Make sure that your custom SQL is efficient and uses correct indexes.
- Multiple runs of the same custom or built-in CRT SQL queries. If this approach is too expensive, caching in the CRT request handler can be applied, as appropriate.

For more details, see the [Commerce for IT pros and developers](#) topics.

When you investigate store performance, follow the suggestions in [Retail Channel performance investigations](#).

### Using telemetry data to find performance issues

If you must troubleshoot the performance (especially slow SQL queries or SQL deadlocks), the environment diagnostics page in Microsoft Dynamics Lifecycle Services (LCS) shows valuable telemetry data. You can use this data to find potential performance issues in code, configuration, or design. For more details, see [Monitoring and diagnostics tools in Lifecycle Services](#). That information should help you determine why some batch processes or form loads are slow.

### Performance testing

Typically, when you test the performance of a system, you should focus on components where there is competition for many shared resources. These resources might differ for different projects, customers, or requirements.

Here are some of the reasons why bottlenecks can occur:

- Resource-intensive calculations, such as statement posting, change calculations for channel data synchronization, warehousing operations that involve a large product assortment, and MRP (Material Requirements Planning) runs
- Complex business logic for multiple terminals or stores that run on a few Commerce Scale Unit
- Integrated third-party systems (integrated from either Commerce or Commerce Scale Unit)
- Real-time transaction services that are frequently called from Commerce Scale Unit.
- Non-standard or extended standard functionality (for example, extended statement posting that uses a custom WHS code)

In general, default and non-real-time POS operations aren't considered bottlenecks because they have their own dedicated resource: the computer that the POS is installed or running on. Performance issues are typically caused by the business logic or "chatty" calls to Commerce Scale Unit.

Ideally, performance testing should be done after some initial optimizations have already been completed by using the information earlier in this topic. If the system doesn't perform well for a single user or process, it won't perform well for concurrent users or processes. For more information, see [Retail Channel performance investigations](#). Additionally, in the Finance documentation, search for "PerfSdk" or "Trace parser."

Because every project is different, it's difficult to give a general answer about the exact performance tests that must be run. For example, if the count of transaction sales lines is low (less than 100,000 per day for all stores), and if no custom extension code has been added for statement posting, a performance test should not be required for posting. However, if the count of sales lines is substantially higher, or if major custom changes have been added, a performance test for posting is a good idea.

Usually, the hardware capabilities of every environment differ. However, performance issues can usually be reproduced in other environments if the code and data are similar. We don't recommend that you use a production environment for performance testing. A good practice is to use the same data in development, test, and production environments. The development environment can then be used to work on and verify a fix. Because many performance-critical code paths are data-dependent, the same issues might not be seen for a Contoso sample database.

After you've completed a performance fix, you should verify the fix in a test environment. Deploy an officially built package to the test environment, and if the issue is fixed, mark the package as a release package.

Any fix of a larger performance issue should be followed by a new performance test. Often, a large issue masks other smaller issues. After the top issue is resolved, the next issues can be found and worked on until the performance meets the customer's expectations.



## Additional resources

[Set up new environments, Azure DevOps, and branches for Commerce projects](#)

[Update code and environments for Retail projects](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Commerce architecture overview

2/18/2021 • 13 minutes to read • [Edit Online](#)

## NOTE

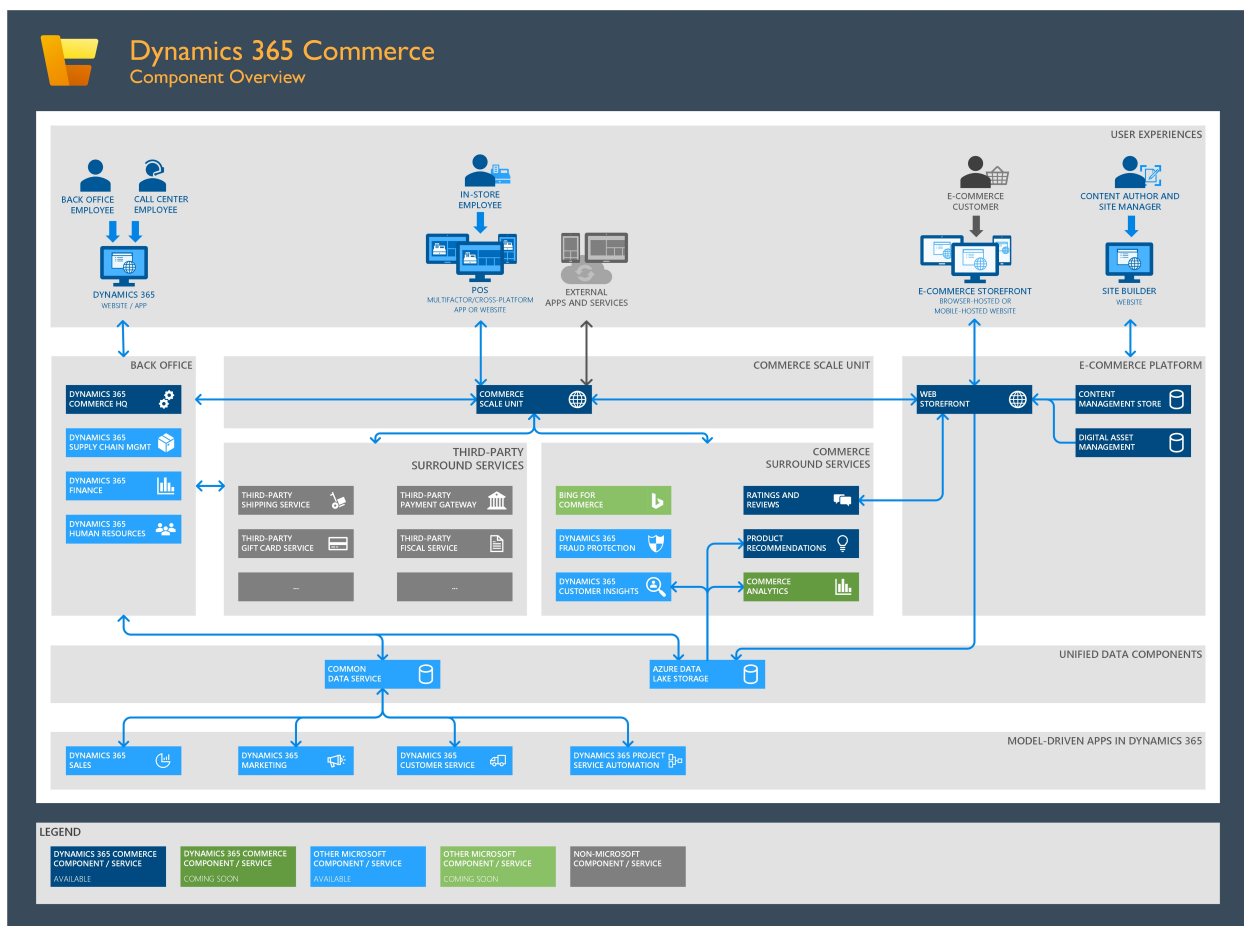
Effective November 2020:

- Common Data Service has been renamed to Microsoft Dataverse. For more information, see [Power Automate Blog](#).
- Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. For more information, see [Terminology updates](#).

This topic will be updated soon to reflect the latest terminology.

This topic provides an overview of all components in the Microsoft Dynamics 365 Commerce ecosystem, including integration points to the suite of Dynamics 365 products.

The following illustration shows an overview of Dynamics 365 Commerce components.



## Architecture benefits

### Omni-enabled headless commerce engine

The Commerce Scale Unit hosts the headless commerce engine. It serves as the central integration point for all commerce business logic, and powers a complete omni-channel solution across physical and digital stores. The Commerce Scale Unit is built by using a portable architecture, and allows for flexible hosting options across cloud, edge, and hybrid topologies.

The headless commerce engine powers all native Dynamics 365 Commerce channels, including in-store and e-commerce channels. It also serves as the single integration point for third-party channel solutions. Therefore, those solutions can take advantage of the power of Dynamics 365 Commerce business logic and integration with other commerce-related services that are provided by Microsoft and independent software vendors (ISVs).

### **Interconnected business processes**

The platform that is shared among the various Dynamics 365 business applications, such as Dynamics 365 Commerce, Dynamics 365 Supply Chain Management, and Dynamics 365 Finance, provides a set of interconnected business processes that users can immediately benefit from. All back-office capabilities across these applications are built on the same web experience and data stores. Therefore, there is a seamless flow of business processes across various functions in the organization, but custom integrations across applications and services aren't required. The out-of-box integration between the headless commerce engine and the back office further expands the coverage of these interconnected business processes across the back office and commerce channels.

### **Unified data**

Dynamics 365 Commerce provides a unified data solution through out-of-box integrations with [Dataverse](#) and [Azure Data Lake Storage](#). Integrations and data sharing across Dynamics 365 business applications such as Dynamics 365 Sales and Dynamics 365 Marketing are supported through Dataverse. Transactional data in Data Lake Storage is used to power various analytics and insight scenarios in the Dynamics 365 Commerce solution. However, it can also be used by any third-party software integration.

### **Powered by AI and analytics**

Because of the accessible, persistent, up-to-date, and unified organizational data that is available in Data Lake Storage, the whole organization has a "single source of truth" that analytics, artificial intelligence (AI), and machine learning (ML) can be applied on top of. In this way, the organization can derive insights and get key performance indicators (KPIs) that can be used to optimize and automate business processes across all channels.

## **Component overview**

### **User experiences**

#### **Dynamics 365**

Dynamics 365 is a collection of applications that together provide comprehensive and flexible enterprise resource planning (ERP) solutions for medium to large businesses. It provides an extensible framework and ecosystem that can be tailored to customer-specific requirements via an extensive set of partners. Dynamics 365 applications provide capabilities for their target business segments. They also take advantage of each other, and other Microsoft services and offerings, to provide solutions that help run customers' complex businesses.

#### **Modern POS**

Modern POS is a cross-platform (Windows, iOS, and Android), multi-form factor (desktop, tablet, and phone) solution for all in-store first-line workers, such as cashiers, sales associates, stock clerks, and store managers. It can be deployed as an app that has offline capabilities. Alternatively, it can be deployed in the cloud and accessed through a web browser. The application is role-based and fully configurable from headquarters. It's also highly customizable, and can be extended or integrated into third-party services by using the Retail software development kit (SDK).

In addition to standard "cash and carry" transaction processing, Modern POS includes features for assisted selling, clienteling, endless aisle, order processing/fulfillment, inventory management, cash/shift management, and reporting. For more information, see [Modern POS \(MPOS\) architecture](#) and [Choose between Modern POS \(MPOS\) and Cloud POS](#).

#### **E-commerce storefront**

The e-commerce storefront is the customer-facing website rendering system. It's built on the React.js framework, and uses a combination of server-side and client-side rendering to deliver responsive web

experiences for one or more online channels. Although the storefront has a rich set of out-of-box capabilities, it's also highly customizable, and delivers an efficient and scalable solution for online business. For more information, see [Online store overview](#).

#### **Site builder**

Site builder is the web-based authoring interface for the content management and storefront website rendering systems. Visual page builder in site builder is a what-you-see-is-what-you-get (WYSIWYG) editor for site managers and content authors who perform the day-to-day workflow tasks of managing and producing the marketing content for the e-commerce experience. In site builder, a marketer can provide more marketing detail for specific products to enhance the shopping experience for consumers. In addition, site builder includes integrated accessibility reporting, URL management, site map generation, and image focal point management, among other features. For more information, see [Online store overview](#).

#### **External services and apps**

The headless commerce engine that is exposed via the Commerce Scale Unit lets partners and customers take advantage of all the same channel-side capabilities and business logic that are used by the out-of-box e-commerce and point of sale (POS) components. Therefore, by tapping into the same data and business process capabilities, it allows for seamless omni-channel capabilities across out-of-box channel components and partner-provided/customer-developed services and applications. It also provides access to all out-of-box and ISV-developed surround services that are available through the Commerce Scale Unit.

#### **Back office**

##### **Dynamics 365 Commerce headquarters**

The Dynamics 365 Commerce application, which is often referred to as the Commerce headquarters component, provides back-office capabilities that enable the configuration of products, employees, business processes, and other functionality that is required for the business. It's also the application that call center workers use to provide assisted commerce-related workflows.

##### **Dynamics 365 Supply Chain Management**

[Dynamics 365 Supply Chain Management](#) provides functionality to help you manage your products throughout the supply chain lifecycle, from production, to inventory and warehouse, to transportation and distribution. For more information, see [Help resources for Supply Chain Management](#).

##### **Dynamics 365 Finance**

[Dynamics 365 Finance](#) provides functionality to automatically manage your global finances. For customers of the Dynamics 365 Commerce application, Dynamics 365 Finance offers an integrated experience for managing stores and e-commerce financial statements alongside the rest of their operations. For more information, see [Dynamics 365 Finance help resources](#).

##### **Dynamics 365 Human Resources**

[Dynamics 365 Human Resources](#) lets businesses get a comprehensive view of their employee resources and manage them in a unified way. It provides integrated experiences from the hiring process through workforce planning and employee time management. For more information, see [Dynamics 365 Human Resources help resources](#).

#### **Commerce Scale Unit**

Retailers are distributed organizations, where the business topography can be represented as a hub and spoke model. Dynamics 365 Commerce supports this model by having head-office capabilities (the hub), and also many distributed channel components (the spokes) that can be deployed and self-managed in-store or in nearby Microsoft-managed Azure datacenters. The spokes are referred to as Scale Units, because they represent physical isolation (a function of scale) and an atomic unit of update.

To facilitate cloud and edge computing scenarios, a Commerce Scale Unit is available both as a software as a service (SaaS) component that is managed by Microsoft (Commerce Scale Unit in the Cloud) and as a self-managed component that can be deployed locally (self-hosted). A single environment can have a mixture of Commerce Scale Units (cloud and self-hosted). Therefore, organizations can tune their investments in

operational overhead on a store-by-store basis by implementing network redundancy for poor connectivity. For more information, see [Select an in-store topology](#).

#### **Commerce Scale Units (cloud)**

Multiple Commerce Scale Units can be associated with each environment. Each Commerce Scale Unit can be independently serviced and updated, and each can serve one or more channels across one or more legal entities. Each Commerce Scale Unit can be deployed to any of the supported Azure regions, and multiple Commerce Scale Units can be deployed to the same region. The independent nature of each Commerce Scale Unit allows for phased rollout of updates across a collection of channels.

#### **Commerce Scale Units (self-hosted)**

The ability to bring the Commerce Scale Units to edge computing helps accommodate scenarios where internet connectivity is poor or unreliable. For retailers, this approach typically means having a physical footprint in their stores. By using a Commerce Scale Unit (self-hosted), retailers can bring the same business logic and capabilities that run in the Azure cloud into their stores. In these cases, although in-store connectivity is presumably more reliable, self-management of these components will involve additional overhead in terms of monitoring and updates. For more information, see [Select an in-store topology](#).

### **E-commerce platform**

#### **Content management system**

A fully featured content management system (CMS) is integrated directly into the e-commerce platform. In addition to rich indexing features, the CMS provides lifecycle management for marketing materials that supplement the product information that is managed by the headless commerce engine. It includes features for localization and multi-item publishing through releases. The system is built on top of a scalable, resilient Azure infrastructure that includes Azure Active Directory (Azure AD) and Azure Cosmos DB.

#### **Digital asset management**

Commerce digital asset management extends the content management store, and keeps track of images, videos, and file downloads that are served by the web storefront site. Its image resizer service optimizes downloaded images for different devices and contexts. In this way, it helps enhance performance while it also manages image quality. Digital asset management is also integrated with Azure Media Services for efficient playback of video streams.

#### **Web storefront**

The CMS stores its pages as a series of modules. The storefront web server assembles those modules into a rendered HTML page. The web storefront is composed of the rendering platform, the commerce data proxy, and the extensibility layer. Those components form a base that is supplemented by a set of modules that power a web-based commerce experience, the Dynamics 365 Commerce module library. The initial module library can be modified to meet each business's unique requirements. Alternatively, it can be supplemented by extensions and modules that are developed by a partner.

### **Commerce surround services**

#### **Dynamics 365 Fraud Protection**

[Dynamics 365 Fraud Protection](#) is integrated into the e-commerce checkout flows that are managed and processed through the Commerce Scale Unit. The connection to the service is automatically provisioned with the Commerce Scale Unit, and customers who sign up for Dynamics 365 Fraud Protection can enable and configure the integration in Commerce headquarters. The service can run either in "evaluate" mode, so that you can assess the effectiveness of the service, or in "protect" mode, so that you can catch fraudulent transactions by using configured business rules. For more information, see [Dynamics 365 Fraud Protection integration with Dynamics 365 Commerce](#).

#### **Dynamics 365 Customer Insights**

[Dynamics 365 Customer Insights](#) helps you gain a deeper understanding of your customers by connecting data from various transactional, behavioral, and observational sources to create a 360-degree customer view and generate insights. Dynamics 365 Commerce makes it easy for retailers to enable the integration with Dynamics 365 Customer Insights and show the generated insights at the POS. These insights include churn probability and

next best action, and they are valuable because they help sales associates have effective conversations with customers and deliver personalized shopping experiences to them. For more information, see [Dynamics 365 Customer Insights integration with Dynamics 365 Commerce](#).

### **Bing for Commerce**

[Microsoft Bing for Commerce](#) is integrated into Dynamics 365 Commerce to provide consistent product discovery and search experiences across all commerce channels that use the Commerce Scale Unit. In Commerce headquarters, retailers can configure boosting and sinking business rules for product discovery and search experiences. (For example, these rules can be used to boost product discovery for discounted products or remove items that are out of stock.) In this way, retailers can turn shopper frustration and site abandonment into active carts and converted sales. By taking advantage of an out-of-box capability that this integration provides, retailers can let customers use images to search for and discover similar products from a catalog without having to describe them.

### **Product recommendations**

Dynamics 365 Commerce can be used to show product recommendations on the e-commerce website and POS devices. These product recommendations are items that a customer might be interested in, and they are based on the purchase trends of other customers in online and brick-and-mortar stores.

Product recommendations let customers easily and quickly find products that they might want to purchase, and cross-selling and upselling can be used to help customers find additional products that they didn't originally intend to buy. When recommendations are used to assist with product discovery, they can help create more conversion opportunities, increase sales revenue, and enhance customer satisfaction and retention. For more information, see [Product recommendations overview](#).

### **Commerce analytics**

Dynamics 365 Commerce's prepackaged, business-managed commerce analytics solution provides retailers with intelligent insights across all points of the Commerce ecosystem by embedding Power BI reports in Commerce headquarters and POS systems. The commerce analytics solution provides a comprehensive set of out-of-box business and transactional reports, dashboards, and KPIs that take advantage of insights across all channels.

The solution standardizes data from various sources (such as transactional, behavioral, observational, or external data sources) into a unified data model that is hosted in Azure Data Lake Storage. Therefore, organizations can obtain a truly complete view of their business performance across channels. For example, they can analyze the performance of discount promotions, monitor web visits and activity, compare in-store visits and purchases with online purchases, track loyalty redemptions, or do customer recency, frequency, monetary (RFM) analysis.

### **Ratings and reviews**

The Commerce ratings and reviews solution lets online retail customers enter product reviews and ratings through the e-commerce storefront. Retailers can then show averaged ratings and review information across their e-commerce websites. Azure Cognitive Services offers automatic moderation of profane words in 40 languages, and because human approval isn't required, moderation costs are reduced. The system also offers moderator tools that can be used to respond to customer concerns, feedback, and take-down requests, and to address data requests from users. For more information, see [Rating and reviews overview](#).

### **Unified data components**

#### **Azure Data Lake Storage**

Customers who bring their own Azure Data Lake Storage accounts can take advantage of structured business data from back-office operations and clickstream data from the e-commerce storefront. This data flows back into intelligence services such as product recommendations, customer insights, and commerce analytics to power customer-centric business processes and user experiences. Those business processes and user experiences can then be embedded back into Dynamics 365 Commerce headquarters, the POS, and e-commerce storefronts. For more information, see [Make Entity store available as Data Lake](#).

## Dataverse

Dataverse is the unified data store that integrates the data from all your business applications. Dynamics 365 applications such as Dynamics 365 Sales, Dynamics 365 Customer Service, and Dynamics 365 Commerce use Dataverse to store business data. Therefore, Dataverse enables cross-business application scenarios, and can power new scenarios through Power Apps and Power Automate. For more information, see [What is Microsoft Dataverse?](#).

## Additional resources

[Dynamics 365 Commerce authentication flows](#)

[Azure Data Lake Storage](#)

[Dataverse](#)

[Modern POS \(MPOS\) architecture](#)

[Dynamics 365 Supply Chain Management](#)

[Dynamics 365 Human Resources](#)

[Dynamics 365 Finance](#)

[Dynamics 365 Fraud Protection](#)

[Dynamics 365 Fraud Protection integration with Dynamics 365 Commerce](#)

[Dynamics 365 Customer Insights](#)

[Microsoft Bing for Commerce](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Commerce authentication flows

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides an overview of the various authentication flows in Microsoft Dynamics 365 Commerce. Although the Dynamics 365 Commerce solution currently supports several authentication scenarios and flows, the core authentication infrastructure of the Commerce Scale Unit (also known as the headless commerce engine) is fully based on [OpenID Connect](#).

## Authentication methods

Access to each of the application programming interfaces (APIs) on the Commerce Scale Unit is natively restricted by one or more of the following roles:

- **Employee** – Access to APIs associated with this role requires point of sale (POS) device activation (a device token) and an authenticated employee.
- **Customer** – Access to APIs associated with this role requires an authenticated customer. E-Commerce sites generally use these APIs for operations such as retrieving order history and changing customer details.
- **Application** – Access to APIs associated with this role requires application-level authentication, such as Azure Active Directory (Azure AD) service-to-service authentication.
- **Anonymous** – APIs associated with this role are primarily used by e-Commerce sites without user authentication.
- **Customized APIs** – Access to APIs associated with this role can be restricted using any of the methods described above such as POS device activation, customer authentication, and anonymous authentication.

For the full list of Commerce Scale Unit APIs and their access restrictions, see [Commerce Scale Unit customer and consumer APIs](#).

## Supported authentication methods

The following table describes the set of supported authentication methods for APIs that require either POS device activation that generates a device token or user authentication that generates a user token.

API CATEGORY	SCENARIO	SUPPORTED AUTHENTICATION METHOD	REQUIRED SETUP	ADDITIONAL DETAILS
Employee	Dynamics 365 POS authentication flows*	Simple cashier user name and password	In Dynamics 365 Commerce headquarters, configure a user name and password for a worker.	<a href="#">Create a worker</a>
Employee	Dynamics 365 POS authentication flows*	Azure AD credentials	In Commerce headquarters, configure a worker that is mapped to Azure AD credentials.	<a href="#">Enable Azure Active Directory authentication for POS sign-in</a>



API CATEGORY	SCENARIO	SUPPORTED AUTHENTICATION METHOD	REQUIRED SETUP	ADDITIONAL DETAILS
Employee	Dynamics 365 POS authentication flows*	Extended sign-in credentials (for example, by using a bar code or a magnetic stripe reader [MSR])	In Commerce headquarters, configure a worker for extended sign-in.	<a href="#">Set up extended logon functionality for MPOS and Cloud POS</a>
Customer	Dynamics 365 Commerce authentication flows	Site user authentication by using Azure AD B2C	<ol style="list-style-type: none"> <li>1. Create an Azure AD business-to-consumer (B2C) application.</li> <li>2. In Commerce headquarters, add the Azure AD B2C application to the accepted list of identity providers.</li> <li>3. In Commerce site builder, configure the Azure AD B2C application.</li> </ol>	<a href="#">Set up a B2C tenant in Commerce</a> <a href="#">Set up custom pages for user sign-ins</a>
Customer	Dynamics 365 Commerce authentication flows	Site user authentication by using an external identity provider that supports OpenID Connect	<ol style="list-style-type: none"> <li>1. Create an Azure AD B2C application, and configure it to support external identity providers.</li> <li>2. In Commerce headquarters, add the Azure AD B2C application to the accepted list of identity providers.</li> <li>3. In Commerce site builder, configure the Azure AD B2C application.</li> </ol>	<a href="#">Set up a B2C tenant in Commerce</a> <a href="#">Set up custom pages for user sign-ins</a>
Customer	Third-party e-Commerce authentication flows	Site user authentication by using an external identity provider that supports OpenID Connect	In Commerce headquarters, add the external identity provider to the accepted list of identity providers.	<a href="#">Configure authentication providers</a>

API CATEGORY	SCENARIO	SUPPORTED AUTHENTICATION METHOD	REQUIRED SETUP	ADDITIONAL DETAILS
Application	Third-party app or service authentication flows	Azure AD service-to-service authentication/application authentication	In Commerce headquarters, add the external identity provider to the accepted list of identity providers.	

\* Sign-in to POS requires device activation for each terminal. For more information, see [Point of Sale \(POS\) device activation](#).

### Unsupported authentication flows

SCENARIO	UNSUPPORTED AUTHENTICATION METHOD	DETAILS
Dynamics 365 POS authentication flows	Authentication without device activation (that is, without a device token)	All POS-related Commerce Scale Unit APIs require a device activation token for authentication.

## Dynamics 365 POS employee authentication flows

The following illustration shows POS employee authentication flows in Commerce.



# Dynamics 365 Commerce

## Dynamics 365 POS Employee Authentication Flows

### POS DEVICE ACTIVATION

#### 1. GRANT DEVICE ACTIVATION PERMISSION

1.1. IT admin maps Azure AD user to worker in Finance and Operations apps to grant device activation permissions.

#### 2. ACTIVATE DEVICE

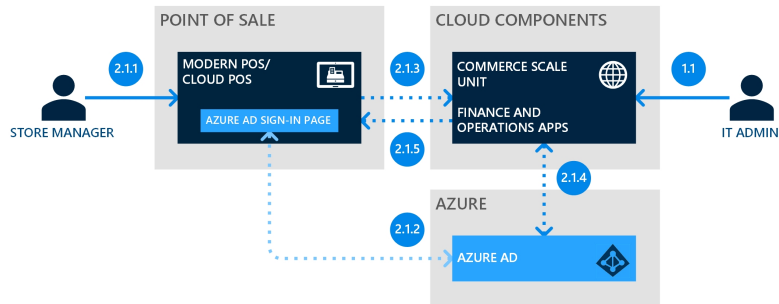
2.1.1. Cashier opens the POS application and signs in with Azure AD credentials.

2.1.2. Azure AD user token is generated via Azure AD.

2.1.3. Azure AD user token is passed to Commerce Scale Unit to activate the device.

2.1.4. Commerce Scale Unit validates Azure AD user token with Azure AD.

2.1.5. Commerce Scale Unit issues device token to the POS application for further authentication.



### POS CASHIER AUTHENTICATION USING SIMPLE USER NAME AND PASSWORD

#### 1. GRANT POS WORKER PERMISSION

1.1. IT admin grants worker permissions for POS sign-in, including worker ID and password.

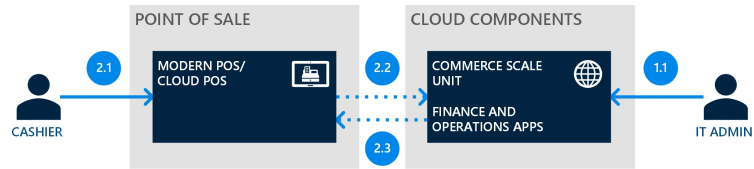
#### 2. SIGN IN TO POS

2.1. Cashier opens the POS application and signs in with worker credentials maintained in Finance and Operations apps.

2.2. Commerce Scale Unit validates user input against user credentials entered in Finance and Operations apps.

2.3. Commerce Scale Unit generates and issues user token to the POS application for further authentication.

**NOTE:** All subsequent requests are authenticated using the device token and user token.



### POS CASHIER AUTHENTICATION USING EXTENDED SIGN-IN

#### 1. GRANT POS WORKER PERMISSION

1.1. IT admin grants worker permissions for POS sign-in, setting up extended logon capabilities.

#### 2. ENROLL IN EXTENDED SIGN-IN

2.1. Store manager signs in to the POS and enrolls cashier with extended sign-in credentials (for example, bar code or MSR).

2.2. Commerce Scale Unit stores hashed and encrypted authentication credentials.

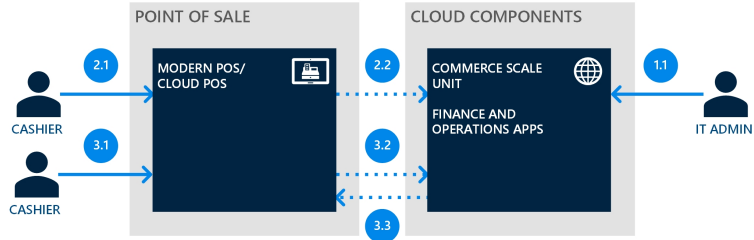
#### 3. SIGN IN TO POS WITH EXTENDED CREDENTIALS

3.1. Cashier signs in to the POS using extended credentials (for example, bar code or MSR).

3.2. Commerce Scale Unit validates user input against user credentials entered in Finance and Operations apps.

3.3. Commerce Scale Unit generates and issues user token to the POS application for further authentication.

**Note:** All subsequent requests are authenticated using the device token and user token.



### POS CASHIER AUTHENTICATION USING AZURE AD

#### 1. GRANT DEVICE ACTIVATION PERMISSION

1.1. IT admin maps Azure AD user to worker in Finance and Operations apps.

#### 2. SIGN IN TO POS

2.1. Cashier opens the POS application and signs in with Azure AD credentials.

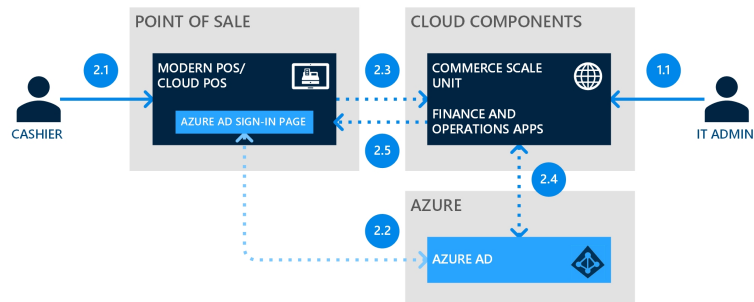
2.2. Azure AD user token is generated via Azure AD.

2.3. Call is made to Commerce Scale Unit to authenticate the user.

2.4. Commerce Scale Unit validates Azure AD user token.

2.5. Commerce Scale Unit generates and issues user token to the POS application for further authentication.

**Note:** All subsequent requests are authenticated using the device token and user token.



The following illustration shows e-Commerce customer authentication flows in Commerce.



## Dynamics 365 Commerce Dynamics 365 E-Commerce Customer Authentication Flows

### E-COMMERCE USER AUTHENTICATION USING AZURE AD B2C AS IDENTITY PROVIDER

#### 1. CONFIGURE AZURE AD B2C APPLICATION

1.1. IT admin creates a new Azure AD B2C application in Azure.

**Note:** Migration of existing customers to the Azure AD B2C instance is supported.

1.2. IT admin adds the Azure AD B2C application to the accepted identity provider list in Finance and Operations apps.

1.3. IT admin configures the Azure AD B2C application in Commerce site builder.

#### 2. SIGN IN TO E-COMMERCE SITE

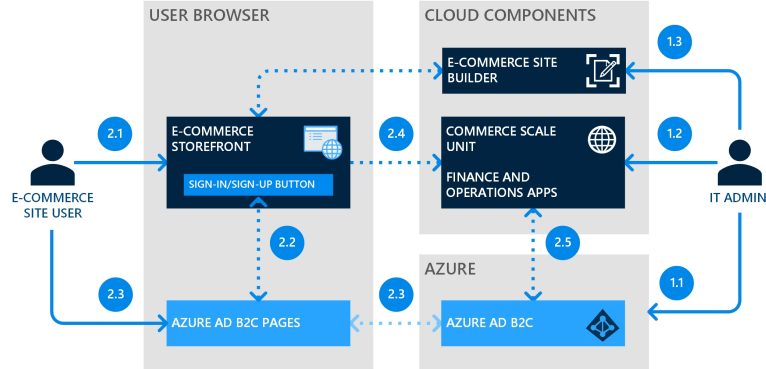
2.1. E-Commerce site user selects the sign-in/sign-up button on the e-Commerce site.

2.2. User is redirected to the Azure AD B2C sign-in/sign-up page.

2.3. E-Commerce site user signs in or signs up on the Azure AD B2C sign-in/sign-up page and receives user token.

2.4. Call is made to Commerce Cloud Scale Unit to validate the user.

2.5. Commerce Scale Unit validates Azure AD B2C user token and creates a new customer record if it does not exist.



### E-COMMERCE USER AUTHENTICATION USING EXTERNAL IDENTITY PROVIDER

#### 1. CONFIGURE AZURE AD B2C APPLICATION

1.1. IT admin configures single sign-on flows on the external identity provider.

1.2. IT admin creates a new Azure AD B2C application in Azure, including policies (for example, password reset), and enables additional external identity providers.

**Note:** Migration of existing customers to the Azure AD B2C instance is supported.

1.3. IT admin adds the Azure AD B2C application to the accepted identity provider list in Finance and Operations apps.

1.4. IT admin configures the Azure AD B2C application for the e-Commerce sign-in and sign-up flows through Commerce site builder.

#### 2. SIGN IN TO E-COMMERCE SITE

2.1. E-Commerce site user selects the sign-in/sign-up button on the e-Commerce site.

2.2. User is redirected to the Azure AD B2C sign-in/sign-up page.

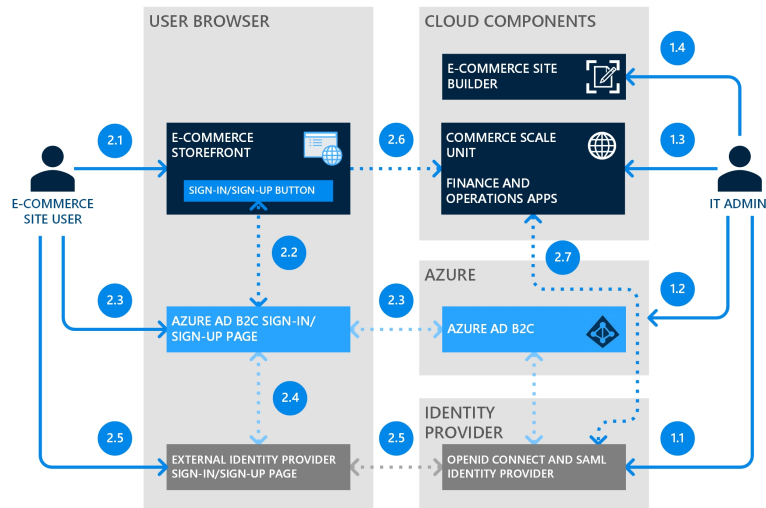
2.3. E-Commerce site user selects the button to be redirected to external identity provider for sign-in/sign-up.

2.4. User is redirected to external identity provider site for sign-in/sign-up.

2.5. E-Commerce site user signs in or signs up on the external identity provider sign-in/sign-up page and receives user token.

2.6. Call is made to Commerce Cloud Scale Unit to validate the user.

2.7. Commerce Scale Unit validates Azure AD B2C user token and creates a new customer record if it does not exist.



## Third-party e-Commerce customer authentication flows

The following illustration shows third-party e-Commerce customer authentication flows in Commerce.



## Dynamics 365 Commerce Third-Party E-Commerce Authentication Flows

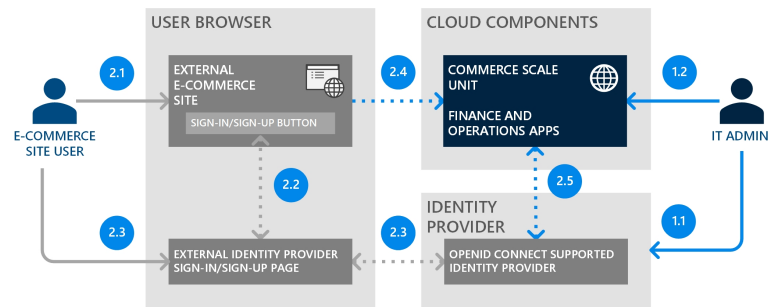
### E-COMMERCE USER AUTHENTICATION USING EXTERNAL IDENTITY PROVIDER

#### 1. CONFIGURE AZURE AD B2C APPLICATION

- 1.1. IT admin configures external identity provider.
- 1.2. IT admin adds the external identity provider to the accepted identity provider list in Finance and Operations apps.

#### 2. SIGN IN TO E-COMMERCE SITE

- 2.1. E-Commerce site user selects the sign-in/sign-up button on the external e-Commerce site.
- 2.2. User is redirected to the sign-in/sign-up page for the external identity provider.
- 2.3. E-Commerce site user signs in or signs up on the external identity provider sign-in/sign-up page and receives user token.
- 2.4. Call is made to Commerce Cloud Scale Unit to validate the user.
- 2.5. Commerce Scale Unit validates user token issued by external identity provider and creates a new customer record if it does not exist.



## Third-party application authentication flows

The following illustration shows third-party application authentication flows in Commerce.



## Dynamics 365 Commerce Third-Party Application Authentication Flows

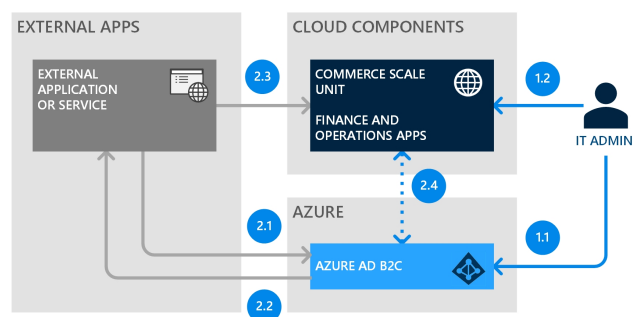
### EXTERNAL APPLICATION OR SERVICE AUTHENTICATION USING AZURE AD APPLICATION AND SECRET

#### 1. CONFIGURE AZURE AD B2C APPLICATION

- 1.1. IT admin creates a new Azure AD application in your Azure AD tenant, using a secret as the form of authentication.
- 1.2. IT admin adds the Azure AD application to the accepted identity provider list in Finance and Operations apps.

#### 2. SIGN IN VIA THIRD-PARTY APPLICATION

- 2.1. Authenticate with Azure AD using the application secret.
- 2.2. Azure AD issues an application token that can be used to authenticate with the Commerce Scale Unit.
- 2.3. Call is made to Commerce Scale Unit using the application token issued by Azure AD.
- 2.4. Commerce Scale Unit validates the Azure AD application token.



## Additional resources

[Dynamics 365 Commerce architecture overview](#)

[Commerce Scale Unit customer and consumer APIs](#)

[POS worker logon](#)

[Enable Azure Active Directory authentication for POS sign-in](#)

[Set up extended logon functionality for MPOS and Cloud POS](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user sign-ins](#)

[Configure authentication providers](#)

## Point of Sale (POS) device activation

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

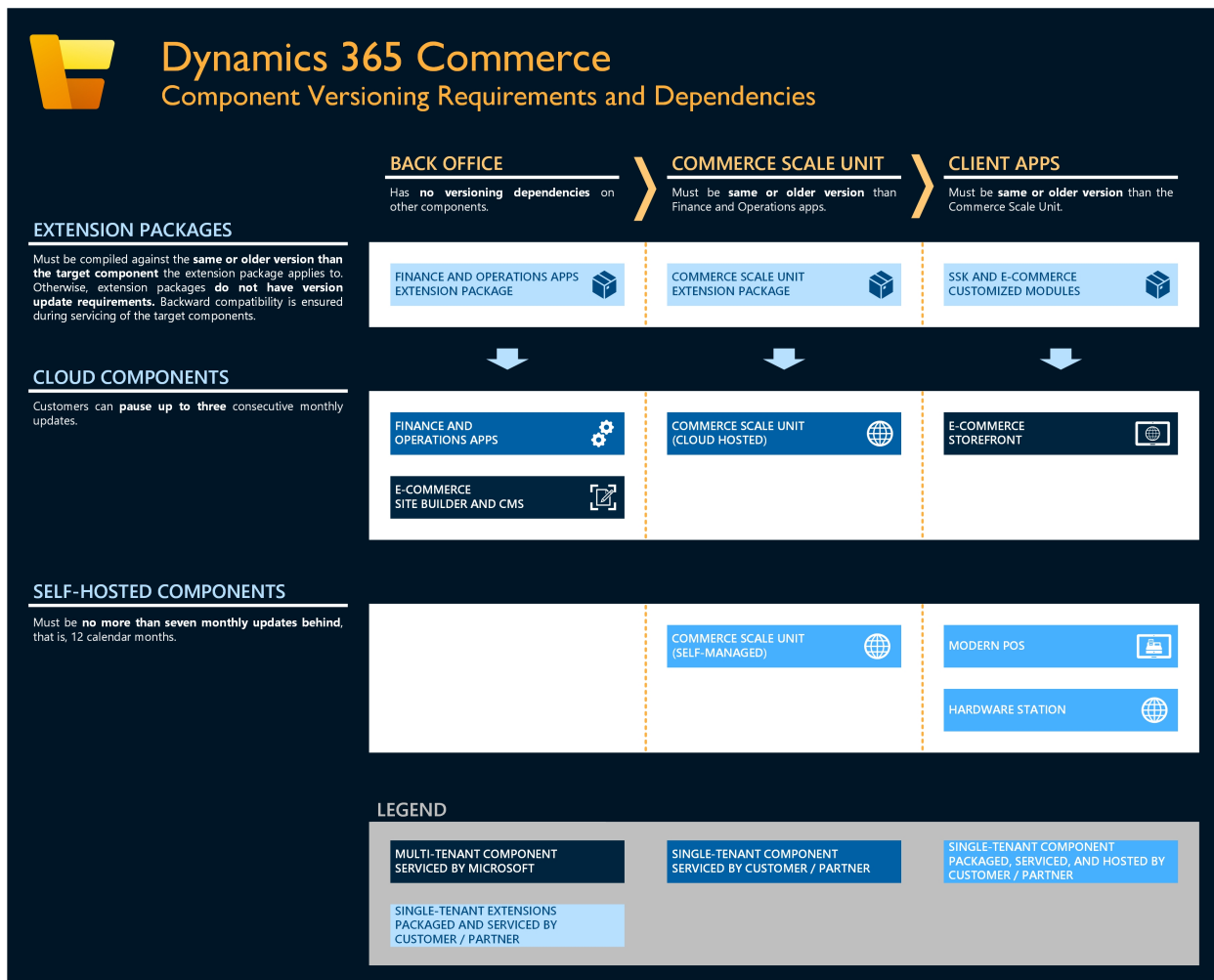
# Dynamics 365 Commerce component versioning requirements

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides an overview of the component versioning requirements and dependencies for all components in the Microsoft Dynamics 365 Commerce ecosystem.

## Overview

The following illustration shows an overview of Dynamics 365 Commerce components and corresponding versioning requirements and dependencies.



## Component dependencies

### Service updates

To ensure compatibility between all Commerce components that are serviced and deployed by customers and partners, you must follow several versioning dependencies during servicing updates. The following list describes all these dependencies.

- Commerce headquarters and Finance and Operations apps must be on the same version as, or a newer version than, Commerce Scale Unit (both cloud and self-hosted).

For example, if Commerce headquarters and Finance and Operations apps are on version 10.0.16,

Commerce Scale Unit must be on version 10.0.16 or earlier (for example, 10.0.15 or 10.0.14).

- **Commerce Scale Unit must be on the same version as, or a newer version than, Modern Point of Sale (POS), Hardware Station, and the Commerce software development kit (SDK) and associated local site configurations (such as modules, data actions, and themes).**

For example, if Commerce Scale Unit is on version 10.0.16, Modern POS, Hardware Station, and the Commerce storefront must be on version 10.0.16 or earlier (for example, 10.0.15 or 10.0.14).

- **Extension packages must be compiled against the same version as, or a newer version than, the target component that the extension applies to.**

For example, if the deployed Commerce Scale Unit is on version 10.0.16, the corresponding extension packages must be compiled against version 10.0.16 or earlier (for example, 10.0.15 or 10.0.14).

### Quality updates

During quality updates, no specific versioning requirements must be followed for each Commerce component, besides what is required for service updates.

## Current supported versions

The following table describes the current supported versions of various Commerce components as of **January 25, 2021**.

COMPONENT	LATEST AVAILABLE RELEASE (FIRST RELEASE AVAILABLE IN SANDBOX)	LATEST AVAILABLE COMPONENT VERSION NUMBER (FIRST RELEASE AVAILABLE IN SANDBOX)	EARLIEST SUPPORTED RELEASE	EARLIEST SUPPORTED COMPONENT VERSION NUMBER
Finance and Operations apps	10.0.16	10.0.16	10.0.12	10.0.12
Commerce Scale Unit (cloud-hosted)	10.0.16	9.26	10.0.12	9.23
Commerce module library	10.0.16	9.26	10.0.11	9.23
Commerce Scale Unit (self-hosted)	10.0.16	9.26	10.0.8	9.18
Modern POS	10.0.16	9.26	10.0.8	9.18
Hardware Station	10.0.16	9.26	10.0.8	9.18

## One Version requirements

Commerce components follow the same [One Version service updates](#) that were announced in July 2018, and also the subsequently published [flexible service updates for Finance and Operations apps](#) that were announced in June 2019. For more information, see [One Version service updates FAQ](#).

### Cloud components

Customers can pause up to three consecutive updates across the following components. (Three updates correspond to approximately six calendar months.)

- Commerce headquarters and Finance and Operations apps



- Commerce Scale Unit (cloud-hosted)
- Commerce SDK and associated local site configurations (such as modules, data actions, and themes)

For example, customers who are currently on version 10.0.2 can pause updates to versions 10.0.3, 10.0.4, and 10.0.5. However, they must then update to version 10.0.6. In this scenario, after version 10.0.7 becomes available, version 10.0.2 is no longer supported.

### **In-store components**

Customers can pause up to seven consecutive updates across the following components. (Seven updates correspond to approximately twelve calendar months.)

- Commerce Scale Unit (in-store hosted)
- Modern POS
- Hardware Station

For example, customers who are currently on version 10.0.2 can pause updates to versions 10.0.3 through 10.0.9. However, they must then update to version 10.0.10. In this scenario, after version 10.0.11 becomes available, version 10.0.2 is no longer supported.

## **Additional resources**

### **One Version service updates**

For more information about One Version service updates, see the following resources:

- [One Version service updates FAQ](#)
- [Modernizing the way we update Dynamics 365](#)
- [New flexible service updates for Finance and Operations apps](#)

### **Component selection**

For more information about how to select the correct components to meet your needs, see the following topics:

- [Select an in-store topology](#)
- [Choose between Modern POS \(MPOS\) and Cloud POS \(CPOS\)](#)

### **Servicing instructions**

For more information about how to service individual components that are described in this topic, see the following topics:

- [Configure and install Commerce Scale Unit](#)
- [Apply updates and extensions to Retail Cloud Scale Unit](#)
- [Configure, install, and active Modern POS \(MPOS\)](#)
- [Configure and install Retail hardware station](#)
- [Package configurations and deploy them to an online channel](#)

### **Extensibility and packing**

For more information about serviceability for extensions, see [Create deployable packages](#).

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

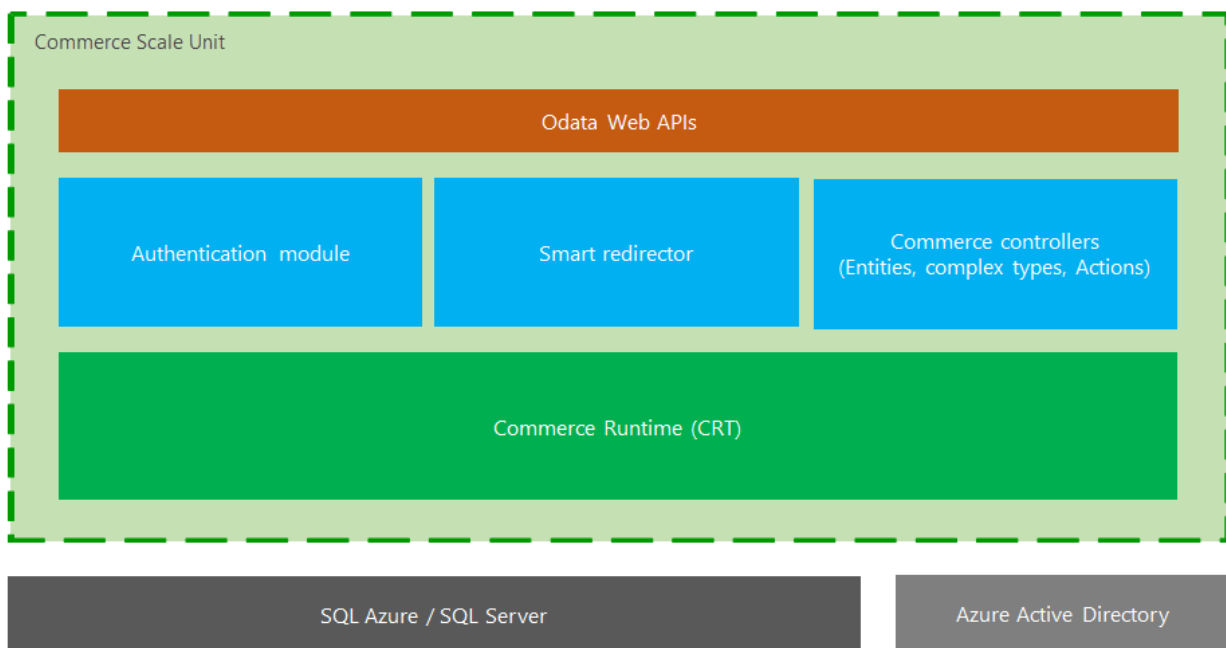
# Commerce Scale Unit architecture

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes the architecture of Commerce Scale Unit. Commerce Scale Unit provides stateless services and business logic for Modern Point of Sale (MPOS) and E-Commerce clients.

## Commerce Scale Unit architecture

The commerce runtime is wrapped in a Commerce Scale Unit layer. Commerce Scale Unit uses a web API and OData to support thin clients both in the store and online on tablets and phones. The commerce runtime communicates with Headquarters through Commerce Data Exchange services. The following diagram shows the architecture of Commerce Scale Unit.



Commerce Scale Unit uses the following concepts.

CONCEPT	DESCRIPTION
Entity type	An entity type is an entity that has a life cycle that you want to monitor. Each entity type has a key. An example of an entity type is <b>Customer</b> .
Complex type	A complex type is an OData concept that is designed to prevent duplication by grouping specific related properties. These related properties can be reused in multiple entities. For example, <b>Customer</b> is an entity type that has a customer address. This customer address is a wrapper that contains an address line, city, state, and ZIP/postal code. Therefore, <b>Customer address</b> is a complex type that can be reused by other entity types. For example, the <b>Order</b> entity type requires the same address information that is associated with the <b>Customer</b> entity type and therefore reuses the <b>Customer address</b> complex type.

CONCEPT	DESCRIPTION
Controller	<p>A controller is a mapping for an entity type that controls create, read, update, and delete (CRUD) behaviors and actions for the entity type. A controller is provided for each commerce entity. You can customize the following controllers:</p> <ul style="list-style-type: none"> <li>• Carts</li> <li>• Catalogs</li> <li>• Categories</li> <li>• Commerce</li> <li>• Commerce Lists</li> <li>• Composite Key Entity</li> <li>• Controller Assembly Resolver</li> <li>• Customers</li> <li>• Employees</li> <li>• Non-Bindable Action</li> <li>• Org Units</li> <li>• Picking Lists</li> <li>• Products</li> <li>• Purchase Orders</li> <li>• Sales Orders</li> <li>• Shifts</li> <li>• Stock Counts Journals</li> <li>• Transfer Orders</li> </ul>
Metadata	<p>Metadata defines the contract between the client and the server.</p>

You can create your own entity type or complex type, extend an existing controller, add a new controller, and customize the metadata. If you customize the commerce runtime, you must also customize various components in Commerce Scale Unit to expose those changes to your Retail Modern POS clients.

**NOTE**

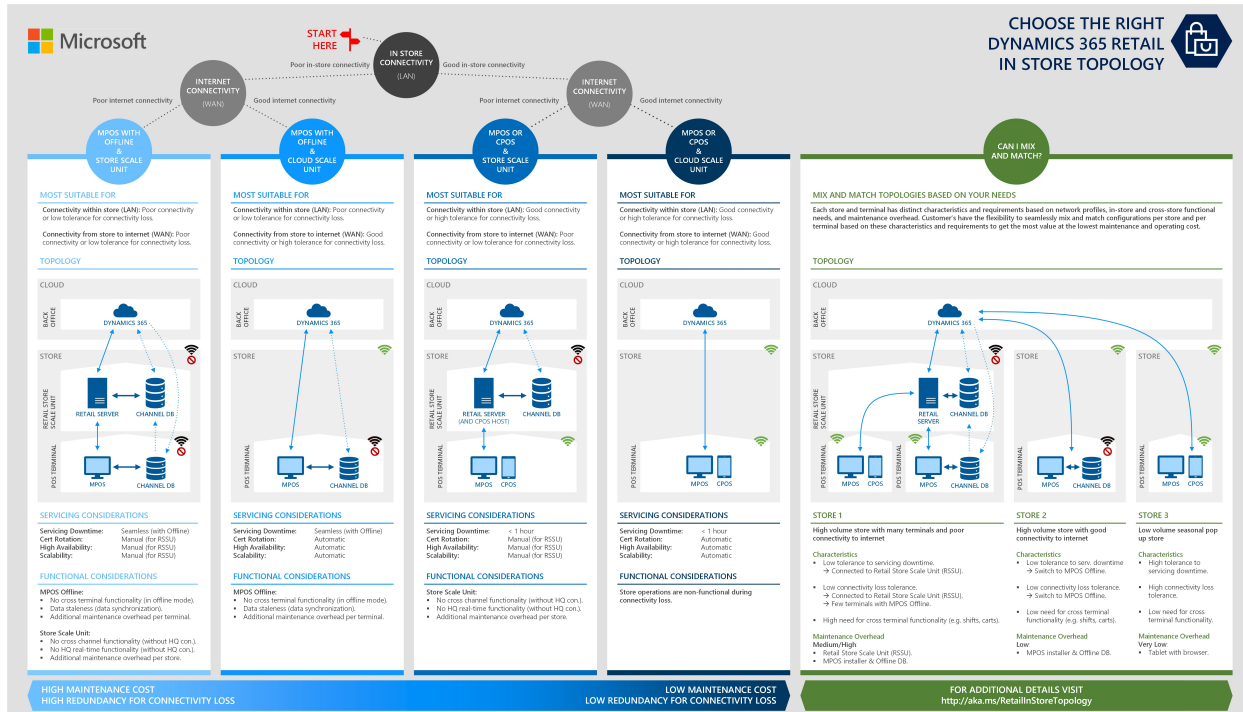
Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Select an in-store topology

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of the various Dynamics 365 Commerce in-store topologies.



## Supported capabilities when connectivity is lost

OPERATION	WITHOUT CONNECTIVITY TO COMMERCE SCALE UNIT (IN MPOS OFFLINE MODE)	WITHOUT CONNECTIVITY TO HQ (COMMERCE SCALE UNIT (SELF-HOSTED))
Cross terminal shifts (such as view, suspend, resume, close)		✓
Cross terminal transactions (such as view, suspend, resume)		✓

## Supported operations when connectivity is lost

For a list of operations that are supported when the POS loses connectivity to the HQ, see [Online and offline point of sale \(POS\) operations](#).

## Supported deployment and maintenance capabilities

Mass deployment is supported in Modern POS, but not in Commerce Scale Unit. For more information, see [Mass deployment of self-service components](#).

## Deployed components

The following components are deployed through a single installer. This means that they do not need to be installed individually.

## Modern POS

DEPLOYED COMPONENT	COMPONENT TYPE	NOTES
Modern POS App	Universal Windows Platform (UWP) Application	The Modern POS application running on the register.
Modern POS Client Broker	COM Surrogate hosting native binaries for the Modern POS	Hosts the Commerce Runtime to support operations to execute in offline mode as well as Async Client Libraries needed to synchronize data between the Modern POS and the HQ.
Channel Database	SQL Database	Register specific Channel Database instance hosting data for the register.

## Commerce Scale Unit (self-hosted)

INSTALLED COMPONENT	COMPONENT TYPE	NOTES
Commerce Scale Unit (self-hosted)	IIS Web Service	Scale unit specific Commerce Scale Unit (self-hosted) instance used by one or more stores.
Channel Database	SQL Database	Scale unit specific Store specific Channel Database instance hosting data for one or more stores.
Async Client Service	Windows Service	Component to synchronize master record data from the HQ to the store and transactional data from the store to the HQ.
Cloud POS	IIS Web Service	Cloud POS application that hosts POS functionality through a web browser.

## Additional resources

### MPOS offline mode

For more information about MPOS offline mode, see:

- [Offline point of sale \(POS\) functionality](#)
- [Online and offline point of sale \(POS\) operations](#)
- [Mass deployment of self-service components](#)

### Commerce Scale Unit (self-hosted)

The Commerce Scale Unit (self-hosted) is a set of components that can be deployed in a customer environment, such as inside a store, that can support continuous operations if connectivity to the back office or headquarters (HQ) is lost.

For more information, see:

- [Configure and install Commerce Scale Unit \(self-hosted\)](#)
- [Commerce Scale Unit \(self-hosted\)](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce Data Exchange best practices

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic is intended for people who implement functionality that is related to data synchronization, Commerce Data Exchange (CDX), in a Microsoft Dynamics 365 Commerce environment. It gives best practices advice for implementations.

## Overview

Proper data configuration and data synchronization are crucial to correctly functioning implementation. Regardless of business requirements, IT infrastructure, and overall preparedness, if data isn't correctly synchronized, the whole environment is effectively useless. Therefore, a top priority is to understand what is required to configure, generate, synchronize, and verify data across the full implementation from Commerce headquarters through the Commerce Scale Unit to the brick-and-mortar stores that use Modern POS with an offline database.

Before you go through this topic, it's important that you understand the concepts of a channel (store), registers and devices, and the Modern POS offline database. Therefore, we recommend that you review some of the resources at the end of this topic, such as the Device management implementation guide and the overview of the Commerce architecture.

The main content of this topic is organized into tables, where the first column includes lists of tag-like "associated areas" to help you more quickly find best practices that are related to your areas of concern. For new implementations, you might find it useful to copy these tables to a location where you can check off the various best practices as they are completed. In this way, you can help ensure that the implementation is prepared as well as possible before you move forward to production.

## Updating configurations

The following should be performed after every update to the Dynamics 365 environment.

ASSOCIATED AREAS	BEST PRACTICE
<ul style="list-style-type: none"><li>Parameters</li><li>Commerce scheduler</li><li>Initialization</li></ul>	Go to <b>Retail and Commerce &gt; Headquarters setup &gt; Commerce scheduler &gt; Initialize commerce scheduler</b> . You will be asked if you would like to proceed with initializing the base configuration data for Commerce scheduler. Performing this action after every update is key to maintaining functionality as it correctly sets the configuration data for new tables or columns. There is also a parameter to <b>Delete existing configuration</b> . Unless explicitly told to do so or working on a non-production environment where losing configuration will not create an impact, leave this set to <b>No</b> .

## Valuable configurations

ASSOCIATED AREAS	BEST PRACTICE
<ul style="list-style-type: none"> <li>• Parameters</li> <li>• Commerce scheduler</li> <li>• Retry</li> </ul>	<p>Go to <b>Retail and Commerce &gt; Headquarters setup &gt; Parameters &gt; Commerce scheduler parameters</b>, and set <b>Try count</b> to <b>3</b>. If the value of this field is too high, download sessions might fail during high-usage times. Additionally, verify (or set) <b>Full dataset generation interval in days</b> to <b>0</b>. This means full dataset generation will not occur unless required by something other than time. Setting these values allows CDX to function in a more expected manner while reducing possible error or performance issues.</p>
<ul style="list-style-type: none"> <li>• Functionality profile</li> <li>• Data retention</li> <li>• Return policy</li> </ul>	<p>Go to <b>Retail and Commerce &gt; Channel setup &gt; POS setup &gt; POS profiles &gt; Functionality profile</b>, and then, in the <b>Functions</b> section, set <b>Days transactions exist</b> to a value that is the same as, or close to the value that is defined for the return policy. For example, if the return policy states an item can be returned within 30 days, set this field to <b>30, 31, or 60</b> if special exceptions are allowed beyond the usual policy (this would be twice the usual policy, allowing for faster returns even beyond the usual policy limits).</p>
<ul style="list-style-type: none"> <li>• Channel database group</li> <li>• Distribution schedule</li> <li>• Offline profile</li> <li>• Pause</li> <li>• Data</li> <li>• Download</li> </ul>	<p>We highly recommend that you have either a "dummy" channel database group (that is, a group that isn't associated with any distribution schedule job) that you assign to the newly generated terminals, or a special offline profile where the <b>Pause offline synchronization</b> option is set to <b>Yes</b>. In this way, data generation can occur when it's required and when the system is most available to do it. (However, the system might pause multiple times as required.)</p>

## Practices that affect performance

ASSOCIATED AREAS	BEST PRACTICE
------------------	---------------



ASSOCIATED AREAS	BEST PRACTICE
<ul style="list-style-type: none"> <li>• Channel database group</li> <li>• Offline profile</li> <li>• Pause</li> <li>• Data</li> <li>• Download</li> </ul>	<p>Don't generate data for offline databases until that data is required so that an offline database can be used. The following scenario shows why this best practice is important.</p> <p>A new Modern POS offline database that has been added to the relevant channel database group inherits all existing download sessions since the last full database synchronization occurred. One hundred new Modern POS instances that have offline terminals are created, and a full synchronization hasn't occurred in two months. Only five scheduler jobs have actual changes every 20 minutes. (For example, these changes might involve prices and discounts, or customers, which can be updated often.) In this scenario, up to 2,000,000 download sessions are immediately generated and must be applied, regardless of whether the newly created terminals are activated and capable of applying this data.</p> <p>Even at the best times, this type of exceptional data generation is large and affects performance. At the worst (that is, busiest) times, it severely impairs the environment's performance. Therefore, we highly recommend that you either have a "dummy" channel database group (that is, a group that isn't associated with any distribution schedule job) that you assign to the newly generated terminals, or set the <b>Pause offline synchronization</b> option for the offline profile to <b>Yes</b>. By setting the <b>Pause offline synchronization</b> option to <b>Yes</b>, you stop data generation for anything that uses the offline profile. Therefore, data generation can occur only when it's required, instead of constantly, and only when the system is most available to do it. (However, the system might pause multiple times as required.)</p>
<ul style="list-style-type: none"> <li>• Distribution schedule</li> <li>• Scheduler jobs</li> <li>• Upload</li> </ul>	<p>No more than one P-job (upload batch job) should occur at any time. If multiple P-job upload batch jobs are created that might occur in parallel, table locking and delays (performance degradation) could occur while data of the uploaded transactions is being applied. The job doesn't have to occur multiple times at the same time. It can just occur frequently.</p>

ASSOCIATED AREAS	BEST PRACTICE
<ul style="list-style-type: none"> <li>Parameters</li> <li>Commerce scheduler</li> <li>Post database sync</li> </ul>	<p>Go to <b>Retail and Commerce &gt; Headquarters setup &gt; Parameters &gt; Commerce scheduler parameters</b>. Under the <b>Post database sync</b> sub-heading, there are two fields that can impact performance for different reasons.</p> <p><b>Clean up irrelevant master data after sync</b> performs the store procedure <b>Strip Master Data</b> after each CDX download session is executed. This step deletes unnecessary records that were included in package generation, but not necessary for the offline functionality for that specific device as a member of a specific store (channel). This feature assists in minimizing the data stored in the offline database. A smaller database can assist performance and minimize size issues if using an Express version of SQL. It is recommended to test this feature in Sandbox first as some custom SQL views could introduce dependencies between tables that this functionality is not aware of, resulting in errors in functionality.</p> <p><b>Optimize database statistics automatically</b> runs an update on the table statistics for each table in the CDX download session applied to the channel database. Outdated table statistics cause more performance issues than fragmented indexes. When enabled, it is also recommended to add a specific flag into the configuration parameters. If using this, go to <b>Retail and Commerce &gt; Headquarters setup &gt; Parameters &gt; Commerce shared parameters</b>, navigate to the <b>Configuration parameters</b> listing, and enter the new flag <b>CDX_ENABLE_UPDATE_STATISTICS_FOR_REQUIRED_T</b> <b>ABLE</b> with the value <b>1</b>. Note that in a future release this configuration parameter will be automatically added to all environments, but it is valuable to verify that it exists.</p>

## Additional resources

- [Commerce Data Exchange troubleshooting](#)
- [Commerce Data Exchange implementation guidance](#)
- [Dynamics 365 Commerce architecture overview](#)
- [Select an in-store topology](#)
- [Device management implementation guidance](#)
- [Configure, install, and activate Modern POS \(MPOS\)](#)
- [Configure and install Commerce Scale Unit \(self-hosted\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce Data Exchange and commerce channel communications

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides an overview of Commerce Data Exchange and its components. It explains the part that each component plays in the transfer of data between Microsoft Dynamics 365 Commerce Headquarters and channels.

## Overview

Commerce Data Exchange is a system that transfers data between Headquarters and channels, such as online stores or brick-and-mortar stores. The database that stores data for a channel is separate from the Commerce database. The channel database holds only the data that is required for transactions. Master data is configured in Headquarters and distributed to channels. Transactional data is created in the point of sale (POS) system or the online store, and then uploaded to Headquarters. Data distribution is asynchronous. In other words, the process of gathering and packaging data at the source occurs separately from the process of receiving and applying data at the destination. For some scenarios, such as price and inventory lookups, data must be retrieved in real time. To support these scenarios, Commerce Data Exchange also includes a service that enables real-time communication between Headquarters and a channel.

## Async Service

Microsoft SQL Server change tracking on the Commerce database is used to determine the data changes that must be sent to channels. Based on a distribution schedule, Headquarters packages that data and saves it to central storage (Azure blob storage). A separate batch process uses the Commerce Data Exchange: Async Client library to insert this data package into the channel database.



## Commerce scheduler

Scheduler jobs are the mechanism for distributing data to and from locations. Jobs are made up of subjobs, which specify the tables and table fields that contain the data to distribute. Headquarters includes predefined scheduler jobs and subjobs that meet the replication requirements of most organizations. The following types of predefined jobs are created:

- **Download jobs** – Download jobs send data that has changed from Headquarters to channel databases. Modifications to records are tracked through SQL Server change tracking.
- **Upload jobs (P jobs)** – Upload jobs pull sales transactions from a channel into the Commerce database. P jobs upload data incrementally. When a P job runs, the Async Client library checks the replication counter for records that have already been received from a location. A record is uploaded only if its replication counter is

more than the largest value that is found. P jobs don't update data that was previously uploaded.

The distribution schedule is used to run the data transfer, either manually or by scheduling a batch job in Headquarters. A distribution schedule can contain one or more channel data groups, and one or more scheduler jobs. To ensure that the scheduler jobs are running smoothly, do not rename the "Default" database configured for the instance, and do not create a second database. All non-Commerce Scale Unit databases are managed by Microsoft, and only one default database is expected.

## Realtime Service

Commerce Data Exchange: Real-time Service is an integrated service that provides real-time communication between Headquarters and channels. Real-time Service enables individual POS computers and online stores to retrieve specific data from Headquarters in real time. Although most key operations can be performed in the local channel database, the following scenarios require direct access to the data that is stored in Headquarters:

- Issuing and redeeming gift cards
- Redeeming loyalty points
- Issuing and redeeming credit memos
- Creating and updating customer records
- Creating, updating, and completing sales orders
- Receiving inventory against a purchase order or transfer order
- Performing inventory counts
- Retrieving sales transactions across stores and completing return transactions



A predefined Real-time Service profile is created.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce Data Exchange troubleshooting

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic is intended for IT personas that are implementing functionality that is related to data synchronization (Commerce Data Exchange [CDX]) in a Microsoft Dynamics 365 Commerce environment. It provides information that will help you troubleshoot CDX in implementations.

## Overview

Data configuration and synchronization are crucial to correct implementation. Regardless of business requirements, IT infrastructure, and overall preparedness, if data isn't correctly synced, the whole environment is effectively useless. Therefore, it's crucial that you understand the process of troubleshooting any issue that is related to data synchronization, from Commerce headquarters, through the Commerce Scale Unit, to the brick-and-mortar stores that use Modern POS (either with or without an offline database) and other in-store components. This topic provides some general guidance about how to mitigate or fix some issues. It also provides information about the best time to create a support request and the most important data to provide if you want to speed up a support request.

Before you go through this topic, it's important that you understand the concepts of a channel (store), registers and devices, and the Modern POS offline database. Therefore, we recommend that you review some of the resources at the end of this topic, such as the CDX implementation guide and the overview of the Commerce architecture.

## Troubleshooting

If an error that occurs doesn't appear in the following table, create a support request, as required, so that Microsoft Support can help you fix the issue. This topic is focused on issues that you can work on directly, without the help of Microsoft Support, and issues that you can directly see but can't fix without the help of Microsoft Support.

ERROR	DESCRIPTION
You receive the following error message: "System.ArgumentNullException: Value cannot be null.Parameter name: connectionString at Microsoft.Dynamics.Retail.CommerceDataExchange.SqlTargetRequestHandler."	An error has occurred because of batch job statuses. (You can see the error in a failed download job on the <b>Download sessions</b> page.) Go to <b>System administration &gt; Inquiries &gt; Batch jobs</b> , find the data writing batch that is associated with the Commerce Scale Unit that the download job was supposed to be applied to, and change the batch job's status to <b>Withhold</b> . In environments that are earlier than version 10.0.12, we recommend that you also create a channel database group that is named <b>Legacy</b> , associate the <b>Default</b> channel database with this new group, and then exclude the new database group from all distribution schedules. CDX jobs should no longer be generated for the <b>Default</b> channel database in the <b>Legacy</b> group.

ERROR	DESCRIPTION
<p>You can't perform the <b>Run now</b> command from the <b>Distribution schedule</b> page unless batch processing is used.</p>	<p>This change was intentionally made in version 10.0.11 because of performance issues that occurred if jobs were run during times when environments were most heavily used. In another change that was made as part of this feature enhancement, recurrence can't be used when the <b>Full data sync</b> command (full job synchronization) is run from the <b>Channel database</b> page in Commerce headquarters. Only a single occurrence can be run. We don't recommend that you change this behavior. However, if you're in a development environment, you can change it by going to <b>Commerce shared parameters &gt; Configuration parameters</b> and setting a new name, <b>CDX_DISABLE_FORCESCHEDULEINBATCH</b>, that has a value of 1.</p>
<p>You receive the following error message: "Microsoft.Dynamics.Retail.CommerceDataExchange.SqlWriteRequestRunException: Failed to run SqlWriteRequestRunner for table AX.&lt;TABLE NAME&gt;."</p>	<p>An error has occurred because the length of one or more <b>DBO</b> tables has been extended, so that truncation of data was required. Therefore, a truncation failure has occurred. Generate a support request. For best practices, see <a href="#">Enable custom Commerce Data Exchange synchronization via extension</a>. These best practices include removing the extended data type (EDT) extension on the table field that is being edited and using the CDX extension table to store the long (full) value that is required.</p>
<p>The download session is failing. The error message states, "...tried too many times."</p>	<p>Go to <b>Retail and Commerce &gt; Headquarters setup &gt; Parameters &gt; Commerce scheduler parameters</b>, and set the <b>Try count</b> field to 3. If the value of this field is too high, download sessions might fail during high-usage times. After you complete this step, the job will set its status to <b>Canceled</b> and stop retrying itself. We recommend that you to read <a href="#">Commerce Data Exchange best practices</a>.</p>
<p>You can't cancel a running CDX job.</p>	<p>If this issue occurs in a production environment, sign in to Microsoft Dynamics Lifecycle Services (LCS), and create a request for immediate support. If the issue occurs in a non-production environment, create a support request.</p>
<p>You use LCS to copy a Commerce headquarters database between environments, but the target environment also uses Commerce Scale Unit (Cloud) (previously known as Retail Cloud Scale Unit).</p>	<p>Create a support request, and specify whether it's acceptable to delete and reprovision the Commerce Scale Unit (Cloud), or whether the Commerce Scale Unit must be maintained as it currently exists. In both cases, downtime will be required.</p>
<p>CDX or environment issues occur after you change the value of the <b>Days transactions exist</b> field in the functionality profile.</p>	<p>If the <b>Days transactions exist</b> value is reduced by a large amount, one or more tables might become locked while purging occurs. Before you reduce the value by a large amount, we recommend that you generate a support request. Microsoft Support can remove the data before the value is changed. In this way, the database impact of the change will be minimized.</p>

ERROR	DESCRIPTION
<p>After you add multiple POS terminals, download sessions take a very long time, or there is overall Commerce headquarters slowness.</p>	<p>When you create a new Modern POS offline database and add it to the relevant channel database group, it inherits all existing download sessions since the last full database synchronization occurred. Even at the best of times, the exceptional data generation that might occur can be too large and therefore affect performance. At the worst (that is, busiest) of times, it can severely impair the environment's performance. We highly recommend that you have either a "dummy" channel database group (that is, a group that isn't associated with any distribution schedule job) that you assign to the newly generated terminals or a special offline profile where the <b>Pause offline synchronization</b> option is set to <b>Yes</b>. In this way, data generation can occur when it's required and when the system is most available to do it. (However, the system might pause multiple times as required.) If it's too late to use this approach, create a support request.</p>
<p>Normal, incremental (delta) synchronization takes much too long, even though the number of affected rows is small.</p>	<p>This issue can occur when a new channel (store) is created, because all the data must be re-created for the new store. We highly recommend that you have a "dummy" channel database that is associated with a "dummy" channel database group, and assign it to the newly generated channel (store). In this way, data generation can occur when it's required and when the system is most available to do it. If it's too late to use this approach, create a support request.</p>
<p>The P-job fails to create an upload session, and you receive the following error message: "System.Data.SqlClient.SqlException (0x80131904): Violation of PRIMARY KEY constraint 'PK_UPLOADSESSION'. Cannot insert duplicate key in object 'crt.UPLOADSESSION'."</p>	<p>If this issue occurs in a production environment, sign in to LCS, and create a request for immediate support. If the issue occurs in a non-production environment, create a support request.</p>
<p>When you try to download an upload session package from the <b>Upload sessions</b> page in Commerce headquarters, you receive the following error message: "Record for Id - &lt;Number&gt; not found."</p>	<p>Create a support request.</p>
<p>CDX download sessions fail to be applied, and you receive the following error message: "Failed to get download session package URI."</p>	<p>If this issue occurs in a production environment, sign in to LCS, and create a request for immediate support. If the issue occurs in a non-production environment, create a support request.</p>
<p>No download sessions are applied, and no upload sessions are created.</p>	<p>If this issue occurs in a production environment, sign in to LCS, and create a request for immediate support. If the issue occurs in a non-production environment, create a support request.</p>
<p>Upload sessions fail, and you receive the following error message: "Infolog for task Default:P-0001 (...) Error when bulk inserting data. Target table: RetailListingStatusLog."</p>	<p>An error has occurred because the upload session package contains multiple records in the <b>RetailListingStatusLog</b> table. These records have the same <b>StatusDateTime</b> value between two or more. If this issue occurs in a production environment, sign in to LCS, and create a request for immediate support. If the issue occurs in a non-production environment, create a support request.</p>

ERROR	DESCRIPTION
When a cashier tries to switch to offline mode or is forced offline, the switch fails.	There are many possible causes. First, verify basic information: Does the computer have available hard drive space? If you're using SQL Server Express, is the size of the offline database at the 10 gigabyte (GB) limit? Are there pending download sessions for the register? (Pending download sessions indicate that the register is no longer up to date. Therefore, offline switching might temporarily be prevented.) Additionally, we recommend that you contact Microsoft Support. If this issue occurs in a production environment, sign in to LCS, and create a request for immediate support. If the issue occurs in a non-production environment, create a support request.

## Resources

- [Commerce Data Exchange best practices](#)
- [Commerce Data Exchange implementation guidance](#)
- [Dynamics 365 Commerce architecture overview](#)
- [Select an in-store topology](#)
- [Device management implementation guidance](#)
- [Configure, install, and activate Modern POS \(MPOS\)](#)
- [Configure and install Commerce Scale Unit \(self-hosted\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Modern POS (MPOS) architecture

2/18/2021 • 2 minutes to read • [Edit Online](#)

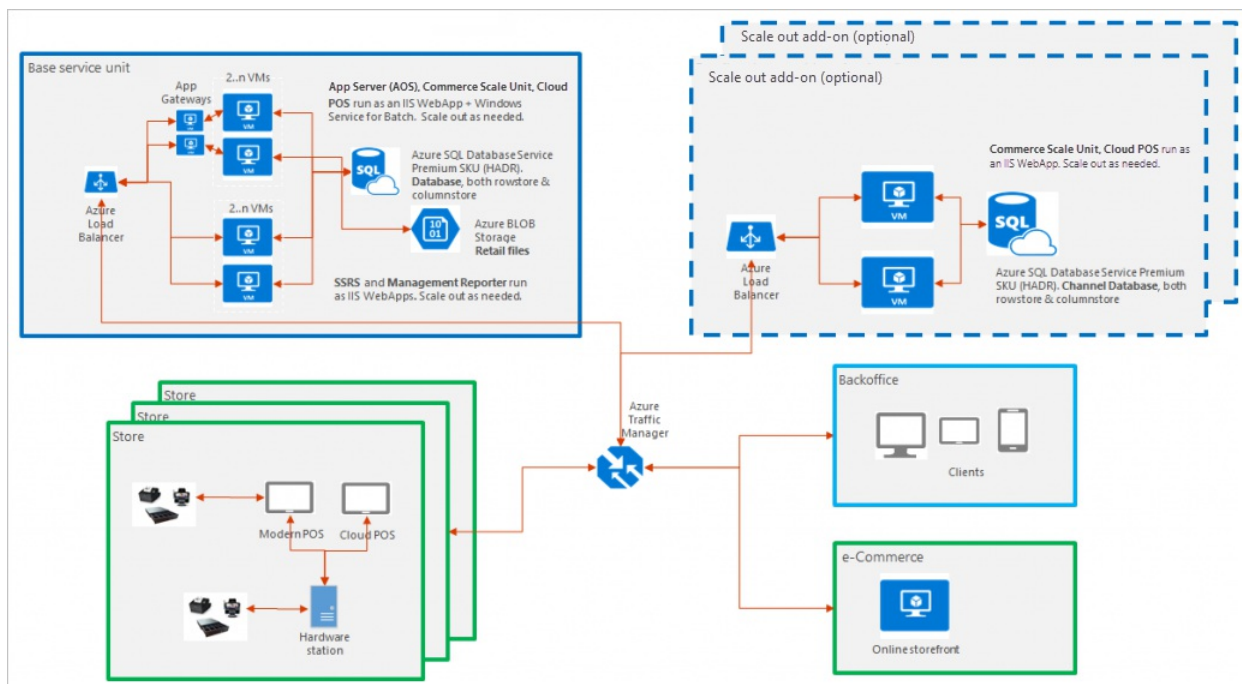
This topic describes the POS topology.

## Modern POS topology

Users of Modern Point of Sale (POS) can perform various tasks on supported laptops, tablets, and phones. These tasks include processing sales transactions, viewing customer orders, managing daily operations and inventory, and viewing role-based reports. Both MPOS and Cloud POS are available in Microsoft Dynamics 365 Commerce.

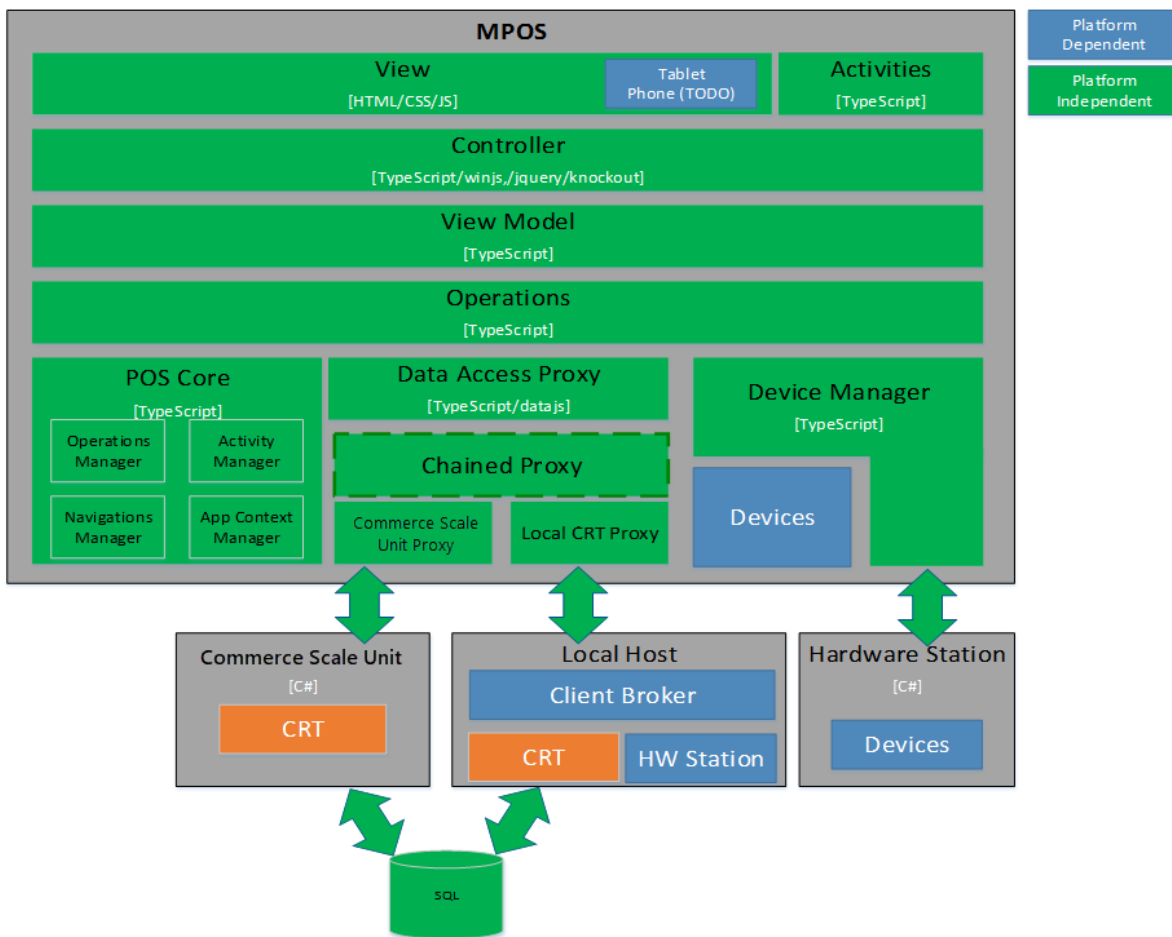
The Cloud POS is a hosted version of the POS app. Both the POS clients don't perform business functions or data processing. All business functions are provided by Commerce Scale Unit. Modern POS and Cloud POS clients can communicate with Commerce Scale Units. Modern POS client can also communicate with peripheral devices, such as cash drawers, credit card readers, and printers, by using Hardware Station. Hardware Station must be deployed in your store, and all Modern POS clients can connect to the same Hardware Station.

The following diagram shows the high-level topology.



## Modern POS architecture

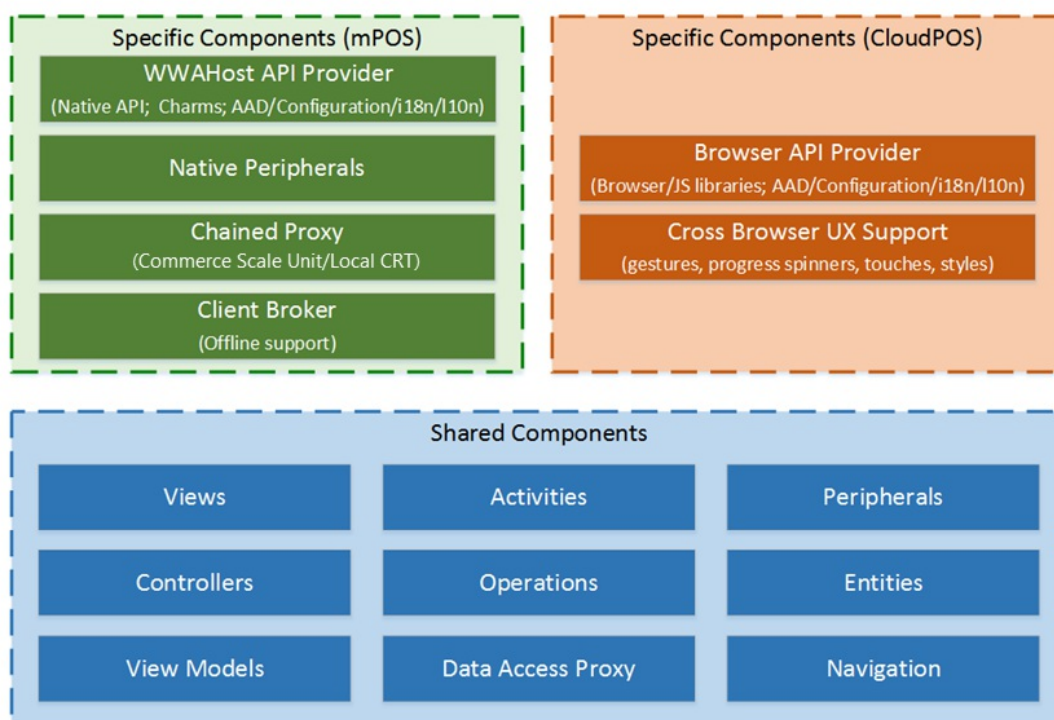
The view, view-controller, and devices layers depend on the operating system (for example, Windows RT) that you plan to deploy Modern POS on. The other layers are independent of the operating system. These layers use TypeScript classes and modules to implement Modern POS functionality such as workflows and entities. The following diagram shows the Modern POS technical architecture.



## Cloud POS and Modern POS architecture

Cloud POS is a hosted version of Modern POS, and varies only in the way that it is rendered on specific devices or in specific browsers. Additionally, Modern POS supports offline mode and therefore a local CRT. Other native peripheral support is also specific to Modern POS.

## CloudPOS v. mPOS. Shared and Specific Components



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Online store publishing architecture

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic contains conceptual information to help developers and system administrators understand how channels and catalogs are published from the commerce module to an online store in Microsoft SharePoint 2013 Products. Understanding the publishing process can help you develop, manage, and troubleshoot your online store.

## Publish an online store channel

When you publish a Commerce online store channel, you replicate the basic structure of your online store between Microsoft Dynamics 365 Commerce and Microsoft SharePoint. You create the basic structure of your online store channel in the **Commerce** module. Before you can publish an online store channel, you must complete the following setup tasks:

1. Add the online store to the organization hierarchy.
2. Create the online store and configure properties.
3. Configure the category hierarchy of your site.

After you've completed these steps, you're ready to publish the product schema to the online store.

1. You create the online store and publish it from the **Online stores** page. The status is changed from **Draft** to **In progress**.
2. Finances and Operations takes a snapshot of the category hierarchies (the Commerce hierarchy) and properties.
3. Commerce Data Exchange: Async Server reads information about the online store, hierarchies, and properties in the Commerce store database, and sends that information to the commerce runtime (CRT).
4. Async Server synchronizes the tables in the channel database.
5. The Commerce publishing job runs from the CRT application programming interface (API) and creates hierarchies for the site that you created in the online store.
6. Commerce Data Exchange: Real-time Service receives the status of the publishing job actions from the CRT API and publishes that status. The status is either **Published** or **Error**.

For the specific procedures to publish a channel, see [Set up an online store](#). After you've published the channel, you can publish a catalog.

## Publish an online store catalog

A product catalog lets you identify the products that you want to offer in your online stores. When you create a catalog, you identify the online stores where the products will be offered, add products, and enhance the product offerings by adding merchandising details. After the catalog is approved, you publish it to make products available in the online store. Before you can publish a product catalog, you must complete the following setup tasks:

1. Set up products, and configure hierarchies, assortments, and variants.
2. Set up product catalogs, and configure attribute groups and workflow.

After you've completed these steps, you're ready to publish the online store catalog.

1. Finances and Operations reads the product tables in the Commerce database.
2. Async Server synchronizes all products in the channel database.

3. The CRT/Publishing Connector creates a *listing*. A listing is an instance of a product for a channel at a given point in time. For example, you have a product that is named “jeans”, and this product has a variant that is named “red”. In this case, the system creates a listing for “red jeans”.
4. The system determines whether any new attributes were added for the listing. If new attribute were added (for example, if the “red jeans” listing includes a new attribute that is named **texture**, and this attribute is marked as **Included** at the channel level), the system creates a custom site column for that attribute. The system also creates a new rule for the list item and completes the process in SharePoint by creating a new row for the “red jeans” listing.
5. The CRT records the publishing status for the listing.
6. Async Server synchronizes the publishing status of the listing with all other publishing statuses. The status is either **Published** or **Error**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail channel performance PowerBI.com solution

2/18/2021 • 3 minutes to read • [Edit Online](#)

## NOTE

This PowerBI.com solution has been deprecated as documented in [Power BI content packs available on AppSource](#).

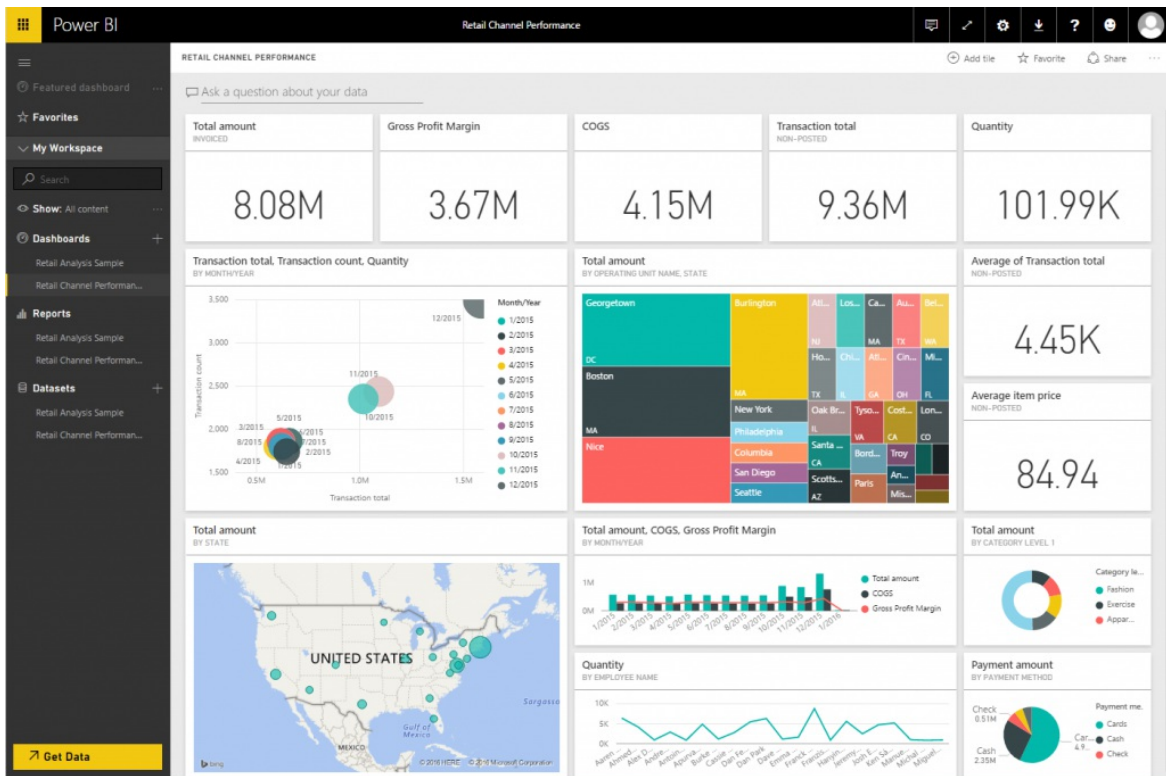
This topic provides information about the Retail channel performance PowerBI.com solution for Dynamics AX. This PowerBI.com solution lets channel managers quickly build channel performance analytics to predict trends and uncover insights, based on sales performance.

The Retail channel performance PowerBI.com solution lets you quickly build your channel performance analytics. The PowerBI.com solution is designed specifically for channel managers who focus on sales performance to predict trends and uncover insights. Its components draw directly from Retail and commerce data in the Dynamics AX database, and provide drill-down reports about organization-wide sales performance across global geography by employee, category, product, terminal, channel, and more. Power BI automatically creates reports and dashboards that give you a great starting point for exploring and analyzing your Retail and commerce data. This article includes the following information:

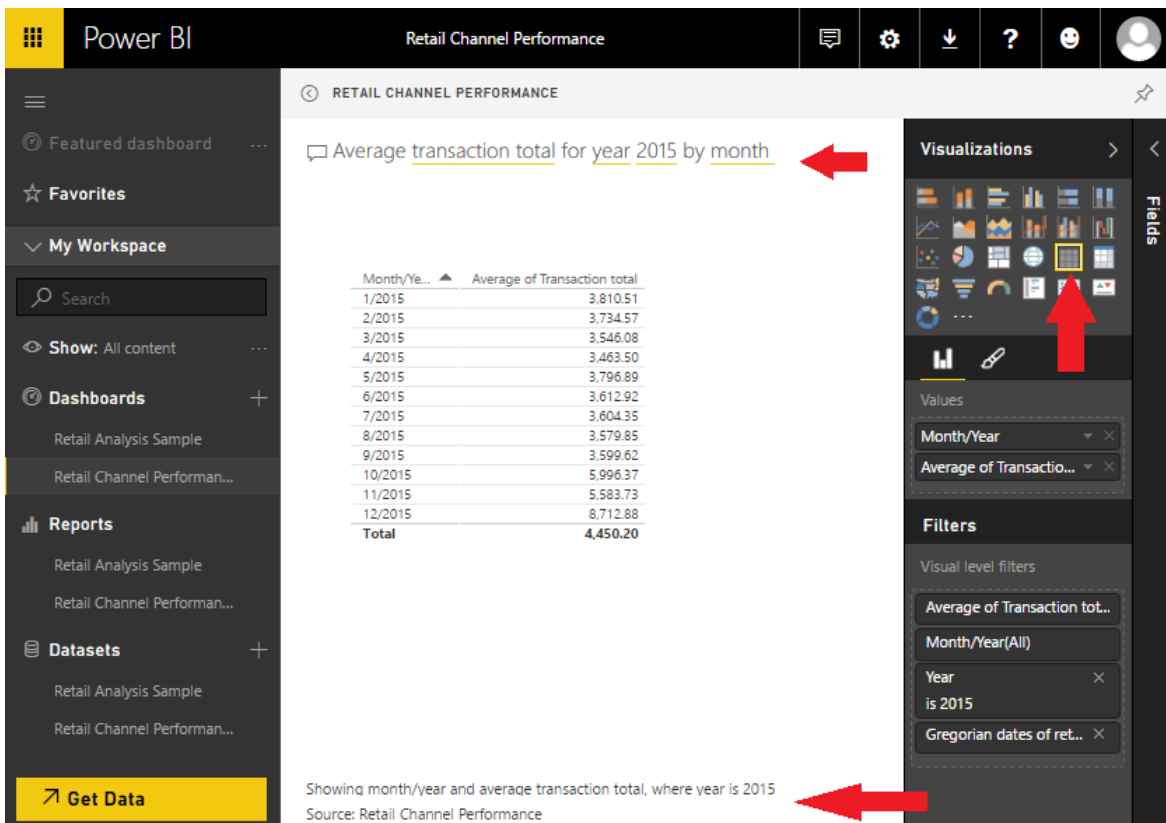
- Learn how to connect the Retail channel performance PowerBI.com solution to a Dynamics AX data source.
- View a list of reports that provide insights into retail channel performance.
- Learn how to modify an existing report in the PowerBI.com solution to make it self-authored.
- Get a glimpse of an actual data model that enables the whole experience in Power BI.

## Connect the Retail channel performance PowerBI.com solution to a Dynamics AX data source

1. Go to <https://www.powerbi.com>, and click **Sign in**. If you don't have an account, you can sign up to try the new Power BI Preview for free.
2. To sign in, enter a Microsoft 365 account that has a Power BI account.
3. If your workspace appears, click **Get Data** at the bottom of the left navigation pane.
4. In **Services** section, click **Get**.
5. Scroll or search to find **Microsoft Dynamics AX Retail channel performance**, and then click **Get it now**.
6. Enter your Dynamics AX URL in the following format: `https://<tenant>.cloudax.dynamics.com` (for example, `https://YourAOSTenant.cloudax.dynamics.com` ). Then click **Next** to pull data from Dynamics AX data storage into this Power BI dashboard.
7. Select **oAuth2** as the authentication method, and then click **Sign in**.
8. To sign in, enter a Microsoft 365 account that has permission to access your Dynamics AX environment.
9. After data is successfully pulled from Dynamics AX into Power BI, you can view your personal **Retail channel performance** dashboard in Power BI by clicking **Retail channel performance dashboard** in the left navigation pane.



10. You can then take advantage of the Q&A feature in Power BI to query your Dynamics AX sales data by using natural language.



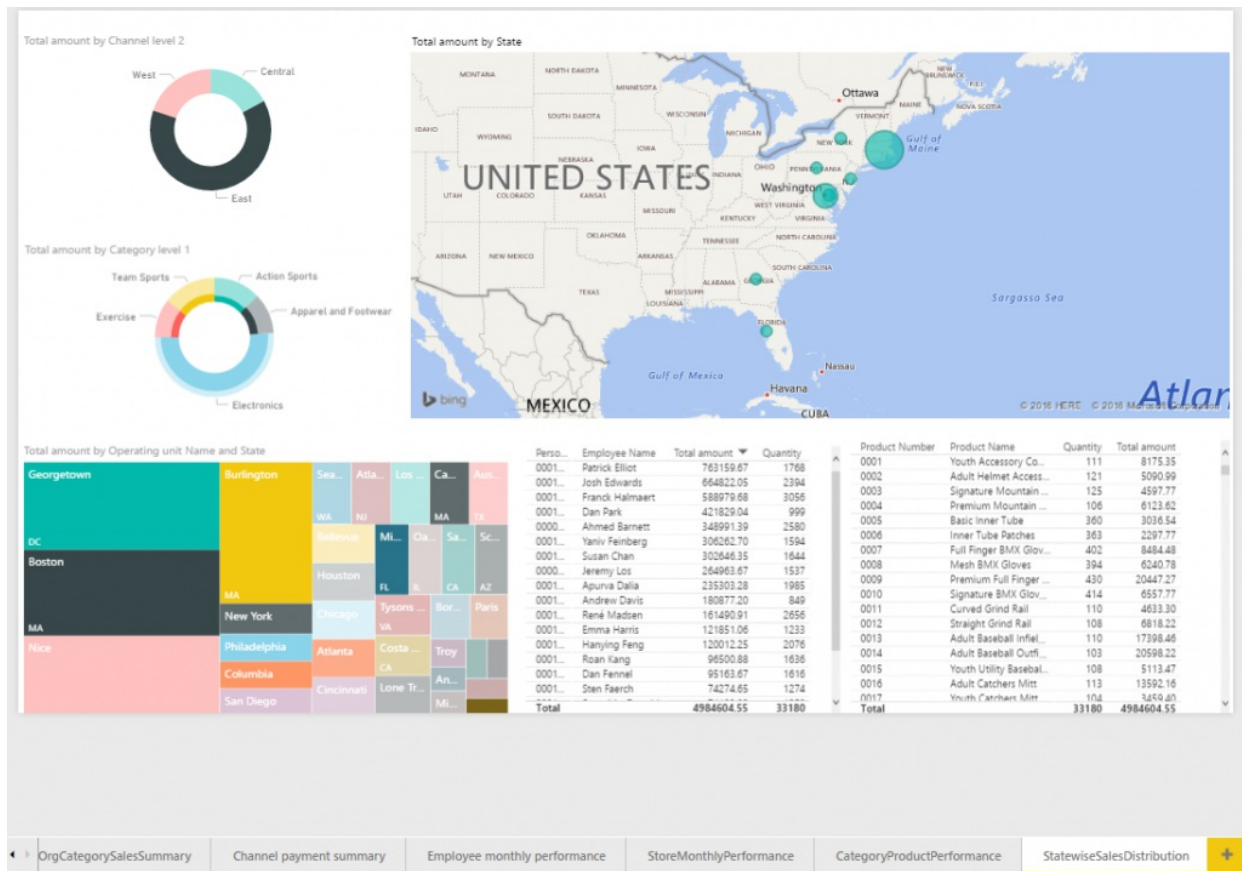
## View a list of reports

By clicking through any of the pinned tiles on the dashboard, you can navigate the following list of reports that provide insights into retail channel performance:

- Geographical sales distribution
- Category sales performance
- Sales summary by Tender type or payment method

- Employee monthly performance
- Store monthly performance
- Product sales performance for the given category in the given store

For example, you might want to do a deeper analysis of geographical sales distribution.

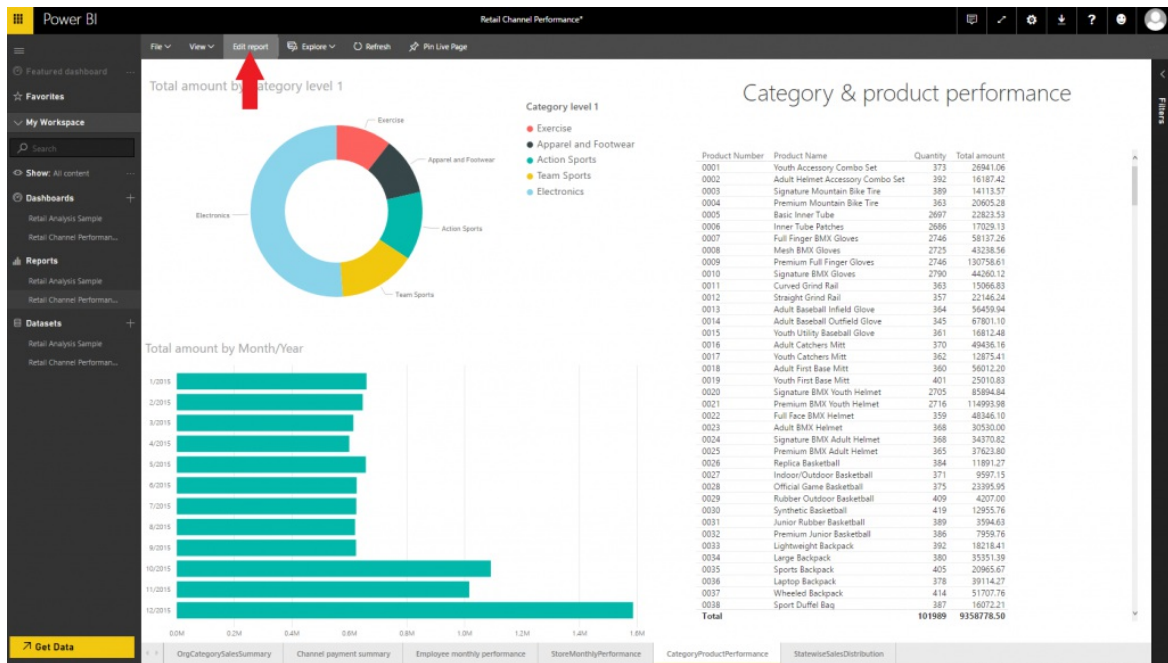


## Modify an existing report in the PowerBI.com solution to make it self-authored

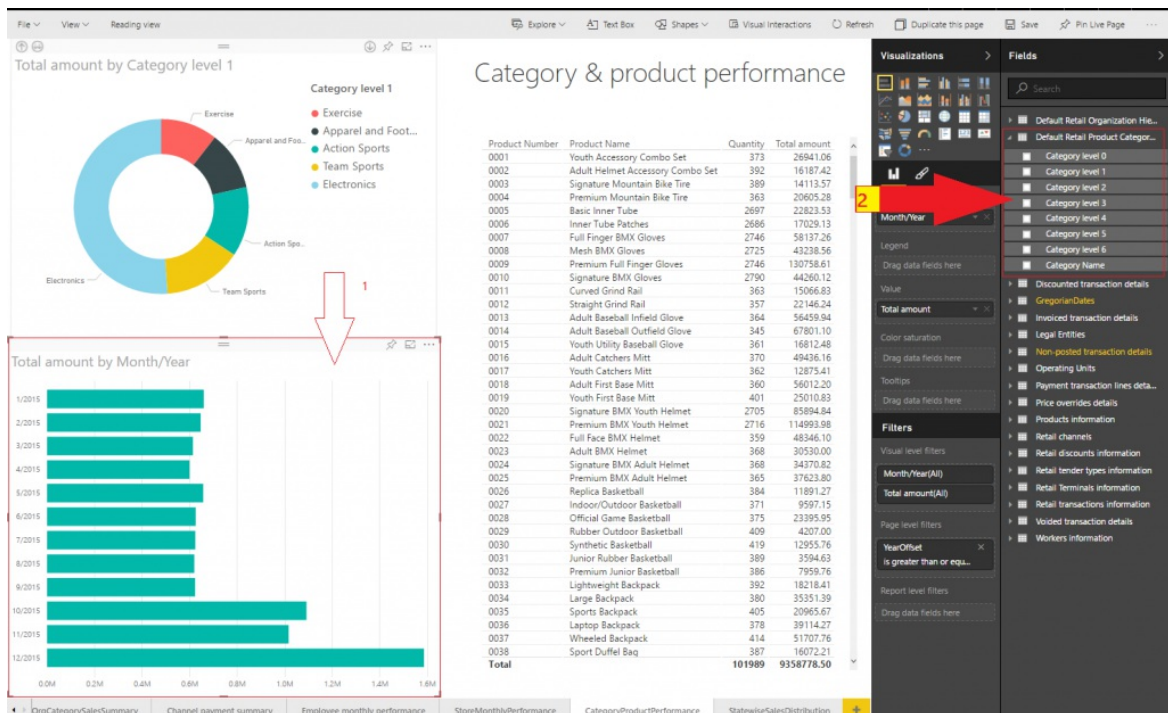
Here's an example that shows how easy it is to modify an existing report in the PowerBI.com solution to make it self-authored. In this example, we will modify an existing report that is named **Category & product performance** by adding **Category level 1** to the **Total amount by Month/Year** chart on that report.

1. Click the **CategoryProductPerformance** tab at the bottom of the window to open the **Category & product performance** report, and then click **Edit report**.

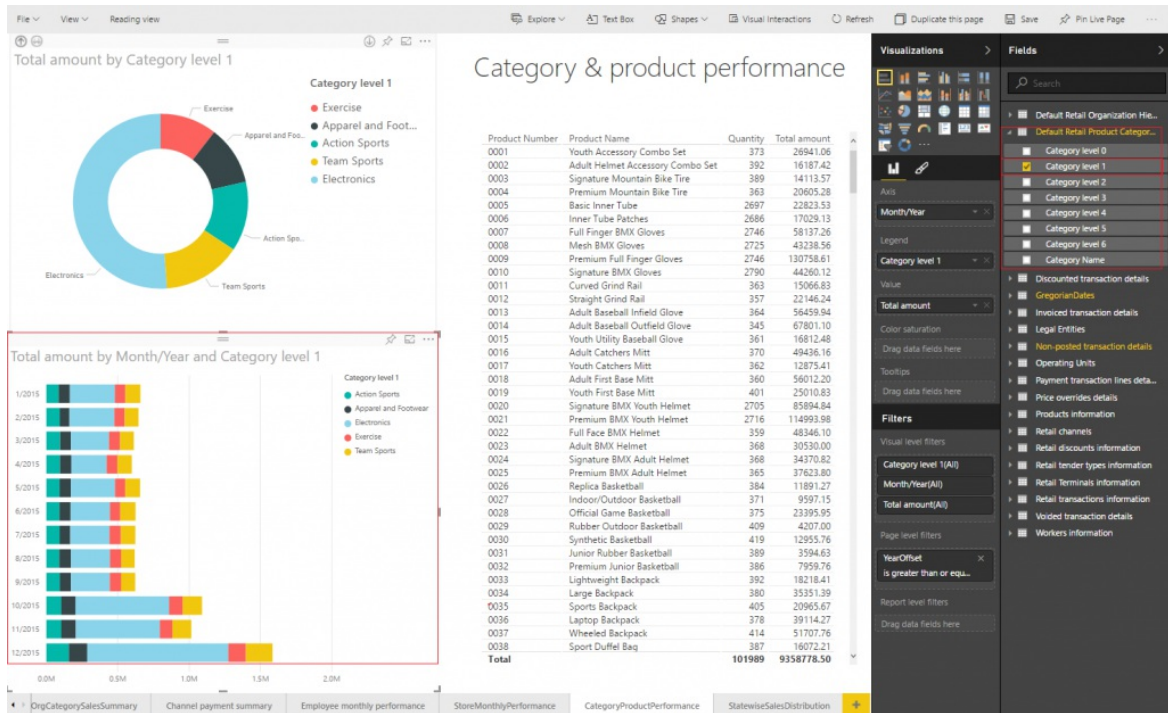




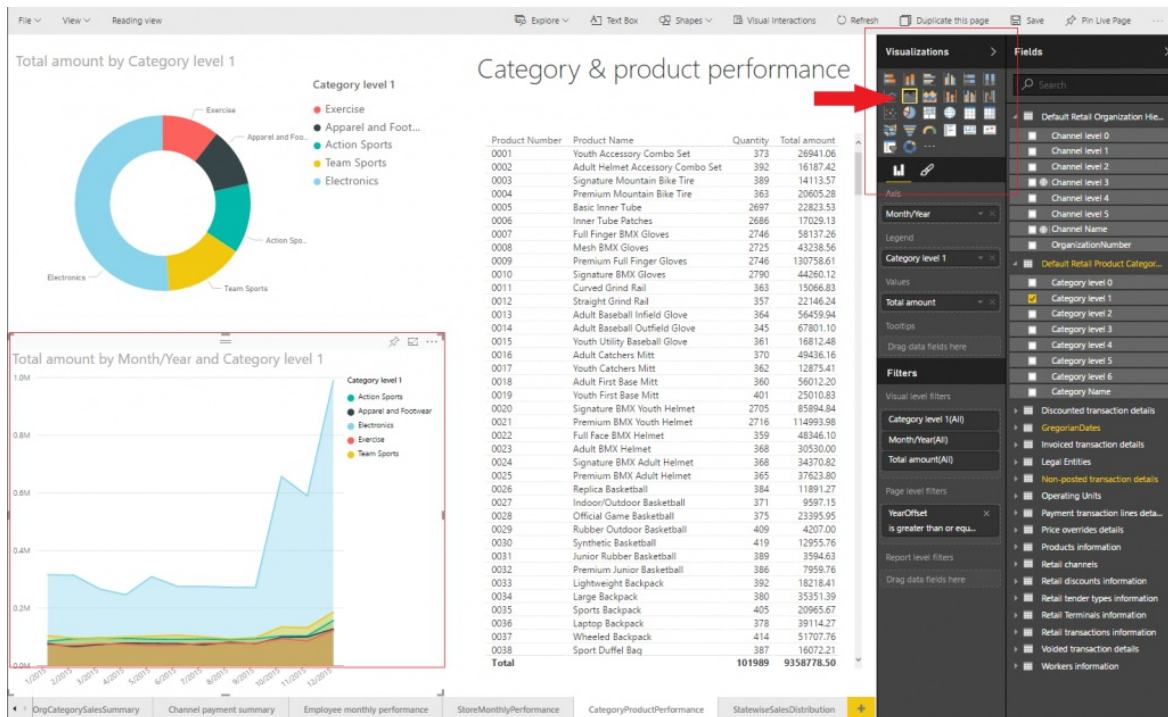
2. Select the chart that is named **Total amount by Month/Year**. Then, on the right side of the window, in the **Fields** pane, expand the **Default Retail Product Category Hierarchy** node.



3. In the list of category levels for this hierarchy, select **Category Level 1**. The name of the chart that you selected this attribute for changes to **Total amount by Month/Year and Category level 1**, and the chart now shows the share of sales in each category for each month.

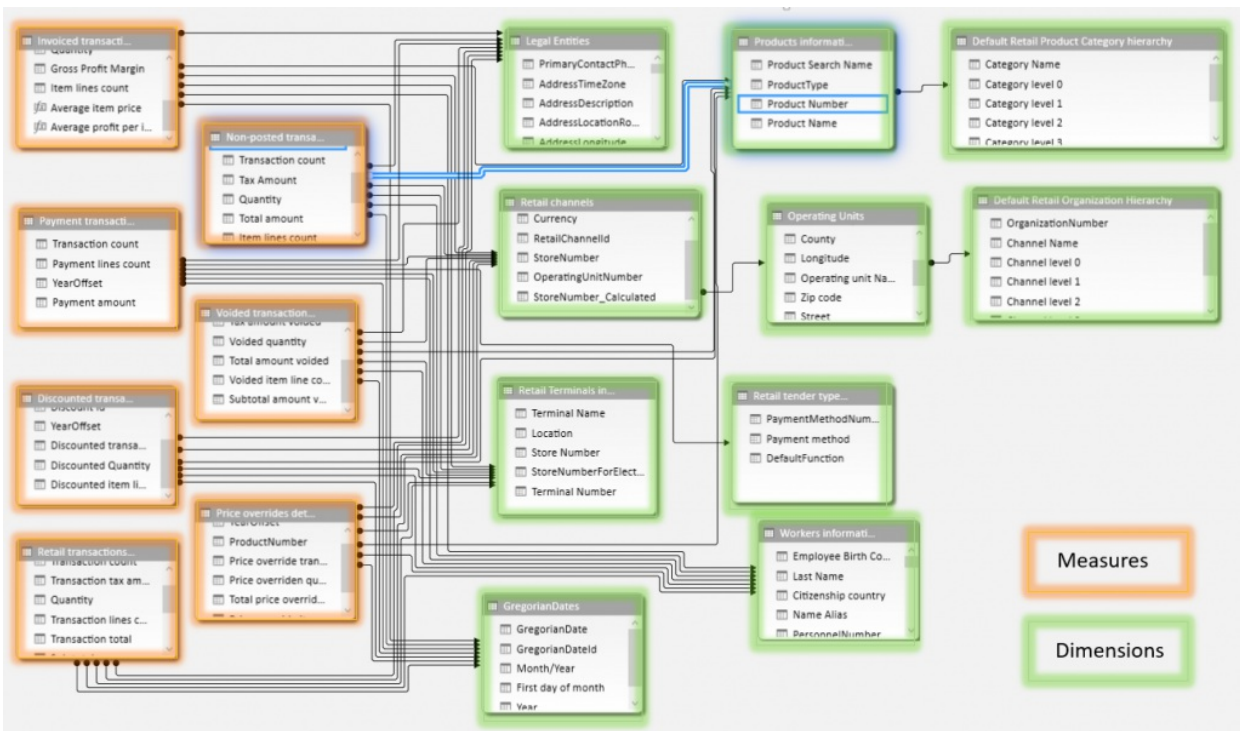


4. Finally, try to change the visualization itself. Select the **Total amount by Month/Year and Category level 1** chart, and then, in the **Visualizations** pane, click **Area chart** or **Stacked area chart**, and see the effect.



## Get a glimpse of the actual data model

The data model that is included in the PowerBI.com solution for the Dynamics AX data entities and aggregated data entities lets you slice and dice across various measures by using different dimensions.



## Additional resources

Features and services available through Power BI integration

Configure Power BI integration for workspaces

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Payment Application Data Security Standards (PA-DSS) certification

2/18/2021 • 2 minutes to read • [Edit Online](#)

## IMPORTANT

If a version of Microsoft Dynamics 365 is implemented and the PA-DSS certification has not yet been completed by Microsoft, there may be potential impact to the retailer's Payment Card Industry (PCI) audit and certification. Specifically, until the PA-DSS certification is complete for the associated version, PCI certification can become more difficult and involved as the Payment Application may come into scope of testing during the certification.

This topic explains the past, current, and pending listings of PA-DSS certification for Microsoft Dynamics 365. To view the current certifications, see [PCI PA-DSS certified payment applications](#).

PA-DSS implementation guides that are available are linked to in the **Version** column in the tables below.

## Past certification

RELEASE	VERSION	BUILD
Microsoft Dynamics AX	2012 R3	6.3
Microsoft Dynamics AX	7	7.0
Microsoft Dynamics 365 for Operations	1611	7.1
Microsoft Dynamics 365 for Finance and Operations, Enterprise edition	July 2017	7.2
Microsoft Dynamics 365 for Retail	July 2017	7.2
Microsoft Dynamics 365 for Finance and Operations	10.0	10.0

## Current certification

RELEASE	VERSION	BUILD
Microsoft Dynamics 365 Commerce	<a href="#">10.0.12</a>	10.0.12

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce capabilities that are available in on-premises deployments

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic lists Commerce capabilities that are available in on-premises deployments.

For more information about on-premises deployments, see [On-premises deployment home page](#).

COMPONENT	ON-PREMISES STATUS
Cloud Point of Sale	Available
Retail Point of Sale	Available
Modern Point of Sale	Available
Commerce Scale Unit	Available
Channel database	Available
Hardware station	Not available
AX 2012 N-1 support	Not available

## Head office capability

### Channel management

The following table indicates which channel management components are available in on-premises deployments.

COMPONENT	ON-PREMISES STATUS
All stores	Available
Online stores	Not available
Call center	Available
Registers	Available
Devices	Available
POS permissions and permission groups	Available
Info codes	Available
Payment methods	Available

COMPONENT	ON-PREMISES STATUS
Email receipts	Available
Email notifications	Available
Sales tax overrides	Available
Functionality profiles	Available
Sales tax groups	Available

### Pricing and discounts

The following table indicates which pricing and discount components are available in on-premises deployments.

COMPONENT	ON-PREMISES STATUS
All discounts	Available
Price groups, price adjustments, category price rules	Available
Trade agreements	Available
Channel navigation category	Available
Affiliation price groups	Available
Commerce channel price groups	Available
Pricing priorities	Available
Price simulator	Not available

### Customer

The following table indicates which customer components are available in on-premises deployments.

COMPONENT	ON-PREMISES STATUS
Loyalty schemes	Available
Loyalty cards	Available
Loyalty points	Available

### Merchandising

The following table indicates which merchandising components are available in on-premises deployments.

COMPONENT	ON-PREMISES STATUS
Products by category	Available
Product categories	Available

COMPONENT	ON-PREMISES STATUS
Channel categories and product attributes	Available
Assortments	Available
Commerce catalogs	Available
Commerce product kits	Available
Commerce product files	Available
Variant groups (size, color, style)	Available
Attribute, attribute type, attribute groups	Available
Vendor catalogs	Available
Bar codes	Available

### Inventory management

The following table indicates which inventory management components are available in on-premises deployments.

COMPONENT	ON-PREMISES STATUS
Packing slip	Available
Fulfilment groups	Available
Cross docking and buyer's push	Available
Qty on hand	Available
Stock counts	Available
Inventory adjustment	Available
Models of delivery	Available
Return locations	Available
Distributed order management	Not available

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Synchronize self-service installers in Dynamics 365 Commerce

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how to use the Asset library and Shared asset library in Microsoft Dynamics Lifecycle Services (LCS), and Dynamics 365 Headquarters, to upload and synchronize self-service installers so that they can be used with the standard self-service download mechanism.

## IMPORTANT

The earlier method of uploading self-service packages is currently still supported. However, it's obsolete and will be removed in the future.

## Key terms

TERM	DESCRIPTION
Shared asset library	In LCS, two types of asset libraries are available: the Shared asset library and the project-level Asset library. For more information about these libraries, see <a href="#">Asset library in Lifecycle Services (LCS)</a> .
Asset library	For more information, see <a href="#">Asset library in Lifecycle Services (LCS)</a> .
Self-service installers	Self-service installers are the Dynamics 365 Commerce components. For more information about the installers, see the links at the end of this topic.

## Overview

The **Retail Self-service package** subsection in the Shared asset library stores all monthly releases for self-service installers. These installers include Modern POS (which includes the offline version), Commerce Scale Unit (which was formerly known as Retail Store Scale Unit [RSSU]), and hardware station. You can also upload customized installers into both this library and the project-level Asset library. By using these locations, you can then synchronize the available installers in Dynamics 365 Headquarters. After synchronization is completed, all the installers that are available between these two libraries (and whatever previously existed in the environment) will be accessible for the standard self-service download processes that are described in detail in separate topics (see the links in the table of terms earlier in this topic).

The following illustration shows a generic example of the **Retail Self-service package** subsection in the Shared asset library (or Asset library).



## Shared asset library

**Select asset type**

- Asset type
- Business database (2)
- Commerce Cloud Scale Unit Extension (1)
- Configuration (0)
- Cortana intelligence application (2)
- Data package (10)
- Database backup (2)
- Deployment (0)
- Downloadable VHD (0)
- Dynamics 365 Retail SDK (0)
- e-Commerce package (0)
- GER Configuration (276)
- Localized financial report (0)
- Localized financial report 2012 (0)
- Marketing asset (0)
- Model (0)
- Model database (1)
- Model store (4)
- Nuget package (0)
- Power BI report model (2)
- Retail Self-service package (4)**
- Software deployable package (157)
- Solution package (1)

**Retail Self-service package files**

+ [Icons] PUBLISH PUBLISH AS GLOBAL VERSIONS

✓	Name	Valid	Version	Scope	Status	Modified date	Created Date
	10.0.10 - Commerce Scale...	✓	1	Global	Published	3/9/2020	3/9/2020
	10.0.10 - Modern POS	✓	1	Global	Published	3/9/2020	3/9/2020
	10.0.10 - Modern POS	✓	1	Global	Published	3/9/2020	3/9/2020
	10.0.10 - Commerce Scale...	✓	1	Global	Published	3/9/2020	3/9/2020

## Synchronize installers in Dynamics 365 Headquarters

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**.
2. On the **Channel deployment** tab, select **Check for package updates** to perform synchronization. The installers that are available for download (through standard self-service processes) are synchronized and updated, depending on which of the installers that are currently available in LCS apply to environment.

### IMPORTANT

Previously, the RetailSelfService table was used as the source that all installer information was pulled from. Information was entered in this table, based on the installers that had been uploaded into headquarters through the earlier package application method. The new self-service population methodology combines all values in the RetailSelfService table (the earlier self-service package upload method) with all available installers in the LCS Shared asset library. The self-service drop-down package selectors will show the options from this newly synchronized, combined source.

As noted at the beginning of this topic, the earlier self-service package upload method is obsolete but will continue to be supported until it's removed in the future.

3. On the same page, you can select default packages that will be used throughout headquarters in their relevant locations (**Devices, All stores, and Channel database**).
4. Perform standard configuration and installation flows for Modern POS, hardware station, or Commerce Scale Unit by using the links in the following table.

### NOTE

There are several installers. Modern POS, Modern POS with offline (note that this is a separate installer), Commerce Scale Unit (self-hosted, formerly named *Retail Store Scale Unit*), hardware station, and the less frequent installers (AX 2012 R3 support installers and the Peripheral Simulator).

COMPONENT	LINK
Modern POS	<a href="#">Configure, install, and activate Modern POS (MPOS)</a>
Hardware station	<a href="#">Configure and install Retail hardware station</a>
Commerce Scale Unit (formerly known as Retail Store Scale Unit)	<a href="#">Configure and install Commerce Scale Unit</a>

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce Scale Unit (self-hosted)

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes Commerce Scale Unit (self-hosted) and when to use it.

## Overview

Commerce Scale Unit (self-hosted) is a set of features that supports selling products in a store that has inconsistent internet connectivity to a back office or headquarters (HQ). The Store Scale Unit is designed specifically for in-store operation, and enables cross-terminal transactions and shift operations despite poor internet service. By automatically connecting to the back office, when you do have internet connectivity, your store can seamlessly process credit card transactions, issue gift cards, and sync data with HQ. The Store Scale Unit is available for download in the standard HQ deployment.

## Is Commerce Scale Unit (self-hosted) right for you?

Before you begin setting up Commerce Scale Unit (self-hosted), take a moment determine whether this option is the right fit for your store. Commerce Scale Unit (self-hosted) is a deployment choice intended for retailers with store locations that have slow or intermittent internet connectivity, and who need the flexibility of the Commerce Scale Unit deployed on premises in each store. In scenarios where a stable internet connection is available and there is low latency to the cloud environment, then it is recommended to consider operating the store as Cloud only, without setting up a Commerce Scale Unit. Consider the following before you begin:

- Carefully choose the store topology configuration for each store to either operate with a self-hosted or cloud-hosted Commerce Scale Unit topology. Reconfiguring a live store from a self-hosted to cloud-hosted Commerce Scale Unit or vice versa may cause a service disruption.
- Commerce Scale Unit will support both MPOS and Cloud POS within the store.
- Commerce Scale Unit (self-hosted) can be set up in a one-box deployment topology on a single computer (recommended) or in a multi-box topology on different computers.
- If you choose the one-box option, most of the settings are pre-configured. For a multi-box topology, you will have to manually configure connections between components.
- With Commerce Scale Unit (self-hosted), users can perform cross-terminal scenarios across multiple POS devices, like suspend/recall transactions and shift operations, even with temporary network disruption to HQ.
- With Commerce Scale Unit (self-hosted), users cannot perform any real-time operations such as issuing gift cards, looking up products, or performing credit card transactions, unless there is internet connectivity to HQ or a payment provider. If most of your transactions involve real-time transactions, then you will always need internet connectivity to enable the connection to HQ or payment provider.
- Direct database connectivity from POS to the channel database is not supported in the Commerce Scale Unit. The POS devices always use the Commerce Scale Unit for performing operations.

### NOTE

It is critical to note that Commerce Scale Unit (self-hosted) does not replace offline. Currently, Retail Modern POS with an offline database is the only way to have offline capabilities.

## Get started with Commerce Scale Unit (self-hosted)

To get started, review the following topic on configuring the Commerce Scale Unit (self-hosted), [Configure and](#)

[install Commerce Scale Unit \(self-hosted\)](#).

## Additional resources

[Configure and install Commerce Scale Unit \(self-hosted\)](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and install Commerce Scale Unit (self-hosted)

2/18/2021 • 26 minutes to read • [Edit Online](#)

This topic explains how you can use self-service to configure a Commerce Scale Unit (self-hosted, previously called Retail Store Scale Unit) in Microsoft Dynamics 365 Commerce Headquarters, download it, and install it on one or more computers in a brick-and-mortar store. Commerce Scale Unit combines the Commerce channel database, Commerce Async Client, Retail Server, and Cloud point of sale (POS) components. A Commerce environment already provides these components in the cloud. However, you can now configure them so that they work locally in a store or datacenter, in either a single-computer setup (the default option) or a multiple-computer setup. This topic also explains how to uninstall and troubleshoot Commerce Scale Unit.

## IMPORTANT

A basic design principle to follow is that if you are not able to customize in a requested manner on a Commerce Scale Unit (Cloud), you should not customize this way with a CSU (self-hosted). It is critical to understand that direct database access is not supported and can easily cause breaks in customizations that use this concept. A CSU (self-hosted) is primarily forenabling cross-terminal scenarios, reducing latency or backup for poor WAN connectivity, and providing scale-out to spread the load of POS terminals across multiple CSU components.

It is critical to note that this component utilizes a server certificate in addition to Azure Service-to-Service authentication. Both the generated Azure web application keys (formerly called *secrets*) and the server certificate must be managed for expiration. By default, a certificate and a generated Azure web application key expires in one calendar year (365 days).

## Before you begin

### IMPORTANT

To help maintain a high level of security across the company, we strongly recommend that you create a new application ID (client ID) and key (secret) for each store that is created. This step requires a new Web app.

1. Generate a Microsoft Azure Active Directory (Azure AD) app registration to create an application ID (client ID) and key (secret). For instructions, see [Create an Azure Active Directory application](#). This topic reviews Azure user permissions and requirements, and explains how to generate an app registration.

### IMPORTANT

If you are installing Commerce Scale Unit for use with an on-premises environment using Active Directory Federation Services, instead of Azure, follow the instructions in the Commerce installation document for on-premises environments. For more information, see [Installation steps for Commerce channel components in an on-premises environment](#).

2. After an application ID (client ID) and key are created for Commerce Scale Unit, the client ID must be accepted in Commerce. Go to **System administration > Setup > Azure Active Directory applications**. Enter the application ID (client ID) in the **Client ID** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.

# Configure a new Commerce Scale Unit

To create a functioning Commerce Scale Unit, complete the procedures in all sections of this topic until the "Multiple-computer installation" section. To complete the configuration and installation, you must first do the initial configuration in Headquarters. Next, you must complete the installation. Finally, you must return to Headquarters to finish the configuration, so that Commerce Scale Unit works correctly.

## IMPORTANT

Channel functionality in an on-premises environment is enabled exclusively via use of Commerce Scale Unit (self-hosted). For an overview, see [Commerce Scale Unit \(self-hosted\)](#). Unlike a cloud deployment, an on-premises environment does not enable seamless, high-availability deployment of channel components via Lifecycle Services (LCS). The only way to use channel components is by installing Commerce Scale Unit (self-hosted).

For on-premises deployments, perform the following steps:

1. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database group**.
2. On the Action pane, select **New**.
3. In the **Name** field, enter **Default**. Enter a description in the **Description** field, if needed.
4. In the **Channel schema** field, select **AX7**.
5. In the **Working folders** field, select **File storage**.
6. On the Action Pane, select **Save**.

1. In Headquarters, go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database**.
2. On the **Channel database** page, on the Action Pane, select **New**.
3. In the **Channel database ID** field, enter a unique value.
4. In the **Channel data group** field, select the **Default** option. Select any option that has been created.

## IMPORTANT

For on-premises deployments, this value will be the **Default** value that was previously described in this topic.

5. In the **Type** field, leave the default value (**Channel database**) selected.
6. You can leave the **Data sync interval** field blank. Alternatively, you can select a value in this field. For example, in the demo data, the value **D60-U15** specifies a 15-minute synchronization interval.

The **Data sync interval** value determines how often the data is synchronized between the channel database (Commerce Scale Unit) and Headquarters. If no value is entered, the default interval that is set up in Commerce Scale Unit is used. This default interval is three minutes.

7. On the **Commerce channel** FastTab, select **Add**, and then, in the **Channel** field, select the appropriate store channel. Repeat this step to add all the channels that should use this database.

You can also add channels that don't use this database. In this way, you keep the data for those channels in the Commerce Scale Unit channel database. The channels that actively use this database can then access that data locally.

### IMPORTANT

For on-premises deployments, select the **Download** button on the Action pane and select **Commerce Scale Unit package**. This will cause a known error and initiate the upload logic so that the following step in this document can be correctly completed. Allow for at least one minute to pass while the upload logic completes.

8. On the **Commerce Scale Unit package** FastTab, in the **Package reference** field, select the appropriate Commerce Scale Unit package. Each environment generates a base Commerce Scale Unit package. Therefore, this field always contains at least one option.
9. On the Action Pane, select **Save**.
10. Go to **Retail and Commerce > Channel setup > Channel profiles**.
11. On the Action Pane, select **New**.
12. In the **Name** field, enter a unique name for the channel profile.

### IMPORTANT

For on-premises deployments, any value can be entered in this field, however, the value **Default** is common.

13. On the Action Pane, select **Save**.
14. On the **Profile properties** FastTab for the new channel profile, select **Add**.
15. In the **Property key** field, select **Retail Server URL**.
16. In the **Property value** field, enter the URL of the Retail Server that will be installed.

The standard format for the URL of Commerce Scale Unit is **https://<Computer Name>:<Port>/RetailServer/Commerce**. In this format, **<Computer Name>** is either the fully qualified domain name (FQDN) of the computer where Commerce Scale Unit is installed or, for systems that aren't joined to a domain, the full computer name. **<Port>** is the port number that should be used in the installation. The port number must be a value between 1 and 65535. If you're using the default HTTPS port (443), you don't have to specify the port number.

17. On the **Profile properties** FastTab for the new channel profile, select **Add**.
18. In the **Property key** field, select **Cloud POS URL**.
19. In the **Property value** field, enter the URL of the Cloud POS instance that should be installed for Commerce Scale Unit.

The standard format for the URL of Cloud POS is **https://<Computer Name>:<Port>/POS**. In this format, **<Computer Name>** is either the FQDN of the computer where the Commerce Scale Unit is installed or, for systems that aren't joined to a domain, the full computer name. **<Port>** is the port number that will be used in the installation. The port number must be a value between 1 and 65535. If you're using the default HTTPS port (443), you don't have to specify the port number.

20. On the Action Pane, select **Save**.

#### NOTE

If media is commonly used, it will be necessary to generate a **Media Server Base URL** for the profile. For testing and simplicity, the URL that exists for the **Default** Channel profile can be reused.

For on-premises deployments, the **Media Server Base URL** will be where all media is stored for POS devices.

21. Go to **Retail and Commerce > Channels > Stores > All stores**.
22. Select the channel ID for the store that will use the new channel database.
23. On the details page for the selected store, on the Action Pane, select **Edit**.
24. On the **General** FastTab for the store, in the **Live channel database** field, select the channel database that you created in step 3.
25. On the Action Pane, select **Save**.
26. On the **General** FastTab for the store, in the **Channel profile** field, select the channel profile that you created in step 12.
27. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel data group**.
28. Select the **Default** data group, and then, on the Action Pane, select **Full data sync**. In the **Select a distribution schedule** field, select job **9999**, and then select **OK**. In the dialog box that appears, select **OK** to confirm the full synchronization. All the data in the channel database is prepared for download.

#### IMPORTANT

For on-premises deployments, there is no **Default** channel data group. Create a new data group (and associate it to the channel database and distribution schedule jobs).

#### Download the Commerce Scale Unit installer

1. In Headquarters, go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database**.
2. In the list of channel databases on the left, select the channel database that you created earlier.
3. On the Action Pane, select **Download**.
4. On the drop-down menu, select **Configuration file**.



#### NOTE

To ensure that the Commerce Scale Unit installer correctly uses the configuration file (XML file), you must save the configuration file to the same location as the installer. If the configuration file is not the same file name as the installer executable, either the executable must be run using the command line to specify the configuration file or you need to rename the XML configuration file to have the same base name as the executable file name.

For on-premises deployments, the configuration file (at this time) requires manual editing:

- StoreSystemAosUrl should have the value used to access headquarters (AX). It is critical to keep a trailing slash at the end of this URL (for example, `https://myContosoURL.com/namespaces/AXSF/`).
- AADTokenIssuerPrefix should have the value `https://NOTUSED.microsoft.com`
- TransactionServiceAzureAuthority should have the value `https://<ADFS FQDN including .com>/adfs`.
- TransactionServiceAzureResource should have the base URL value of the **StoreSystemAosUrl** edited as shown above. For instance, based on the above example `https://myContosoURL.com` would be used as the value, removing the `/namespaces/AXSF/` portion of the URL.

5. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)

Browsers might block the download pop-up that is generated. Select either **Allow once** or **Options for this site > Always allow**. Then select **Download** again.

6. On the Action Pane, select **Download**.

7. On the drop-down menu, select **Commerce Scale Unit package**.

8. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)

The correct installation package is automatically selected for download, based on the Commerce Scale Unit package on the selected channel database.

9. After the setup installer has been saved, on the Notification bar, select **Run**. (This step might differ, depending on the type of browser.)

#### Run the Commerce Scale Unit installer

#### NOTE

Before running the Commerce Scale Unit (self-hosted) installer, verify that the configuration file is named the same as the installer executable. This would look like **ExecutableInstallerName.xml** and put both files in the same folder.

Alternatively, there is a command line delimiter to specify the configuration file manually. If you plan to install and use Retail Cloud POS, you must initialize the configuration the first time that you run the installer, as described in the following procedure.

Before you run the Commerce Scale Unit installer, make sure that all [system requirements](#) are met.

#### IMPORTANT

If you are installing Commerce Scale Unit for use with an on-premises environment, you must start it from a command line using administrator privileges as follows: `StoreSystemSetup.exe -UseAdfsAuthentication`

The Commerce Scale Unit installer first extracts the associated files. It then begins the installation.

1. On the first page of the installer, select the components to install. You can install the following components:

- Commerce channel database together with Async Client
- Retail Server
- Cloud POS

**NOTE**

- To install Cloud POS, you must also select and install Retail Server. If you will use only Modern POS in the store, clear the **Retail Cloud POS** check box, and continue with the installation process as it's described here.
- By default, the installer installs all components on one computer. To install the components across multiple computers, you must complete additional manual steps. For more information, see the "Multiple-computer installation" section.

After you've selected all the components to install, select **Next** to continue.

2. The installer validates that all prerequisites are met. If a valid version of Microsoft SQL Server isn't found, the installer downloads and installs Microsoft SQL Server 2014 Express with Service Pack 2 (This downloading feature is deprecated and it is critical to download and install the correct version of Microsoft SQL Server for your implementation).

**NOTE**

- To meet the prerequisites, SQL Server must have full-text search, and it must support, at a minimum, Transport Layer Security (TLS) 1.2. For Microsoft SQL Server 2014, Service Pack 2 must be installed. For Microsoft SQL Server 2016, Service Pack 1 is the minimum required. For Microsoft SQL 2017, manual installation of the CLR types must be performed. It is critical to install a supported version of Microsoft SQL Server.
- If a system restart is required, the installer will prompt the user. This prompt is based upon a Windows system registry key that tells all applications if a restart is required. While it is recommended to restart prior to continuing the installation, a restart is not mandatory and the installer can continue without restarting the computer.

3. Verify the URL for Application Object Server (AOS), and then select **Next**. (The AOS URL is the URL that is used to access Headquarters.)

**NOTE**

The Retail Server URL is automatically entered from the configuration file.

4. Select a valid Secure Sockets Layer (SSL) certificate to use for HTTPS communication.

The certificate must use private key storage, and server authentication must be listed in the enhanced key usage property. Additionally, the certificate must be trusted locally, and it can't be expired. It must be stored in the personal certificate store location on the local computer.

5. If a specific user is required, enter the user name and password that the application pool should run under. By default, the installer automatically generates a service account to use. This approach is more secure and is recommended.

#### NOTE

It is important to note that service accounts, out of box, still function under the same password policy that is defined for all other accounts. This means that the minimum password age policy still applies to the Retail Server service account and must be updated when necessary. By default, on Windows Server 2012 R2, this is typically 42 days. If the password does expire on a used service account, the Commerce Scale Unit components will fail to continue functioning until the issue is resolved.

6. On the next page, enter the user account and password for the Retail Server application pool and Async Client. By default, this account is automatically generated. However, you can manually enter the user account and password.
7. Enter the HTTPS port to use, and verify that the host name of the computer is correct. Then select **Next** to continue.

#### NOTE

- The installer automatically enters the host name. If, for any reason, the host name must be changed for the installation, change it here. The host name must be the FQDN of the system, and it must be entered in the **Host name** field for the selected Store system entry.

8. Enter the application ID (client ID) and key (secret) that are associated with this Commerce Scale Unit installation. Additionally, verify the channel database ID, which is automatically entered from the configuration file. Then select **Install**. If you will use Retail Cloud POS, make sure that the **Configure Retail Cloud POS** check box at the bottom of the page is selected. This configuration requests Azure AD sign-in and automatically generates all required information in Azure, so that Retail Cloud POS can be used on-premises. If you are installing Commerce Scale Unit for use with an on-premises environment, you must clear this option.

For information about how to create web applications in Azure, see [Create an Azure Active Directory application](#).

#### IMPORTANT

- When installing Commerce Scale Unit for use with an on-premises environment, Cloud POS does not require an Azure or AD FS application to be configured, so it is important to unmark **Configure Retail Cloud POS**.
- When installing Commerce Scale Unit for use with an on-premises environment, the Client ID (Application ID) and Secret (Key) used will be the values generated by the PowerShell script performed in the configuration steps performed in steps 6-8 in the [Installation steps for Commerce channel components in an on-premises environment](#) topic. (Step 6 creates the Client ID and step 8 resets the Secret to be copied.)

When you create the Web App, the initial URI and URL don't have to be any specific value. Only the application ID (client ID) and key (secret) that are created are important.

9. After the application ID (client ID) and key (secret) are created for Commerce Scale Unit, the application ID (client ID) must be accepted in Commerce. Follow the next procedure to finish the configuration in Headquarters.
10. After the installation is complete, the final health page appears. This page shows whether the installation was successful. It also shows the health of each component, based on basic connection tests, and the location of this topic. If the installation wasn't successful, the page shows the location of the log files. We recommend that you keep this final health page open until you've completed the configuration of Commerce Scale Unit and all components are working correctly (Requiring the completion of the

following section).

### Finish the configuration in Headquarters

The last steps require validation and verification that the Azure application ID (client ID) and key (secret) are correctly accepted in Headquarters, so that connections can be made between the environment and the new Commerce Scale Unit.

1. After the application ID (client ID) and key (secret) are created for Commerce Scale Unit and entered in the installer, the application ID (client ID) must be accepted in Headquarters.

In Headquarters, go to **System administration > Setup > Azure Active Directory applications**. Enter the application ID (client ID) in the **Client ID** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.

2. If Cloud POS is configured for use, a client ID is shown at the end of the installation. You must add this client ID to the **Commerce shared parameters** page in Commerce.

#### IMPORTANT

In an on-premises environment, this step is not required to be completed.

- a. In Headquarters, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
  - b. Select **Identity providers**.
  - c. On the **Identity providers** FastTab, select the provider that begins with `HTTPS://sts.windows.net/`. The values on the **Relying parties** FastTab are set, based on your selection.
  - d. On the **Relying parties** FastTab, select **Add**. Enter the client ID that is listed on the final health page of the Commerce Scale Unit installer. Set the **Type** field to **Public** and the **UserType** field to **Worker**. Then, on the Action Pane, select **Save**.
  - e. Select the new relying party, and then, on the **Server resource IDs** FastTab, select **Add**. In the **Server Resource ID** column, enter `https://retailstorescaleunit.retailserver.com`.
  - f. On the Action Pane, select **Save**.
3. In Headquarters, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
  4. Select **Identity providers**.
  5. On the **Identity providers** FastTab, select **Add**.
  6. In the new **Issuer** row, enter the URL of the newly installed Commerce Scale Unit. At the end of the URL, add **/auth**. The URL will resemble

`https://<My Case-Sensitive Computer Name>:<Port Number>/RetailServer/auth`.

#### NOTE

The URL described above is case sensitive. There will be a new identity provider line for each Commerce Scale Unit that is installed. Each will have a URL that resembles this URL.

7. In the **Name** column, enter a description for the store that the URL belongs to.
8. In the **Type** column, select **Open ID Connect**.

#### NOTE

This new row must be duplicated for every Commerce Scale Unit installation (that is, for every unique URL).

9. On the Action Pane, select **Save**.
10. On the **Identity providers** FastTab, select the newly created line. The values on the **Relying parties** FastTab are set, based on your selection.
11. On the **Relying parties** FastTab, select **Add**, and add the following two entries:
  - In the **ClientId** column, enter **Cloud POS**. Set the **Type** field to **Public** and the **UserType** field to **Worker**.
  - In the **ClientId** column, enter **Modern POS**. Set the **Type** field to **Public** and the **UserType** field to **Worker**.
12. On the Action Pane, select **Save**.
13. In Commerce, go to **Retail and commerce > Retail and commerce IT > Distribution Schedule**, and run CDX Job 1110.
14. When you've finished, return to the installer, and select **Finish**.

The final page of the installer includes valuable information that you can use to test and validate that all components work correctly. Keep this page open until you've completed the validation.

#### NOTE

If the installer doesn't show a check mark for Retail Server or Async Client, wait 10 minutes so that any cached values can be updated in the cloud. Then check again. If the installer still isn't fully successful, run a full synchronization on the new channel database that this installation uses.

If you followed all the steps correctly, your configuration should have these characteristics:

- In Azure, two web applications have been automatically generated through the installer:
  - Retail Store Scale Unit Cloud POS
  - Retail Store Scale Unit Retail Server for Cloud POS
- In Azure, a web application (that is, an App registration in the new Azure portal) has been manually created for each Commerce Scale Unit installation (for example, CommerceScaleUnitHouston). A key (secret) has been created that can be used in the installer, as described earlier.
- The application ID (client ID) of the manually created web application has been added to the **Azure Active Directory applications** page in Commerce, as explained in step 1 of the preceding procedure.
- The Cloud POS application ID (client ID) that was shown at the end of the Commerce Scale Unit installer has been added on the **Identity providers** FastTab, as explained in the final steps of the "Run the Commerce Scale Unit installer" section.

### Multiple-computer installation

Only advanced users should install Commerce Scale Unit across multiple computers. The following set of procedures explains how to install the Commerce channel database and Async Client on one computer, and Retail Server and Cloud POS on a second computer. The instructions assume that both systems are on the same domain, and that users for the services that will be installed have already been created on both systems. It's important that you do all configuration in Headquarters.

#### Installation on the first computer

On the first computer, run the Commerce Scale Unit self-service installer as described earlier in this topic, but

make the following changes.

1. Select only Commerce channel database and Async Client as the components to install. Then select **Next** to continue with the installation.

**NOTE**

You can use a generated service account for Async Client, because Async Client won't be accessed outside the computer that it's installed on.

2. Enter the client ID and key (secret). Keep these details available, so that you can use them again on the second computer.
3. After a client ID and key (secret) are created for Commerce Scale Unit, the client ID must be accepted in Commerce. Go to **System administration > Setup > Azure Active Directory applications**. Enter the client ID in the **Client ID** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.
4. When setup is successful, start SQL Server Configuration Manager.
5. Go to **SQL Server Network Configuration > Protocols** for the SQL Server instance.
6. Right-click, and then select **Properties**.
7. In the **Flags** section, change the value of **Set Force Encryption** to **Yes**.
8. In the **Certificate** section, select the SSL certificate on the drop-down menu. This SQL Server SSL certificate is the same certificate that is used in the installer.
9. Select **OK**.
10. Go back to **Protocols** in SQL Server Configuration Manager, and enable the following protocols:
  - Named Pipes
  - TCP/IP
11. Right-click the **TCP/IP** protocol, and then select **Properties**.
12. In the **IP Address** section, scroll down the list to **IPALL**.
13. Enter **TCP Port = 1433**.
14. Select **OK**.
15. Start Microsoft Windows Firewall with Advanced Security.
16. In Windows Firewall, create an inbound rule to open TCP port 1433.

For detailed information about SQL Server and Windows Firewall, see [Configure a Windows Firewall for Database Engine Access](#).

**Installation on the second computer**

On the second computer, run the Commerce Scale Unit Self-service installer as described earlier in this topic, but make the following changes.

1. Select only Retail Server and Cloud POS as the components to install. If only Retail Server must to be installed, don't select Cloud POS. Then select **Next** to continue with the installation.
2. Enter the domain user credentials (user name and password) that have permission to access SQL Server on the first computer. Then select **Next**.

**NOTE**

A generated service account can't be used, because Retail Server requires access to the SQL database on the first computer. You must use a domain account.

3. Enter the same Client ID and key (secret) that are used on the first computer.

**IMPORTANT**

It's critical that you add this Client ID to Commerce headquarters as described earlier.

4. Select **Configure Cloud POS**, and then enter Azure AD credentials that have the correct permissions to create Azure Web Apps.

For more information about Azure Web Apps, how to create them, and how to generate a new key (secret), see [Use portal to create an Azure Active Directory application and service principal that can access resources](#). Note that the sign-in URL and the App ID URI are not important.

5. When setup is successful, don't exit the installer.

**NOTE**

At first, the health check ping won't be successful, because the database isn't yet set up correctly. After you've completed the remaining steps of this procedure, you can test the health check again.

6. Start **Microsoft Internet Information Services (IIS)**, select the **Retail Server** website, and select the **Retail Server** web application.
7. Explore the working directory.
8. Open the Web.config file, and then, in the **connectionStrings** section, add **Server name**. **Server name** is the name of the first computer where you installed components. Save the file.
9. If the certificate that is used isn't a valid, trusted certificate from a trusted authority, open CERTMGR.MSC, and follow these steps:
  - a. Import the SQL Server SSL certificate that you created earlier, and add it to Trusted Root.
  - b. Open a **Command Prompt** window as an administrator, type **IISRESET**, and then press Enter.
10. If Cloud POS is configured for use, a client ID is shown. You must add this client ID to the **Commerce shared parameters** page.
  - a. In Commerce, go to **Retail and commerce > Headquarters setup > Parameters > Commerce shared parameters**.
  - b. Select **Identity providers**.
  - c. On the **Identity providers** FastTab, select the provider that begins with `HTTPS://sts.windows.net/`. The values on the **Relying parties** FastTab are set, based on your selection.
  - d. On the **Relying parties** FastTab, select **Add**. Enter the client ID that is listed in the Commerce Scale Unit installer. Set the **Type** field to **Public** and the **UserType** field to **Worker**. Then, on the Action Pane, select **Save**.
  - e. Select the new relying party, and then, on the **Server resource IDs** FastTab, select **Add**. In the **Server Resource ID** column, enter `https://retailstorescaleunit.retailserver.com`.
  - f. On the Action Pane, select **Save**.
11. In Commerce, go to **Retail and commerce > Headquarters setup > Commerce scheduler >**

**Channel database**, and follow these steps:

- a. Select the channel database that you created at the beginning of this topic.
  - b. On the Action Pane, select **Full Sync > Job 9999**. Full synchronization might require several minutes.
  - c. In the Commerce Scale Unit installer, retest to verify that all functionality is working correctly.
12. Start Cloud POS from the computer that you're using for POS operations. (This computer is a third computer that is separate from the Commerce Scale Unit systems.)
13. Activate the Cloud POS device that you're using for this computer.
14. Do a simple sale to verify full end-to-end functionality.

### Help secure Commerce Scale Unit

According to current security standards, the following options should be set in a production environment:

- You should completely disable SSL (v2 and v3) on the computer.
- You should enable and use only TLS version 1.2 (or the current highest version).

#### NOTE

By default, SSL and all versions of TLS except TLS 1.2 are disabled. To edit or enable these settings, follow these steps:

1. Press the Windows logo key+R to open a **Run** window.
2. In the **Open** field, enter **Regedit**, and then select **OK**.
3. In the **User Account Control** window, select **Yes** (if this step is required), and then, in the new **Registry Editor** window, go to **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\SecurityProviders\SCHANNEL\Protocols**.
4. The following keys have been automatically entered to enable only TLS 1.2:
  - TLS 1.2\Server:Enabled=1
  - TLS 1.2\Server:DisabledByDefault=0
  - TLS 1.2\Client:Enabled=1
  - TLS 1.2\Client:DisabledByDefault=0
  - TLS 1.1\Server:Enabled=0
  - TLS 1.1\Client:Enabled=0
  - TLS 1.0\Server:Enabled=0
  - TLS 1.0\Client:Enabled=0
  - SSL 3.0\Server:Enabled=0
  - SSL 3.0\Client:Enabled=0
  - SSL 2.0\Server:Enabled=0
  - SSL 2.0\Client:Enabled=0

- No additional network ports should be open, unless they are required for known, specified reasons.
- You must disable cross-origin resource sharing, and you must specify the allowed origins that are accepted.
- You should use only trusted certificate authorities to obtain certificates that will be used on Commerce Scale Unit computers.

#### IMPORTANT

It's critical that you review security guidelines for IIS and the Payment Card Industry (PCI) requirements.



## Troubleshoot Commerce Scale Unit

Here is a checklist of things to verify:

1. Open the configuration file and verify that the Retail Server URL specified contains the suffix **/Commerce** and is correctly formed based on what is expected for the machine name and port used. Validate that there is no trailing or additional slash (the character **/**) in the URL or at the end of it.
2. In Commerce, on the **Commerce shared parameters** page, verify that the correct client ID has been added to the **Relying parties** FastTab. Additionally, verify that the correct `https://retailstorescaleunit.retailserver.com` entry has been added to the **Server resource IDs** FastTab.
3. In Commerce, verify that every client ID that was generated for each store exists on the **Azure Active Directory applications** page.
4. In Commerce, on the **Channel profile** page, verify that the URLs are correct. To do this, verify that the computer name in each URL is correct, the URL is correctly formatted, and so on.
5. In Commerce, on the **Channel database** page, verify that full synchronization correctly occurred for the new channel database.
6. Verify that the store is correctly configured to use the new channel database.

If the Retail Server stops functioning after a period of time, there are two simple things to verify:

- Verify if the password policy requires the service account that was generated to change the password (password expiration).
- Re-run the same installer over the current installation (idempotent), which will update a service account's password or allow the user to update the selected account's password.

## Uninstall Commerce Scale Unit

Use Control Panel in Microsoft Windows to uninstall Commerce Scale Unit components.

1. Press the Windows logo key, and then enter **Control Panel** in the search box. In the search results, select **Control Panel**.
2. In Control Panel, select **Programs > Uninstall a program**.
3. In the **Programs and Features** window, select **Microsoft Dynamics 365 Commerce Scale Unit**, and then, above the list of programs, select **Uninstall**.
4. Wait for the uninstaller to finish removing the program.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure, install, and activate Modern POS (MPOS)

2/18/2021 • 21 minutes to read • [Edit Online](#)

This topic describes how to configure, download, and install Modern POS on various platforms. It then describes how to activate Modern POS through device activation.

## NOTE

There are two Modern POS installers: Modern POS and Modern POS with offline (this installer also installs the offline database).

Starting in release 10.0.11, altering customized files that are stored in the ClientBroker folder could cause issues when installing a newer release. These issues might include the inability to go offline or a newer installer failing to complete successfully. A workaround is to remove the files in the ClientBroker folder in the Modern POS directory before performing the installation using the newer installer.

Starting in 10.0.15 release, customizations to files in the Client broker folder for Modern POS can cause an error when updating from a previous version. The known workaround is to delete all files from the Client broker folder prior to running the newer Modern POS installer. For automation, this can easily be scripted as a pre-step for the installer. All files in this folder must be deleted. When this error occurs, the newer installer will update the current installation correctly.

## Technology

The self-service process lets you download the appropriate version of the Modern POS installer and install it on the physical device that you want to use as the point of sale (POS) register. Device activation is the main onboarding step that ties the physical device to a register in Headquarters. Here are the main technical functions of this feature:

- Tie a physical device to a business entity (register).
- Provide enhanced security through Microsoft Azure Active Directory (Azure AD) and a device token/ID.
- Stop unauthorized remote use of Modern POS. (In other words, deactivate a device remotely.)
- Initialize settings for easy Modern POS functioning (number sequence, hardware profile, merchant information) as the first touchpoint of the POS.
- Comply with payment card industry (PCI) standards, and report on device information from Headquarters.

## NOTE

If you are installing Modern POS for use with an on-premises environment, Modern POS does not use Azure Active Directory credentials for device activation.

## Setup

Before you start the steps that are outlined in this topic, follow these steps.

- Verify that you have credentials to sign in to Headquarters.
- Verify that you have administrative or root access to install Modern POS on a device.
- Verify that you can access the Commerce Scale Unit from the device.
- Verify that the environment contains the Commerce permission groups and jobs in the **Human resources**

module. These permission groups and jobs should have been installed as part of the demo data.

## Download and install Modern POS

### Verify that the device is correctly configured

1. In Headquarters, go to **Retail and Commerce > Channels > Channel deployment**.
2. On the **Channel deployment** page, select the **Registers** tile.
3. On the **Registers** page, select a store register. The demo data thoroughly defines the Houston store and registers for self-service. To find the Houston registers, enter **Houston** in the filter at the top of the list of devices.
4. Select a register by selecting the register number in the **Register number** column. In the Houston store, register Houston-3 is well defined and is therefore useful as an example.
5. On the page for the register, under **General**, verify that the **Support offline** option is set to **No**. To use offline support, on the Action Pane, select **Edit**, and then set **Support offline** option to **Yes**.

### Download the Modern POS installer

1. On the **Welcome** page, use the menu in the upper left to go to **Retail and Commerce > Channels > Channel deployment**.
2. On the **Channel deployment** page, select the **Devices** tile.
3. Select a device.

#### NOTE

- The Houston devices are well defined. Houston-3 is useful as an example for a Microsoft Windows desktop or tablet. Houston-21 is useful as an example for a Windows Phone.
- When you select a device, the **Download** button on the Action Pane becomes available.

4. Select **Download**, and then select **Configuration file**. Note the following:
  - Browsers might block the download pop-up that is generated. You must select either **Allow once** or **Options for this site > Always allow**. Then, while the device is still selected, select **Download** again.
  - The configuration file must be saved to the same location as the Modern POS installer. For security reasons, delete this file after installation is completed. If the configuration file is not the same file name as the installer executable, either the executable must be run using the command line to specify the configuration file or you need to rename the XML configuration file to have the same base name as the executable file name.
5. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)
6. Select **Download**, and then select **Retail Modern POS**. Note the following:
  - Browsers might block the download pop-up that is generated. You must select either **Allow once** or **Options for this site > Always allow**. Then, while the device is still selected, select **Download** again.
  - The installation package that you must use varies, depending on whether you require offline support, and whether the device that Modern POS will be installed on is a Windows tablet or a phone device (such as a Windows Phone, an Android device, or an iOS device). The correct package is automatically selected for download, based on the register settings and the application type that is set for the device. If the offline package is selected for a Windows tablet, but Microsoft SQL Server isn't already installed

(or if it doesn't meet the requirements for the offline package), SQL Server is downloaded and installed silently.

7. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)
8. After the setup installer has been saved, on the Notification bar, select **Run**. (This step might differ, depending on your browser.)

### Before running the Modern POS installer

- Make sure that all [system requirements](#) are met.
- It is recommended to temporarily turn off antivirus applications. It has been noted that on aggressive antivirus solutions, the installation may stall due to the antivirus solution checking active files while in use.
- The installer will sideload a modern application. Therefore, a Group Policy entry must be set to allow for sideloaded applications. The installer will change the associated registry key as follows to allow for this installation:
  - **Path:** HKLM:SoftwarePoliciesMicrosoftWindowsAppx
  - **Property:** AllowAllTrustedApps
  - **Value:** 1
- If offline is used (an offline database created), then a default SQL Server instance must exist. If SQL Server instances exist, but none are set as the default, then the installer will fail to install the offline database.

If you are installing Modern POS for use with an on-premises environment, you must start the installer from a command line as follows:

```
ModernPosSetupOffline.exe -UseAdfsAuthentication
```

### Run the Modern POS installer on a Windows computer

The Modern POS installer first extracts the associated files and then starts the installation.

1. The installer validates that all prerequisites are met. Note the following:
  - If a system restart is required, the installer informs you about this requirement, but the installation can typically continue.
  - A sideloaded installation of Modern POS requires a Group Policy change. The installer informs you if this change is required and then makes the change automatically.
2. If you selected offline support, but a valid version of SQL Server isn't found, the installer downloads and installs Microsoft SQL Server 2014 Express with Service Pack 2 (SP2). To meet the prerequisites, SQL Server must have Full-text search installed. Additionally, a minimum of SP2 must be installed for Microsoft SQL Server 2014, or a minimum of Service Pack 3 (SP3) must be installed for Microsoft SQL Server 2012. Note the following:
  - The installer tries to download the correct language. However, if you require a specific language, we highly recommend that you manually install SQL Server. If the installer can't correctly determine the language, it installs the English version of SQL Server 2014 Express with SP2 by default. Typically, after the SQL installation is completed, the system requires a restart before the installation of Modern POS can continue.
  - This process might require a long time, depending on the speed of the computer and the Internet connection. If a prerequisite fails during this step, first retry the installer. If the installer continues to fail, see the [Troubleshooting](#) section of this topic.
3. The installer installs Modern POS.
4. On the page that states that installation was successful, select **Close** to exit the installer.

You can now start the program.

#### NOTE

This installation occurs only for the administrator user who ran the installer. For all other users, a desktop icon to install Modern POS is created. Every time that a user signs in, he or she must double-click this icon. The program will then be installed or updated, as required. If a user doesn't use the desktop icon after an update, the POS client will request that the user run from the desktop icon instead to update correctly prior to running.

### Run the installer on any other device (Windows Phone, Google Android device, or Apple iOS device)

1. If the application wasn't downloaded directly to the device, transfer the downloaded app file and the associated configuration file to the same folder on the device. Depending on the type of device, the app file will be an APPX, APK, or IPA file.

Note that this step can be done in various ways. For example, the files can be accessed through a shared folder, transferred via USB cable, or securely mailed to the user's device.

2. Use a file explorer on the device to browse to the app directory.
3. Tap the app to begin application installation. (If the configuration file was saved to the same location, the Commerce Scale Unit URL will be automatically entered when you start the application and begin device activation.)

Note that some devices require that you double-tap the file to begin application installation. Some devices might not notify you that an application has been installed. On those devices, we recommend that you look at the application list to verify that the application was correctly installed.

4. When the installation is completed, you should be able to start the application from the application list on the device. For example, after you install the application on a Windows Phone, you can start it from the home screen tiles list.

You can now start the program.

## Create a worker

For this topic, we have already created workers and assigned them to the Houston address book in the demo data that is provided. Therefore, this topic will use pre-generated data.

### Create a worker

1. Go to **Retail and Commerce > Employees > Workers**.
2. On the Action Pane, select **New** to create a new employee.
3. Enter the first and last name. For example, enter **John** as the first name and **Smith** as the last name.
4. Verify that the **Legal entity** field is set to **USRT**, the **Worker type** field is set to **Employee**, and the **Employment start date** field is set to the current date at 12 AM, so that the worker's employment starts immediately.
5. Select the **Assign a position** check box. Select position number **000544**, which is the Store manager position.
6. Set the **Personnel action type** field to **Hire Action** to hire a new employee immediately.
7. Select **Continue**.
8. On the Action Pane, select **Complete** to finish creating the new worker.
9. Return to the worker list. Search for the newly created worker (for example, John Smith). Select the worker's name to see the details of the new worker.
10. On the Action Pane, select **Edit**.
11. Verify that the language for the worker is **en-us**.

12. Under **Worker summary**, in the **Address books** field, select the **Houston** store.
13. On the **Commerce** tab, you can reset the POS password. For this tutorial, reset the password to **123**.
14. On the **Commerce** tab, under **Screen layout**, assign a screen layout. For example, select **F2MP16:9M (Fabrikam MPOS Manager (16:9))**.
15. On the Action Pane, select **Save**.
16. Go to **Retail and Commerce > Periodic > Distribution schedule**.
17. Select the **1060 – Staff** job, and then, on the Action Pane, select **Run now** to sync the worker data to the channel database.
18. After the new worker has been created and synced to stores, worker John Smith can sign in to any POS device that is used in the HOUSTON store that he is assigned to, and he can perform transactions on that device. However, the device must be activated first. The following section explains how to activate a device for a new worker.

### **Map an Azure AD account to a worker who has POS permissions for device activation**

You must complete this procedure before you activate Modern POS for a new worker.

1. In Commerce, from the **Worker** page, open the **Worker details** page for the worker that you created in the previous procedure.
2. On the Action Pane, select **Edit**.
3. On the **Commerce** tab, select the **POS permissions** link. Under **POS permission group**, verify that the value is **Manager**.
4. When you've finished, return to the **Worker details** page for the new worker.

To return to the **Worker details** page, select the **Close** button (X) on the right side of the Action Pane.

5. On the Action Pane, select **Commerce**, and then select **Associate existing identity**.
6. In the dialog box that appears, select the Azure AD account that is named **admin AX Admin**. (If an alternative administrator Azure AD account has been created, select that account instead.)
7. Select **OK**. In the demo data, the Azure AD account that is associated with the administrator account in Headquarters is your administrator Azure AD account.
8. On the Action Pane, select **Save**, and then refresh the page. The **External identity** section should be now updated with the new information.

Note that the **External identifier** field will remain empty. This behavior is expected. Therefore, you can ignore it.

This procedure should be completed before you activate Retail Cloud POS or Modern POS. For more information, see [Manage activation accounts and validate devices](#).

### **Run the Validate Devices for Activation check**

1. In Headquarters, open the **Device** page (**Retail and Commerce > Setup POS > Devices**).
2. Select the device to validate for device activation, and then select **Validate Devices for Activation**. For example, select device **HOUSTON-3**.
3. In the dialog box that appears, select the worker to validate the device for (that is, the worker that you mapped to the Azure AD account in the previous procedure). For example, select worker **000160**.
4. Select **OK**, and make sure that you receive the following message: "Pre-Activation validation completed for Device HOUSTON-3 and Staff 000160. Validation: Passed"

## Activate a device

## NOTE

It is possible for the Safari browser to show an error during device activation of a Cloud POS device due to an Azure Active Directory token being unattainable. You can resolve this issue by utilizing the [Microsoft Enterprise SSO plug-in for Apple devices](#).

1. Start Modern POS on your computer. Read the instructions on the **Before you start** page, and make sure that they are completed. Then select **Next**.
2. Select **Activate**. You're redirected to the Azure AD sign-in page.
3. Enter the Azure AD account that you mapped earlier, such as `admin@<MyCompany>.onmicrosoft.com`, and the password.
4. When activation is completed, select **Get Started**.
5. Sign in to Modern POS by using worker account **000160** and the password **123**.

The device should now be activated and ready to use.

## Update the Modern POS application

### NOTE

To learn more about deployable packages, see [Apply a deployable package](#).

1. After a Modern POS application is uploaded into the environment, the version of the package can be selected on the device. The package listings should include the new uploaded application.
2. To update the Modern POS application, follow the steps in the [Download and install Modern POS](#) section. To do an in-place update, just run the newer version of the self-service installer. Uninstallation isn't required or recommended. Device activation status will be maintained after the update.
3. The installer will use the currently installed configuration settings. If the configuration file has changed, because of various configuration changes in Commerce, an update won't change the Modern POS application settings.

## Troubleshooting

### Troubleshoot installation

- Your browser blocks the download pop-up that is generated.

**Solution:** Select either **Allow once** or **Options for this site > Always allow** (or the equivalent commands in the browser that you're using). Then, while the correct register is still selected, select **Download** again.

- The installation package that you must use depends on whether you require offline support. The correct package is automatically selected for download. For the offline package, SQL Server must be installed and must meet the requirements for the offline package.

**Solution:** No action is required. If SQL Server isn't already installed (or if it doesn't meet the requirements), it's downloaded and installed. The installer gives generic information about the download and installation of SQL Server Express 2014. This installation might require a long time.

- The installation occurs only for the administrator user who ran the installer, but not for any other users.

**Solution:** The installer generates a desktop icon that is used to install, upgrade, and run Modern POS. This icon is generated for every user on the computer. When a user who must install Modern POS double-clicks this icon, the program is installed. The user can then start to use Modern POS.

- SQL Server isn't successfully downloaded and installed through the self-service Modern POS installer.
  - **Solution 1:** A list of reasons shows the prerequisites that failed. If the list includes **SMO** or **SQL Management Objects**, first try to run the installer again. SQL Server Management Objects (SMO) are installed during SQL Server installation. Therefore, it's possible that the operating system didn't pick up the registration of the executable program that you used. When you run the installer a second time, the prerequisites are retested, and the prerequisite check should correctly verify the required executable program. If the installer continues to fail, restart the system to fully complete the registration of SQL Server, and then rerun the installer.
  - **Solution 2:** Manually download and install SQL Server (Microsoft SQL Server Express or another version) by using Advanced Tools. During installation, select **Full-text search** as an additional feature.
- The installation of Modern POS fails, because the registration of performance (perf) counters failed.

**Solution:** Follow these steps to fix this issue:

1. Open a **Command Prompt** window as an administrator.
2. Enter the following command.

```
lodctr /s:"perf_backup.txt"
```

3. Enter the following command.

```
lodctr /R
```

4. If the system doesn't rebuild the performance counter settings from the system backup, rerun the **lodctr /R** command.
  5. Rerun the Modern POS installer.
- If you're using a downloaded virtual hard disk (VHD) instead of a cloud-hosted environment, the downloader might fail.
    - **Solution 1:** In a downloaded VHD, the Azure Storage Emulator must be installed and must be running correctly. Otherwise, the self-service packages can't be downloaded correctly.
    - **Solution 2:** A failure might have occurred during the process of integrating the VHD into Microsoft Hyper-V. You must manually edit permissions before the packages can be downloaded correctly. Follow these steps:
      1. In File Explorer, browse to **C:\Microsoft Dynamics 365\70\Retail Server**.
      2. Right-click the **SelfServicePackages** folder, and then select **Properties**.
      3. On the **Security** tab, select **Edit**.
      4. In the **Permissions for SelfServiceDeployment** dialog box, select **Add**.
      5. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, select **Locations**.
      6. In the **Locations** dialog box, select the first entry in the list (the local computer), and then select **OK**.
      7. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, enter the name **IIS\_IUSRS**, and then select **Check names**. The object name should be changed to **IIS\_IUSRS**. Select **OK**.
      8. In the **Permissions for SelfServiceDeployment** dialog box, select the new **IIS\_ISURS** user. Under **Permissions for IIS\_IUSRS**, select **Allow** for the **Full control** permission. Select **OK**.
      9. In the **Open permission** dialog box, select **OK**.



- The latest iOS version does not support your self-signed certificate.

**Solution 1:** Utilize a domain and generate a proper domain-based certificate.

**Solution 2:** Download the open source OpenSSL library and perform the following after completing installation:

1. Using PowerShell, create a private key for the root Certificate Authority (CA) using a command such as `$ openssl genrsa -des3 -out rootCA.key 2048`. 2. You will be prompted for a password, which must be remembered for later usage. 3. Next, generate the root certificate using a command such as `$ openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem`. There will be a prompt for the password entered previously and some basic certificate information.

#### NOTE

The number of days the certificate is valid for can be altered. In the above example this is 1024 days.

- d. Create a new `info.ext` file and enter the following details:
  - o `keyUsage = keyEncipherment, dataEncipherment - extendedKeyUsage = 1.3.6.1.5.5.7.3.1 - subjectAltName = @alt_names - [alt_names] - DNS.1 = <FULLY QUALIFIED DOMAIN NAME OF HOST COMPUTER>`
- e. Generate the signing request and private key using a command such as `openssl req -new -nodes -out server.csr -newkey rsa:2048 -keyout server.key`.
- f. Issue the certificate using the previously generated root certificate using a command such as `$ openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out server.crt -days 500 -sha256 -extfile info.ext`. There will be another prompt for the root key password and you will need to specify the number of days that the certificate is valid (500 days in this example).
- g. Generate the IIS certificate using a command such as `$ openssl pkcs12 -inkey server.key -in server.crt -export -out server.pfx`. This command will request a new password, which will be used later when the certificate is imported.
- h. Open `Certmgr.msc` and go to **Trusted Root Certificate Authorities**. Use the **Import** action to import the previously generated `rootCA.pem` root CA file.
  - i. In the same window, go to **Personal** and use the **Import** action to import the previously generated `server.pfx`.
  - j. Next, open the **IIS Manager**, select the `RetailHardwareStationWebSite` and select **Edit Bindings** from the right-most menu.
  - k. In the new window, select the HTTPS site binding, and the select **Edit**. In the final screen, select the newly installed certificate and select **OK**.
  - l. Verify the certificate is correctly being used. In a web browser, go to "https://<hostname>/HardwareStation/ping".
- m. Install the certificate on the iOS device:
  - o Copy the `rootCA.pem` file and rename the copy to `rootCA.crt`. - Using OneDrive or another file hosting location, upload the `rootCA.crt` and `server.crt` so that they can be downloaded onto the iOS device.
- n. On the iOS device, go to **Settings > General > Profiles** and select the downloaded profile for the `rootCA.crt`. Select **Install**.
- o. Validate that the profile status updates to **Verified**. Repeat the same process for the `server.crt` file.
- p. Go to **Settings > General > About > Certificate Trust Settings** and enable the installed

root certificate.

- q. On the iOS device, use the hardware station ping URL specified previously to verify that the certificate is trusted.
- r. Open the POS application in **Non-drawer mode** and pair to the hardware station as typically performed.

### Troubleshoot device activation for Modern POS

- The Microsoft account (Azure AD) sign-in page doesn't open.

**Solution:** The Azure AD endpoint might be unreachable. Wait a few minutes, and then try again.

- After you enter the Azure AD account, you receive an error message that states that the user isn't authorized.

**Solution:** Verify that the Azure AD user is mapped to a worker who has POS permission to activate devices. The **Manage device** permission for the worker should be set to **Yes**.

- Device activation isn't completed. It fails during one of the steps.

**Solution:** Follow this checklist to verify that all data is correct:

- Complete the Validate Devices for Activation check in Headquarters, and make sure that the device passes validation.
- On the client computer where you're activating the device, access the Commerce Scale Unit URL health check, and make sure that the health check is passed. Use the following format for the URL:  
`https://MyCompanyNameet.axcloud.dynamics.com/commerce/healthcheck?testname=ping`
- The worker must be mapped to an Azure AD account (under **External identity**).
- The Azure AD account that is mapped must belong to the same tenant.
- To map the worker to the Azure AD account, sign in to Headquarters by using the Admin account for Microsoft Dynamics Lifecycle Services (LCS).
- Make sure that the worker is set up as a Commerce user in the Manager role. (This item is checked by validation.)
- Make sure that the channel is published. (This item is checked by validation.)
- Make sure that the channel database has the synced data from Headquarters, and that download jobs are running.
- Set up the hardware profile under **Registers**. (This item is checked by validation.)
- Make sure that the register and store have a screen layout. (This item is checked by validation.)
- Make sure that a primary address is set up for the legal entity.
- Make sure that the language is set up for the Commerce Data Exchange: Real-time Service user profile (JBB in the demo data).
- Make sure that the Real-time Service profile has the correct access.
- Make sure that the electronic funds transfer (EFT) configuration value is present.

### Troubleshoot Modern POS connectivity

On a single-computer system, such as a developer topology or a demo environment, or when Commerce Scale Unit and Modern POS are installed on the same computer, Modern POS can't complete device activation.

**Solution:** This issue occurs because Modern POS can't make network calls to the same computer (that is, calls to itself). To mitigate this issue, you must enable an AppContainer loopback exception so that communications can occur to the same computer. Various applications will help enabling this loopback for Modern POS. For more information about loopback, see [How to enable loopback and troubleshoot network isolation](#).

## Additional resources

## Install the POS layout designer

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage activation accounts and validate devices

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how an IT Pro can set up Commerce activation accounts for workers to activate Modern POS or Cloud POS devices.

## Setting up a device activation account for a single worker

This procedure should be completed before you activate Cloud POS.

1. In Commerce, from the **Workers** page, open the **Worker details** page for the worker to assign AAD device activation privileges to. Click **Edit**.
2. On the **Commerce** tab, click the **POS permissions** link. Make sure that the worker is in the Manager Permission group, or that **Manage Devices** is set to **Yes** for the worker.
3. On the **Commerce** tab, under **External identity**, update the values for the following fields:
  - Alias
  - UPN
  - External identifier
4. You can update the **External identity** fields by using an existing AAD account or creating a new AAD account. To update the fields, access the **External identity** options from the **Commerce** main menu (**Commerce > Associate existing identity** or **Commerce > Create new identity**).
5. To use an existing AAD account, select **Commerce > Associate existing identity**. In the slider, click the AAD account that has the correct name, and then click **OK**. The AAD account that is associated with that name and alias is the user's Activation account for Modern POS.
6. Complete and save the changes on the **Workers** page, and then refresh the page. The section that contains external identity information should be updated with the new information. The mapped AAD account is now your Activation account for Cloud POS and Modern POS. This account is mapped to a worker for the required POS permissions. You can use this AAD account for Modern POS or Cloud POS activation.
7. The **Create external identity** feature creates a new AAD account for you by using the alias that you enter. To update the fields, access the **External identity** options from the **Commerce** main menu (**Commerce > Create new identity**).
8. You can either manually enter the alias to generate or use the **Reset to default** button. Then manually enter a strong password, and click **OK**.
9. If the worker is created successfully, you receive a message on the **Workers** page. The mapped AAD account is now the user's Activation account for Cloud POS and Modern POS. This account is mapped to a worker for the required POS permissions. You can use this AAD account for Modern POS or Cloud POS activation.

## Setting up device activation accounts for multiple workers

You can set up activation accounts for multiple workers in bulk. However, this functionality is supported only if you're creating new external identities, not if you're associating identities.

1. In the workers form, select the list of workers to set the activation account for.

2. Click **Commerce > Create external identity** to update the fields. Any AAD accounts that are associated with the workers appear in this pane.

#### **NOTE**

These accounts aren't device activation accounts until you map them by using the external identity flow options.

3. If you want to use the existing AAD accounts as activation accounts, you can't map them in bulk. Cancel the selection of those workers, and then map them individually by using **Use existing external identity**.
4. To create new AAD accounts and associate them with the workers, so that they can be used as activation accounts, update the **Alias** and **Password** fields, and then click **OK**. In the main worker form, you receive a message as activation accounts are created for each worker.

## Run the Validate Devices for Activation check at headquarters

Before handing an activation account to a worker, an IT Pro must run the Validate devices check for the devices assigned to the worker. This will help identify any potential failures of device activation in advance and fix it before it is given to the worker.

1. Open the **Device** page in HQ (**Retail and commerce > Channel setup > POS setup > Devices**).
2. Select the device to validate for device activation, and then click **Validate devices for activation**. For example, select device **HOUSTON-2**.
3. In the dialog box that appears, select the worker to validate the device for (that is, the worker that you mapped to the AAD account in the previous procedure). For example, select worker **000160**.
4. Click **OK**, and make sure that you receive a message similar to the following: "Pre-Activation validation completed for Device HOUSTON-2 and Staff 000160. Validation: Passed"

## Checklist to follow before activation

1. Complete the **Validate devices for activation** check in HQ, and make sure that the device passes validation.
2. On the client machine where you're activating the device, access the Commerce Scale Unit URL health check, and make sure that the health check is passed. Use the following format:  
`https://clxtestax404ret.cloud.test.dynamics.com/en/healthcheck?testname=ping`
3. The worker must be mapped to an AAD account (under **External identity**).
4. The AAD account to map must belong to the same tenant.
5. To map the worker to the AAD account, sign in to HQ by using the Admin account for Microsoft Dynamics Lifecycle Services (LCS).
6. Make sure that the worker is set up as a Commerce user in the Manager role (checked by validation).
7. Make sure that the channel data is present in the channel database.
8. Set up the hardware profile under **Registers > Register** (checked by validation).
9. Make sure that the register and store have a screen layout (checked by validation).
10. Make sure that a primary address is set up for the legal entity.
11. Make sure that the electronic funds transfer (EFT) configuration value is present.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Point of sale (POS) device activation

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article explains the new guided device activation for Cloud POS and Modern POS, and explains the client simplifications that help users easily activate devices without having to manually enter register and device ID information.

## Checklist to follow before activation

1. Complete the **Validate devices for activation** check in Headquarters (HQ), and make sure that the device passes validation.
2. On the client machine where you're activating the device, access the Commerce Scale Unit URL health check, and make sure that the health check is passed. Use the following format:  
`https://c1xtestax404ret.cloud.test.dynamics.com/en/healthcheck?testname=ping`.
3. The worker must be mapped to a Microsoft Azure Active Directory (AAD) account (under **External identity**).
4. The AAD account to map must belong to the same tenant.
5. To map the worker to the AAD account, sign in to HQ by using the Admin account for Microsoft Dynamics Lifecycle Services (LCS).
6. Make sure that the worker is set up as a user in the Manager role (checked by validation).
7. Make sure that the channel is published (checked by validation).
8. Make sure that the channel database has the synced data from HQ, and that download jobs are running. To check this, run the following command in the channel database for the store.

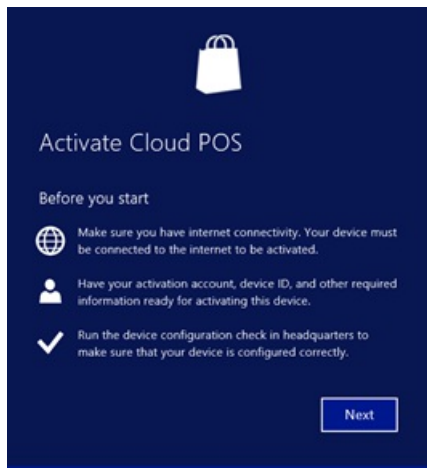
```
select * from crt.STORAGELOOKUPVIEW
```

Make sure that data is returned, and that the result isn't empty.

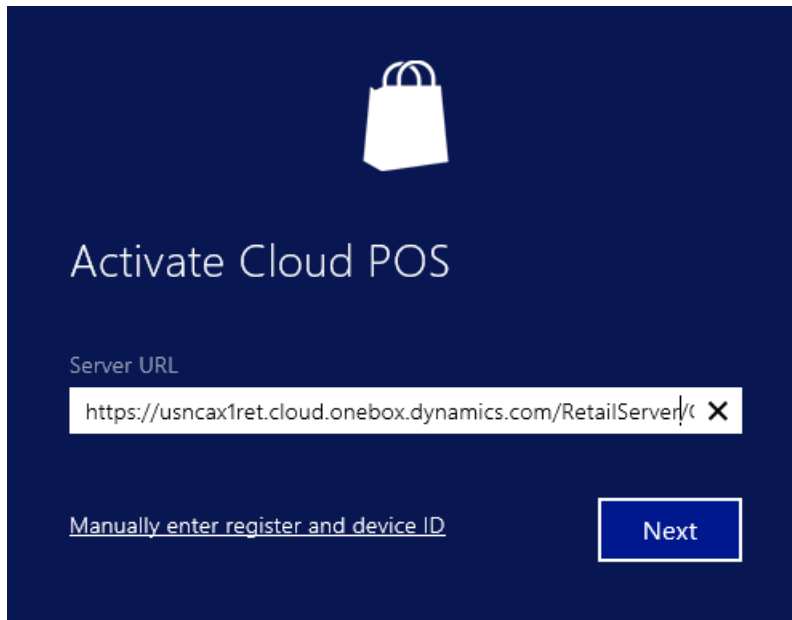
9. Set up the hardware profile under **Register** (checked by validation).
10. Make sure that the register and store have a screen layout (checked by validation).
11. Make sure that a primary address is set up for the legal entity.
12. Make sure that the language is set up for the Commerce Data Exchange: Real-time Service user profile (JBB in the demo data).
13. Make sure that the Real-time Service profile has the correct access.
14. Make sure that the electronic funds transfer (EFT) configuration value is present.

## Activate a Modern POS or Cloud POS device by using guided activation

1. Open the initial device activation page for Modern POS or Cloud POS. You're prompted to sign in.
2. On the **Before you start** page, follow the instructions, and then click **Next**.

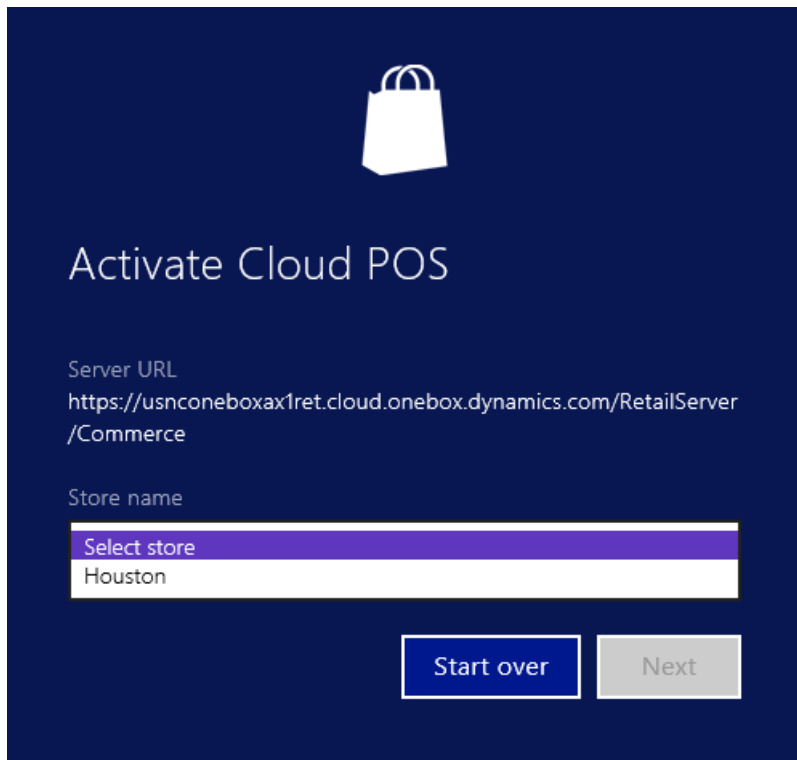


3. Start Cloud POS or Modern POS.
4. Use your AAD credentials to sign in. The AAD account must already be mapped. For instructions, see [Configure, install, and activate Modern POS \(MPOS\)](#). For Cloud POS, the server URL is automatically entered in the address bar. For Modern POS, you must copy and paste the server URL.



5. Click **Next** to populate the list of stores.
6. Select the correct store in the list.

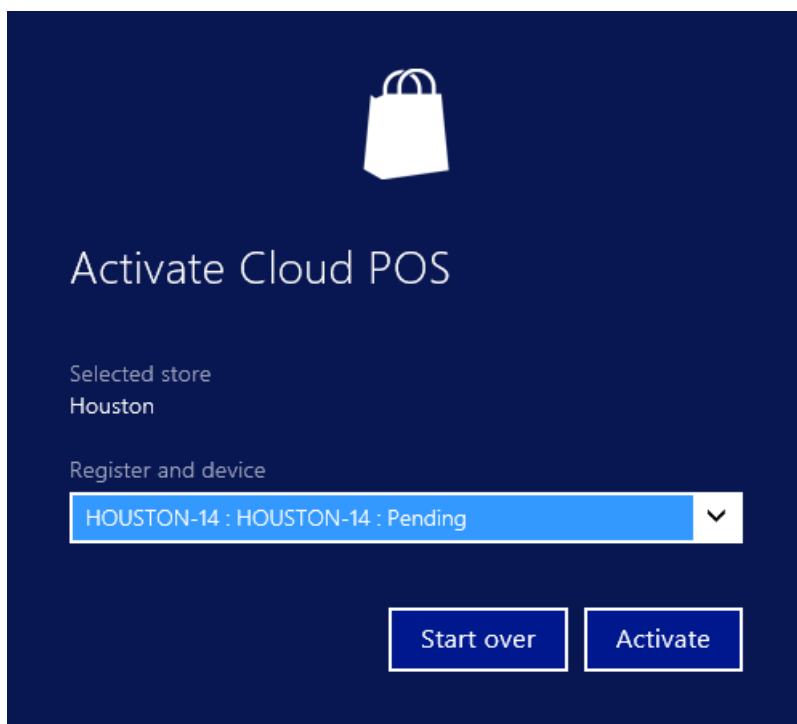




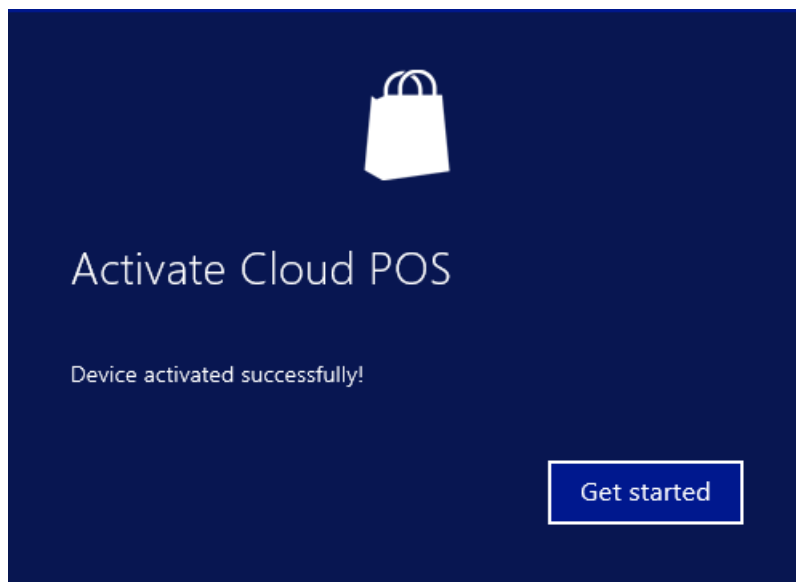
7. Select the correct register and device.

**NOTE**

The device can be **Pending**, **De-activated**, or **Activated**. Alternatively, if you turned on the HQ **Allow devices to be associated to registers from store setting**, you might see a list of registers that have no device associated with them.



8. Click **Activate**. The device should be activated.



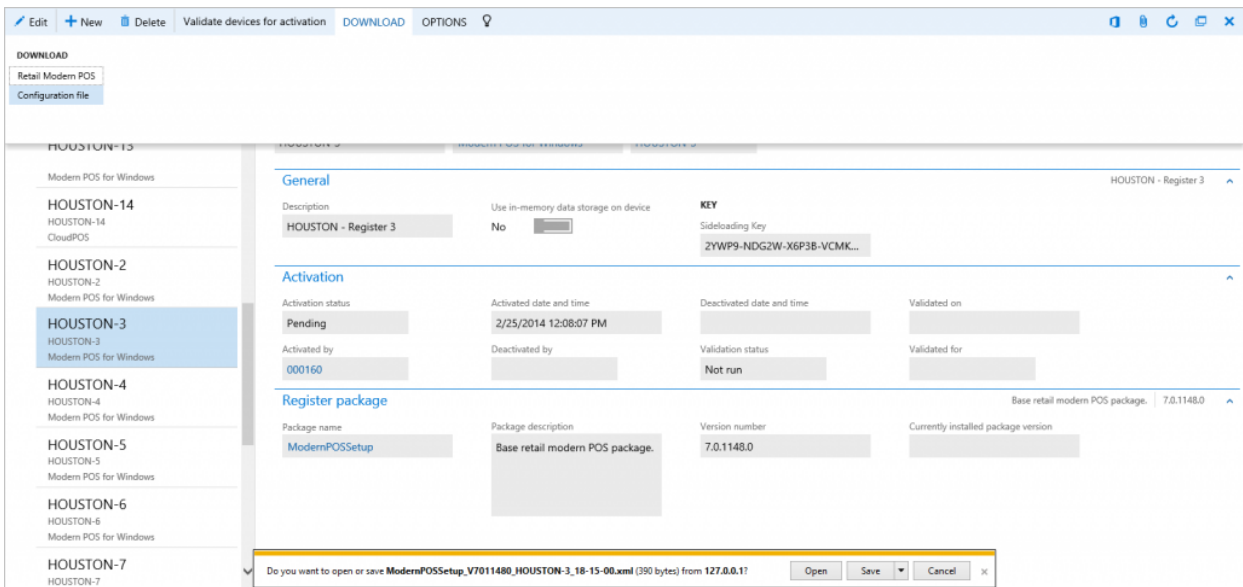
## Create a device ID from Modern POS and Cloud POS

We have added features to create a device (that is, automatically generate a device ID) from Modern POS or Cloud POS, so that the device can be associated with a register that doesn't yet have devices mapped to it. This functionality can be used in Modern POS only if you set the HQ settings as follows.

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters > General**.
2. Under **Devices**, set **Allow register association from device** to **Yes**.
3. In the Modern POS client, you can now add a device when you select a register that is listed as **No associated devices** in the guided activation flow.
4. After you select the register, you can either select a device that doesn't have register mapping or use the **Or, Add a Device** link.
5. Click the **Or, Add a Device** link, and then either enter the new device ID or select **Automatically create a new device ID for me**.
6. Click **Activate** to create a new device ID, associate it with the selected register, and complete the activation.

## Activate the device for Modern POS by using a configuration file

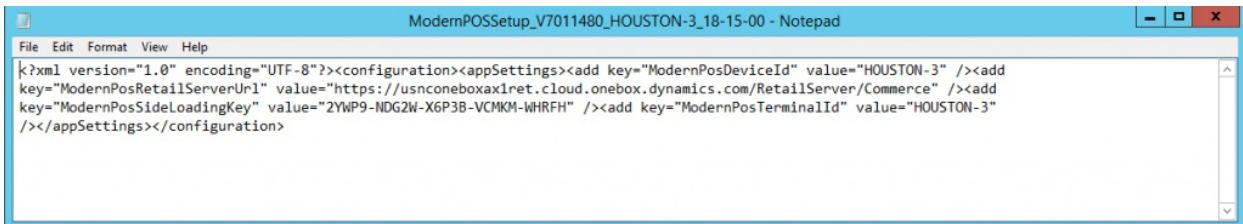
IT Pros can now easily configure device activation for Modern POS by using a configuration file that can be downloaded together with Modern POS. This file is now available on the **Devices** page for the appropriate Modern POS device (**Retail and Commerce > Channel setup > POS setup > Devices**).



The configuration file is used to enter the Commerce Scale Unit URL, device ID, and register number for device activation. During installation, the installer selects this file and populates the values for device activation.

**IMPORTANT**

The user must put the file in the same folder as the Modern POS self-service package and run the .exe file.



Modern POS starts in Manual entry mode, and the Commerce Scale Unit URL, device ID, and register ID are pre-populated for activation.

## Activate the device for Cloud POS by using syntactic sugar

IT Pros can now configure device activation for Cloud POS by providing the device ID and register ID as the part of the Cloud POS URL. The link is available in the **Cloud POS URL** field on the **Devices** page. (**Retail and Commerce > Channel setup > POS setup > Devices**).

Cloud POS starts in Manual entry mode, and the Commerce Scale Unit URL, device ID, and register ID are pre-populated for activation.

Filter

- CloudPOS
- HOUSTON-1  
HOUSTON-1  
Modern POS for Windows
- HOUSTON-10  
HOUSTON-10  
Modern POS for Windows Phone
- HOUSTON-11  
HOUSTON-11  
CloudPOS
- HOUSTON-12  
Modern POS for Windows
- HOUSTON-13  
Modern POS for Windows
- HOUSTON-14  
HOUSTON-14  
CloudPOS
- HOUSTON-2  
HOUSTON-2  
Modern POS for Windows
- HOUSTON-3

## Devices

Device ID  
HOUSTON-14
Application type  
CloudPOS
Register number  
HOUSTON-14

---

### General

Description  
[Redacted]

Use in-memory data storage on device  
No

Cloud POS URL  
<https://usnconeboxax1pos.cloud.g>

---

### Activation

Activation status <b>Pending</b>	Activated date and time [Redacted]	Deactivated date and time [Redacted]	Validated on 7/10/2015 03:30:00 PM
Activated by [Redacted]	Deactivated by [Redacted]	Validation status <b>Passed</b>	Validated for [Redacted]

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Security best practices for Cloud POS in shared environments

2/18/2021 • 17 minutes to read • [Edit Online](#)

Retail Cloud POS is a web application that runs in the context of a browser. This topic provides recommendations that can help secure Retail Cloud POS in a shared environment.

## Background

Retail Cloud POS is a web application that runs in the context of a web browser. Therefore, it's vulnerable to attack when a user can run any script in the context of the web application. One requirement for such attacks is that the user must have physical access to the computer, either in person or by using Remote Desktop Connection. Vulnerability to attack is an existing issue in most browsers that provide developer tools, and that enable scripts to be run without sufficient privilege control. Because the web application will have little influence over its hosting environment, one way to mitigate security issues is to add defense-in-depth. The defense-in-depth can be built by taking advantage of the restrictive policies of both the browser and the operating system.

## Hardening instructions for a Retail Cloud POS computer

Here are some of the defense-in-depth recommendations for the operating system and/or browser that will have an activated instance of Retail Cloud POS. The settings should be enabled or set by a high-privileged account for the operating system. Retail Cloud POS should be used by a low-privileged account that can't override those settings. We recommend that you enable all the following settings. Otherwise, you could create a security loophole that will be prone to security exploitation.

- **Required** - Disable script execution in the browser's address bar.
- **Required** - Disable the browser's developer console.
- **Required** - Retail Cloud POS should be accessed by a low-privileged user.
- **Required** - Set up group policies to enable a kiosk session.
- **Recommended** - Set up a proxy to access only websites included in a safe list.

## Disable script execution in the address bar of the browser that runs Retail Cloud POS

### Internet Explorer

There is no option to disable script execution in the address bar in Internet Explorer. One alternative is to hide the address bar itself.

1. Create a shortcut for the Retail Cloud POS URL, and copy it to each store worker's Microsoft Windows desktop.
2. Run **regedit.exe** to change the registry to disable the Internet Explorer address bar.  
[HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\Internet Explorer\ToolBars\Restrictions]  
"NoNavBar"=dword:00000001

### Microsoft Edge

By design, Microsoft Edge prevents script execution in the address bar. Therefore, no action is required.

## Disable the developer console in the browser that runs Retail Cloud

# POS

## Internet Explorer

Use Group Policy Editor to enable the following group policy to disable the Internet Explorer developer console:

\Administrative Templates\Windows Components\Internet Explorer\Toolbars\Turn off Developer

Tools="Enabled"

## Microsoft Edge

Run **regedit.exe** to change the registry to disable the developer console.

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\MicrosoftEdge\F12]

"AllowDeveloperTools"=dword:00000000

## Retail Cloud POS should be accessed by a low-privileged user

A point of sale (POS) user must be a non-administrative account that doesn't have privileges to change applied policies.

## Set up group policies to enable a kiosk session

We recommend that you apply the following restrictions for Retail Cloud POS users:

- Restrict access to the file system.
- Restrict access to Control Panel.
- Restrict access to removable drives.
- Restrict access to shells that run commands.
- Restrict access to the registry.
- Restrict access to application management.

The following table lists the group policies to enable kiosk mode. The set of policies requires that you start your browser at the sign-in script. These policies can be adjusted to your requirements. You should always assess any security implications or talk to a specialist.

SETTING	STATE	COMMENT	PATH
Enable screen saver	Disabled	No	\Control Panel\Personalization
Allow DFS roots to be published	Disabled	No	\Shared Folders
Allow shared folders to be published	Disabled	No	\Shared Folders
Add Search Internet link to Start Menu	Disabled	No	\Start Menu and Taskbar
Show Quick Launch on Taskbar	Disabled	No	\Start Menu and Taskbar
Show the Apps view automatically when the user goes to Start	Disabled	No	\Start Menu and Taskbar

SETTING	STATE	COMMENT	PATH
Show "Run as different user" command on Start	Disabled	No	\Start Menu and Taskbar
Add the Run command to the Start Menu	Disabled	No	\Start Menu and Taskbar
Show Start on the display the user is using when they press the Windows logo key	Disabled	No	\Start Menu and Taskbar
Show Windows Store apps on the taskbar	Disabled	No	\Start Menu and Taskbar
Turn off shell protocol protected mode	Disabled	No	\Windows Components\File Explorer
Turn on menu bar by default	Disabled	No	\Windows Components\Internet Explorer
Turn on Script Execution	Disabled	No	\Windows Components\Windows PowerShell
Hide the "Add a program from CD-ROM or floppy disk" option	Enabled	No	\Control Panel\Add or Remove Programs
Hide the "Add programs from Microsoft" option	Enabled	No	\Control Panel\Add or Remove Programs
Hide the "Add programs from your network" option	Enabled	No	\Control Panel\Add or Remove Programs
Hide Add New Programs page	Enabled	No	\Control Panel\Add or Remove Programs
Remove Add or Remove Programs	Enabled	No	\Control Panel\Add or Remove Programs
Hide the Set Program Access and Defaults page	Enabled	No	\Control Panel\Add or Remove Programs
Hide Change or Remove Programs page	Enabled	No	\Control Panel\Add or Remove Programs
Go directly to Components Wizard	Enabled	No	\Control Panel\Add or Remove Programs
Remove Support Information	Enabled	No	\Control Panel\Add or Remove Programs
Hide Add/Remove Windows Components page	Enabled	No	\Control Panel\Add or Remove Programs

SETTING	STATE	COMMENT	PATH
Disable the Display Control Panel	Enabled	No	\Control Panel\Display
Hide Settings tab	Enabled	No	\Control Panel\Display
Prevent changing color scheme	Enabled	No	\Control Panel\Personalization
Prevent changing theme	Enabled	No	\Control Panel\Personalization
Prevent changing visual style for windows and buttons	Enabled	No	\Control Panel\Personalization
Prohibit selection of visual style font size	Enabled	No	\Control Panel\Personalization
Prevent changing color and appearance	Enabled	No	\Control Panel\Personalization
Prevent changing desktop background	Enabled	No	\Control Panel\Personalization
Prevent changing desktop icons	Enabled	No	\Control Panel\Personalization
Prevent changing mouse pointers	Enabled	No	\Control Panel\Personalization
Prevent changing screen saver	Enabled	No	\Control Panel\Personalization
Prevent changing sounds	Enabled	No	\Control Panel\Personalization
Prevent addition of printers	Enabled	No	\Control Panel\Printers
Prevent deletion of printers	Enabled	No	\Control Panel\Printers
Hide "Set Program Access and Computer Defaults" page	Enabled	No	\Control Panel\Programs
Hide "Get Programs" page	Enabled	No	\Control Panel\Programs
Hide "Installed Updates" page	Enabled	No	\Control Panel\Programs
Hide "Programs and Features" page	Enabled	No	\Control Panel\Programs



SETTING	STATE	COMMENT	PATH
Hide the Programs Control Panel	Enabled	No	\Control Panel\Programs
Hide "Windows Features"	Enabled	No	\Control Panel\Programs
Hide "Windows Marketplace"	Enabled	No	\Control Panel\Programs
Turn off automatic learning	Enabled	No	\Control Panel\Regional and Language Options\Handwriting personalization
Hide Regional and Language Options administrative options	Enabled	No	\Control Panel\Regional and Language Options
Hide and disable all items on the desktop	Enabled	No	\Desktop
Remove the Desktop Cleanup Wizard	Enabled	No	\Desktop
Hide Internet Explorer icon on desktop	Enabled	No	\Desktop
Remove Computer icon on the desktop	Enabled	No	\Desktop
Remove My Documents icon on the desktop	Enabled	No	\Desktop
Hide Network Locations icon on desktop	Enabled	No	\Desktop
Remove Properties from the Computer icon context menu	Enabled	No	\Desktop
Remove Properties from the Documents icon context menu	Enabled	No	\Desktop
Do not add shares of recently opened documents to Network Locations	Enabled	No	\Desktop
Remove Recycle Bin icon from desktop	Enabled	No	\Desktop
Remove Properties from the Recycle Bin context menu	Enabled	No	\Desktop
Do not save settings at exit	Enabled	No	\Desktop

SETTING	STATE	COMMENT	PATH
Turn off Aero Shake window minimizing mouse gesture	Enabled	No	\Desktop
Prevent adding, dragging dropping and closing the Taskbar's toolbars			Enabled
Prohibit adjusting desktop toolbars	Enabled	No	\Desktop
Force Start to be either full screen size or menu size	Enabled	No	\Start Menu and Taskbar
Go to the desktop instead of Start when signing in	Enabled	No	\Start Menu and Taskbar
Turn off personalized menus	Enabled	No	\Start Menu and Taskbar
Lock the Taskbar	Enabled	No	\Start Menu and Taskbar
Turn off notification area cleanup	Enabled	No	\Start Menu and Taskbar
Remove Balloon Tips on Start Menu items	Enabled	No	\Start Menu and Taskbar
Prevent users from customizing their Start Screen	Enabled	No	\Start Menu and Taskbar
Remove common program groups from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Favorites menu from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Search link from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove frequent programs list from the Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Games link from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Help menu from Start Menu	Enabled	No	\Start Menu and Taskbar
Turn off user tracking	Enabled	No	\Start Menu and Taskbar
Remove All Programs list from the Start menu	Enabled	No	\Start Menu and Taskbar

SETTING	STATE	COMMENT	PATH
Remove Network Connections from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove pinned programs list from the Start Menu	Enabled	No	\Start Menu and Taskbar
Do not keep history of recently opened documents	Enabled	No	\Start Menu and Taskbar
Remove Recent Items menu from Start Menu	Enabled	No	\Start Menu and Taskbar
Do not use the search-based method when resolving shell shortcuts	Enabled	No	\Start Menu and Taskbar
Do not use the tracking-based method when resolving shell shortcuts	Enabled	No	\Start Menu and Taskbar
Remove Run menu from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Default Programs link from the Start menu.	Enabled	No	\Start Menu and Taskbar
Remove Documents icon from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Music icon from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Network icon from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Pictures icon from Start Menu	Enabled	No	\Start Menu and Taskbar
Do not search communications	Enabled	No	\Start Menu and Taskbar
Remove Search Computer link	Enabled	No	\Start Menu and Taskbar
Remove See More Results / Search Everywhere link	Enabled	No	\Start Menu and Taskbar
Do not search for files	Enabled	No	\Start Menu and Taskbar
Do not search Internet	Enabled	No	\Start Menu and Taskbar
Do not search programs and Control Panel items	Enabled	No	\Start Menu and Taskbar

SETTING	STATE	COMMENT	PATH
Remove programs on Settings menu	Enabled	No	\Start Menu and Taskbar
Prevent changes to Taskbar and Start Menu Settings	Enabled	No	\Start Menu and Taskbar
Remove Downloads link from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Homegroup link from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Recorded TV link from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove user's folders from the Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Videos link from Start Menu	Enabled	No	\Start Menu and Taskbar
Force classic Start Menu	Enabled	No	\Start Menu and Taskbar
Remove Clock from the system notification area	Enabled	No	\Start Menu and Taskbar
Prevent grouping of taskbar items	Enabled	No	\Start Menu and Taskbar
Do not display any custom toolbars in the taskbar	Enabled	No	\Start Menu and Taskbar
Remove access to the context menus for the taskbar	Enabled	No	\Start Menu and Taskbar
Hide the notification area	Enabled	No	\Start Menu and Taskbar
Prevent users from uninstalling applications from Start	Enabled	No	\Start Menu and Taskbar
Remove user folder link from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove user name from Start Menu	Enabled	No	\Start Menu and Taskbar
Remove links and access to Windows Update	Enabled	No	\Start Menu and Taskbar
Remove the "Undock PC" button from the Start Menu	Enabled	No	\Start Menu and Taskbar

SETTING	STATE	COMMENT	PATH
Remove Notifications and Action Center	Enabled	No	\Start Menu and Taskbar
Disable showing balloon notifications as toasts.	Enabled	No	\Start Menu and Taskbar
Remove the Security and Maintenance icon	Enabled	No	\Start Menu and Taskbar
Remove the networking icon	Enabled	No	\Start Menu and Taskbar
Remove the battery meter	Enabled	No	\Start Menu and Taskbar
Remove the volume control icon	Enabled	No	\Start Menu and Taskbar
Turn off feature advertisement balloon notifications	Enabled	No	\Start Menu and Taskbar
Do not allow pinning Store app to the Taskbar	Enabled	No	\Start Menu and Taskbar
Do not allow pinning items in Jump Lists	Enabled	No	\Start Menu and Taskbar
Do not allow pinning programs to the Taskbar	Enabled	No	\Start Menu and Taskbar
Do not display or track items in Jump Lists from remote locations	Enabled	No	\Start Menu and Taskbar
Turn off automatic promotion of notification icons to the taskbar	Enabled	No	\Start Menu and Taskbar
Lock all taskbar settings	Enabled	No	\Start Menu and Taskbar
Prevent users from adding or removing toolbars	Enabled	No	\Start Menu and Taskbar
Prevent users from rearranging toolbars	Enabled	No	\Start Menu and Taskbar
Do not allow taskbars on more than one display	Enabled	No	\Start Menu and Taskbar
Turn off all balloon notifications	Enabled	No	\Start Menu and Taskbar
Remove pinned programs from the Taskbar	Enabled	No	\Start Menu and Taskbar

SETTING	STATE	COMMENT	PATH
Prevent users from moving taskbar to another screen dock location	Enabled	No	\Start Menu and Taskbar
Prevent users from resizing the taskbar	Enabled	No	\Start Menu and Taskbar
Turn off taskbar thumbnails	Enabled	No	\Start Menu and Taskbar
Remove Task Manager	Enabled	No	\System\Ctrl+Alt+Del Options
Code signing for device drivers	Enabled	No	\System\Driver Installation
Turn off Windows Update device driver search prompt	Enabled	No	\System\Driver Installation
Disallow selection of Custom Locales	Enabled	No	\System\Locale Services
Disallow changing of geographic location	Enabled	No	\System\Locale Services
Disallow user override of locale settings	Enabled	No	\System\Locale Services
CD and DVD: Deny read access	Enabled	No	\System\Removable Storage Access
CD and DVD: Deny write access	Enabled	No	\System\Removable Storage Access
Floppy Drives: Deny read access	Enabled	No	\System\Removable Storage Access
Floppy Drives: Deny write access	Enabled	No	\System\Removable Storage Access
Removable Disks: Deny read access	Enabled	No	\System\Removable Storage Access
Removable Disks: Deny write access	Enabled	No	\System\Removable Storage Access
All Removable Storage classes: Deny all access	Enabled	No	\System\Removable Storage Access
Tape Drives: Deny read access	Enabled	No	\System\Removable Storage Access
Tape Drives: Deny write access	Enabled	No	\System\Removable Storage Access

SETTING	STATE	COMMENT	PATH
WPD Devices: Deny read access	Enabled	No	\System\Removable Storage Access
WPD Devices: Deny write access	Enabled	No	\System\Removable Storage Access
Prevent access to the command prompt	Enabled	No	\System
Prevent access to registry editing tools	Enabled	No	\System
Prevent the wizard from running.	Enabled	No	\Windows Components\Add features to Windows 10
Turn off Program Compatibility Assistant	Enabled	No	\Windows Components\Application Compatibility
Search, Share, Start, Devices and Settings don't appear when the mouse is pointing to the upper-right corner of the screen	Enabled	No	\Windows Components\Edge UI
Disable help tips	Enabled	No	\Windows Components\Edge UI
Turn off tracking of app usage	Enabled	No	\Windows Components\Edge UI
Do not show recent apps when the mouse is pointing to the upper-left corner of the screen	Enabled	No	\Windows Components\Edge UI
Prevent users from replacing the Command Prompt with Windows PowerShell in the menu they see when they right-click the lower-left corner or press the Windows logo key+X	Enabled	No	\Windows Components\Edge UI
Turn off switching between recent apps	Enabled	No	\Windows Components\Edge UI
Turn on or off details pane	Enabled	No	\Windows Components\File Explorer\Explorer Frame Pane

SETTING	STATE	COMMENT	PATH
Turn off Preview Pane	Enabled	No	\Windows Components\File Explorer\Explorer Frame Pane
Do not display the Welcome Center at user logon	Enabled	No	\Windows Components\File Explorer
Turn on Classic Shell	Enabled	No	\Windows Components\File Explorer
Remove CD Burning features	Enabled	No	\Windows Components\File Explorer
Remove DFS tab	Enabled	No	\Windows Components\File Explorer
Hide these specified drives in My Computer	Enabled	No	\Windows Components\File Explorer
No Entire Network in Network Locations	Enabled	No	\Windows Components\File Explorer
Remove File menu from File Explorer	Enabled	No	\Windows Components\File Explorer
Do not allow Folder Options to be opened from the Options button on the View tab of the ribbon	Enabled	No	\Windows Components\File Explorer
Remove Hardware tab	Enabled	No	\Windows Components\File Explorer
Hide the Manage item on the File Explorer context menu	Enabled	No	\Windows Components\File Explorer
Remove Shared Documents from My Computer	Enabled	No	\Windows Components\File Explorer
Remove "Map Network Drive" and "Disconnect Network Drive"	Enabled	No	\Windows Components\File Explorer
Remove the Search the Internet "Search again" link	Enabled	No	\Windows Components\File Explorer
Remove Security tab	Enabled	No	\Windows Components\File Explorer
Remove Search button from File Explorer	Enabled	No	\Windows Components\File Explorer



SETTING	STATE	COMMENT	PATH
Remove File Explorer's default context menu	Enabled	No	\Windows Components\File Explorer
Prevent access to drives from My Computer	Enabled	No	\Windows Components\File Explorer
Turn off Windows+X hotkeys	Enabled	No	\Windows Components\File Explorer
No Computers Near Me in Network Locations	Enabled	No	\Windows Components\File Explorer
Request credentials for network installations	Enabled	No	\Windows Components\File Explorer
Prevent users from adding files to the root of their Users Files folder.	Enabled	No	\Windows Components\File Explorer
Turn off Accelerators	Enabled	No	\Windows Components\Internet Explorer\Accelerators
File menu: Disable closing the browser and Explorer windows	Enabled	No	\Windows Components\Internet Explorer\Browser menus
File menu: Disable Save As... menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
File menu: Disable Save As Web Page Complete	Enabled	No	\Windows Components\Internet Explorer\Browser menus
File menu: Disable New menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
File menu: Disable Open menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Help menu: Remove 'Send Feedback' menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Help menu: Remove 'For Netscape Users' menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Help menu: Remove 'Tip of the Day' menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus

SETTING	STATE	COMMENT	PATH
Help menu: Remove 'Tour' menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Turn off Shortcut Menu	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Hide Favorites menu	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Disable Open in New Window menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Turn off Print Menu	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Turn off the ability to launch report site problems using a menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Disable Save this program to disk option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Tools menu: Disable Internet Options... menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
View menu: Disable Full Screen menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
View menu: Disable Source menu option	Enabled	No	\Windows Components\Internet Explorer\Browser menus
Turn off Developer Tools	Enabled	No	\Windows Components\Internet Explorer\Toolbars
Turn off toolbar upgrade tool	Enabled	No	\Windows Components\Internet Explorer\Toolbars
Hide the Command bar	Enabled	No	\Windows Components\Internet Explorer\Toolbars
Hide the status bar	Enabled	No	\Windows Components\Internet Explorer\Toolbars

SETTING	STATE	COMMENT	PATH
Disable customizing browser toolbars	Enabled	No	\Windows Components\Internet Explorer\Toolbars
Disable customizing browser toolbar buttons	Enabled	No	\Windows Components\Internet Explorer\Toolbars
Turn off add-on performance notifications	Enabled	No	\Windows Components\Internet Explorer
Do not allow users to enable or disable add-ons	Enabled	No	\Windows Components\Internet Explorer
Disable changing Advanced page settings	Enabled	No	\Windows Components\Internet Explorer
Turn off Favorites bar	Enabled	No	\Windows Components\Internet Explorer
Prevent per-user installation of ActiveX controls	Enabled	No	\Windows Components\Internet Explorer
Turn off Reopen Last Browsing Session	Enabled	No	\Windows Components\Internet Explorer
Turn off Tab Grouping	Enabled	No	\Windows Components\Internet Explorer
Prevent managing the phishing filter	Enabled	No	\Windows Components\Internet Explorer
Turn off Managing SmartScreen Filter for Internet Explorer 8	Enabled	No	\Windows Components\Internet Explorer
Prevent managing SmartScreen Filter	Enabled	No	\Windows Components\Internet Explorer
Turn off the Security Settings Check feature	Enabled	No	\Windows Components\Internet Explorer
Enforce full-screen mode	Enabled	No	\Windows Components\Internet Explorer

SETTING	STATE	COMMENT	PATH
Disable Import/Export Settings wizard	Enabled	No	\Windows Components\Internet Explorer
Prevent Internet Explorer Search box from appearing	Enabled	No	\Windows Components\Internet Explorer
Turn off Quick Tabs functionality	Enabled	No	\Windows Components\Internet Explorer
Turn off tabbed browsing	Enabled	No	\Windows Components\Internet Explorer
Disable changing Automatic Configuration settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing Temporary Internet files settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing Calendar and Contact settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing certificate settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing default browser check	Enabled	No	\Windows Components\Internet Explorer
Disable changing color settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing connection settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing font settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing language settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing link color settings	Enabled	No	\Windows Components\Internet Explorer

SETTING	STATE	COMMENT	PATH
Disable changing Messaging settings	Enabled	No	\Windows Components\Internet Explorer
Prevent managing pop-up exception list	Enabled	No	\Windows Components\Internet Explorer
Turn off pop-up management	Enabled	No	\Windows Components\Internet Explorer
Disable changing Profile Assistant settings	Enabled	No	\Windows Components\Internet Explorer
Prevent changing proxy settings	Enabled	No	\Windows Components\Internet Explorer
Disable changing ratings settings	Enabled	No	\Windows Components\Internet Explorer
Turn off the auto-complete feature for web addresses	Enabled	No	\Windows Components\Internet Explorer
Turn off suggestions for all user-installed providers	Enabled	No	\Windows Components\Internet Explorer
Turn off the quick pick menu	Enabled	No	\Windows Components\Internet Explorer
Search: Disable Find Files via F3 within the browser	Enabled	No	\Windows Components\Internet Explorer
Search: Disable Search Customization	Enabled	No	\Windows Components\Internet Explorer
Turn off ability to pin sites in Internet Explorer on the desktop	Enabled	No	\Windows Components\Internet Explorer
Turn off the offer to update to the latest version of Windows	Enabled	No	\Windows Components\Store
Turn off the Store application	Enabled	No	\Windows Components\Store

SETTING	STATE	COMMENT	PATH
Prohibit New Task Creation	Enabled	No	\Windows Components\Task Scheduler

## Set up a proxy to access only websites included in a safe list

You can define a list of websites that a store worker (cashier) requires for normal operations, and set up an administrator-controlled proxy that has access only to these websites. Retail Cloud POS requires access to the following websites:

- Retail Cloud POS website
- Microsoft Azure Active Directory sign-in page
- Commerce Scale Unit website
- Bing Maps resources
- Media resources
- Credit Card Payment acceptance page (optional)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and install Retail hardware station

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic explains how to configure, download, and install Retail hardware station by using self-service. It also explains how to uninstall Retail hardware station.

## IMPORTANT

It is critical to note that this component utilizes a server certificate. Server certificates must be managed for expiration. By default, a certificate expires in one calendar year (365 days).

## Download Retail hardware station by using self-service

### Configure a new Retail hardware station

#### NOTE

If you're running the February 2016, non-upgraded version of Retail (Initial release), skip step 6.

1. Use your Azure AD credentials to sign in to the Retail trial.
2. On the **Welcome** page, use the menu in the upper left to go to **Retail > Channels > Retail stores > All retail stores**.
3. On the **All retail stores** page, select the retail channel ID of the desired store. The details view for the store appears.

#### NOTE

The Houston store is the most thoroughly prepared store in the demo data.

4. On the **Retail store details** page, on the **Hardware stations** FastTab, select **Add**.

#### NOTE

The Retail Server URL that is used for the selected store is read-only. This URL will be important during the installation of Retail hardware station.

5. In the **Hardware station type** field, select **Shared** to indicate that this hardware station is an Internet Information Services (IIS), installed hardware station that will be used by external point of sale (POS) systems.

#### NOTE

The value **Shared** signifies that the installation is a truly shared hardware station installation, and that it works through HTTPS communication. By contrast, the value **Dedicated** signifies that the hardware station is a part of Modern POS, and that it works through inter-process communication.

6. Select a hardware station profile.

7. Enter the host name of the computer that you're installing Retail hardware station on. Additionally, enter the electronic funds transfer (EFT) terminal ID that is associated with that computer for merchant account information.
8. To utilize the configuration file or initial installation using mass deployment, enter the certificate thumbprint that is to be used during the installation that's detailed in the next section.

#### Download the Retail hardware station installer

1. Use your Azure AD credentials to sign in to the Retail headquarters or Retail trial.
2. On the **Welcome** page, use the menu in the upper left to go to **Retail > Channels > Retail stores > All retail stores**.
3. On the **All retail stores** page, select the retail channel ID of the desired store. The details view for the store appears.

##### NOTE

The Houston store is the most thoroughly prepared store in the demo data.

4. On the **Retail store details** page, select the **Hardware stations** FastTab.

##### NOTE

The Retail Server URL that is used for the selected store is read-only. This URL will be important during the installation of Retail hardware station.

5. Select the hardware station to download, and then select **Download**.

##### NOTE

- Browsers might block the download pop-up that is generated. You must select either **Allow once** or **Options for this site > Always allow**. Then select **Download** again.

6. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)
7. If needed for mass deployment or command line deployment, repeat the above steps for the configuration file download, which is a button next to the **Download** button that you previously selected.

##### NOTE

- If the configuration file downloaded does not have the same base file name as the installer, either rename the XML configuration file to be the same base name or run the installer using the command line to specify the configuration file.
- Note that the configuration file is not required for the installation of Commerce hardware station.

8. After the files have been saved, run the installer. (This step might differ depending on your browser.)

#### Run the installer

##### NOTE

Before you run the Retail hardware station installer, make sure that all [system requirements](#) are met.



The Retail hardware station installer first extracts the associated files and then begins the installation.

1. The installer validates that all prerequisites are met. If a sideloading key is required, the installer requests it. This key is found on the **Devices** page for each device, under **General**.

**NOTE**

- If a system restart is required, the installer informs you of this requirement but can continue the installation.
- Before you can use hardware that is based on the Object Linking and Embedding for Retail Point of Sale (OPOS) standard, the OPOS Common Control Objects must be installed. If they aren't installed, the installer informs you of this requirement but can continue the installation.

2. Enter the Retail Server URL (for example, `https://MyCompanyNameret.axcloud.dynamics.com/Commerce`), and then select **Next**.

**NOTE**

You can find the Retail Server URL at the top of the **Hardware stations** FastTab on the **Retail store details** page.

3. Select a valid Secure Sockets Layer (SSL) certificate to use for HTTPS communication.

**NOTE**

The certificate must use private key storage, and server authentication must be listed in the enhanced key usage property. Additionally, the certificate must be trusted locally, and it can't be expired. It must be stored in the personal certificate store location on the local computer.

4. The next page requests the user that should be used for the IIS application pool. By default in version 1611 and later, the installer can automatically create and use a service account. If you're on a domain or require more specific controls, clear the check box, and then enter the user name and password that the application pool should run under.
5. Enter the HTTPS port to use.

**NOTE**

- You can find the HTTPS port in Retail. (See the configuration instructions earlier in this topic).
- The installer automatically enters the host name. If, for any reason, you must change the host name for the installation, you can change it here. The host name must be the fully-qualified domain name (FQDN) of the system, and it must be entered in the **Host name** field for the selected hardware station entry.

6. The installer installs Retail hardware station and then indicates whether the installation was successful.
7. When the installation is completed, the Install merchant information tool may start. This installer connects to the environment and installs the merchant account information (such as the EFT ID) for the selected hardware station.

#### NOTE

- If the hardware station that was installed won't be used for payment-related work, don't close the **Install merchant information** window without completing the remaining steps. The hardware station won't work unless this installation is successfully completed.

- For version 10.0.6 and above, the install merchant information tool is no longer used. Instead, the merchant information for the hardware station is set by the POS at the time of logon or when the hardware station is made active. If the retail server is not available when the hardware station is subsequently made active, the last known merchant properties will be used by until the connection to the retail server is re-established. If the POS client is not upgraded to version 10.0.6 at the same time the hardware station is upgraded, merchant properties will not be updated until the POS client is upgraded to an equal or later version.

8. The Install merchant information tool might request Azure AD credentials. Enter the Azure AD credentials of the user who is installing Retail hardware station.
9. The Retail Server URL is determined through the Retail hardware station installation and is entered automatically. The installer uses this URL to load the list of stores that the user is connected to via the address book.
10. Select the retail store that the hardware station was installed for.
11. Select the hardware profile that matches the hardware station that was installed on the current computer.
12. Verify that the host names and EFT terminal IDs are correct, based on the current computer and the Retail hardware station configuration that has already been completed in Retail. After you've verified this information, select **Install**.
13. When you receive a message that states that the merchant account information was installed correctly, exit the installer by selecting the **Close** button.

## Help secure Retail hardware station

Current security standards state that the following options should be set in a production environment:

#### NOTE

The hardware station installer automatically makes these registry edits as part of the installation through self-service.

- SSL should be disabled.
- Only Transport Layer Security (TLS) version 1.2 (or the current highest version) should be enabled and used.

## NOTE

By default, SSL and all version of TLS except TLS 1.2 are disabled. To edit or enable these values, follow these steps:

1. Press the Windows logo key+R to open a **Run** window.
2. In the **Open** field, type **Regedit**, and then select **OK**.
3. If a **User Account Control** window appears, select **Yes**.
4. In the new **Registry Editor** window, go to **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\SecurityProviders\SCHANNEL\Protocols**.  
The following keys have been automatically entered to allow for TLS 1.2 only:
  - TLS 1.2\Server:Enabled=1
  - TLS 1.2\Server:DisabledByDefault=0
  - TLS 1.2\Client:Enabled=1
  - TLS 1.2\Client:DisabledByDefault=0
  - TLS 1.1\Server:Enabled=0
  - TLS 1.1\Client:Enabled=0
  - TLS 1.0\Server:Enabled=0
  - TLS 1.0\Client:Enabled=0
  - SSL 3.0\Server:Enabled=0
  - SSL 3.0\Client:Enabled=0
  - SSL 2.0\Server:Enabled=0
  - SSL 2.0\Client:Enabled=0

- No additional network ports should be open, unless they are required for known, specified reasons.
- Cross-origin resource sharing must be disabled and must specify the allowed origins that are accepted.
- Only trusted certificate authorities should be used to procure certificates that will be used on computers that run Retail hardware station.

## IMPORTANT

- Most common, lower-security software and services will stop working after all lower-security standards are disabled. To use them again, go to the preceding registry keys, and set the **Enabled** key from **0** to **1**.
- It's critical that you review security guidelines for IIS and Payment Card Industry (PCI) requirements.

## Troubleshooting

**Modern POS can detect the hardware station in its list for selection, but it can't complete the pairing**

**Solution:** Verify the following list of potential failure points:

- The computer that is running Modern POS trusts the certificate that is used on the computer that runs Retail hardware station.
  - To verify this setup, in a web browser, go to the following URL:  
`https://<Computer Name>:<Port Number>/HardwareStation/ping`
  - This URL uses a ping to verify that the computer can be accessed, and the browser indicates whether the certificate is trusted. (For example, in Internet Explorer, a lock symbol appears in the address bar. When you select this symbol, Internet Explorer verifies whether the certificate is currently trusted. You can install the certificate on the local computer by viewing the details of the certificate that is shown.)
- On the computer that runs Retail hardware station, the port that will be used by the hardware station is opened in the firewall.

- Retail hardware station has properly installed merchant account information through the Install merchant information tool that runs at the end of the Retail hardware station installer.

### **Modern POS can't detect the hardware station in its list for selection**

**Solution:** Any one of the following factors can cause this issue:

- Retail hardware station hasn't been set up correctly in Commerce headquarters. Use the steps earlier in this topic to verify that the hardware station profile and the hardware station are correctly entered.
- The jobs haven't been run to update the channel configuration. In this case, run the 1070 job for channel configuration.
- The hardware station isn't accessible from that computer. Verify that the hardware station URL ping test is accessible from a web browser. This URL can be found at the end of the hardware station installer and is in the following form: `https://<Computer Name>:<Port Number>/HardwareStation/ping`

## Uninstall Retail hardware station

You can use Control Panel in Microsoft Windows to uninstall Retail hardware station.

1. Press the Windows logo key, and then, in the search box, type **Control Panel**. In the list of search results, select **Control Panel**.
2. In Control Panel, select **Programs > Uninstall a program**. The **Programs and Features** window opens.
3. Select **Microsoft Dynamics 365 for Retail hardware station**, and then select **Uninstall** above the list of programs.
4. Wait for the uninstaller to finish removing the program.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Mass deployment of self-service components

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how you can use self-service to do silent servicing updates and initial deployments. It also explains some aspects of special deployment. This topic will be updated as the feature is developed and more functionality becomes available. Currently, only the capability for silent servicing updates is available.

## Delimiters for mass deployment

The following table shows the delimiters that can currently be used in execution commands for mass deployment.

DELIMITER	DESCRIPTION
-S or -Silent	Silently run the installer. No graphical user interface (GUI) is used. The <b>-Q</b> and <b>-Quiet</b> delimiters have the same effect and can also be used.
-C or -Config	Specify the location and file name of the configuration file to use as part of this installation.
-FilePath	Specify a custom installation location. We don't recommend that you use this delimiter for a standard installation.
-LogFile	Specify a custom file location for the installation logs. We don't recommend that you use this delimiter for a standard installation.
-SkipPrerequisiteCheck	Skip the check for prerequisites and prerequisite installation. You should use this delimiter only for development and testing. We don't recommend that you use it for a standard installation.
-SkipSystemInfoCollection	Skip the process of collecting system information at the beginning of the installation. You should use this delimiter only for development and testing. We don't recommend that you use it for a standard installation.
-SkipMerchantInfo	Skip the installation of merchant account information at the end of the self-service installer for Hardware station. You should use this delimiter only for development and testing. We don't recommend that you use it for a standard installation.

DELIMITER	DESCRIPTION
-SkipAppxInstallation	Beginning in the October 2018 release of Dynamics 365, this delimiter will skip the installation of the APPX Retail Modern POS application. This delimiter is required to perform the application installation through the SYSTEM account or a service account (Any account that does not have a user profile).

## Silent servicing

### Before you begin

Note that silent servicing maintains all components that are currently installed. If any configuration is still required, complete it before you begin to follow the instructions in this topic.

### Examples of commands for silent servicing

This section shows examples of commands that are used for self-service mass deployment. These commands work for all the standard self-service installers, such as the installers for Modern POS (both the installer that has offline support and the installer that doesn't have offline support), Hardware station, and Commerce Scale Unit (self-hosted).

#### Silently update the current installation of Modern POS

The following command silently updates the current installation of Modern POS. This command has the standard command structure that is used for silent servicing of components that are currently installed. The structure uses the basic values of `<InstallerName>.exe` and the command for silent installation, `-S`. This command uses the configuration file that is located in the same file location as the installer, if a configuration file exists there.

```
ModernPOSSetup_V72.exe -S
```

#### NOTE

A configuration file is still required for Retail Store Scale Unit. However, the installer keeps all the values that are currently installed, whenever it can.

#### Silently update the current installation of Commerce Scale Unit (self-hosted)

The following command silently updates the current installation of Commerce Scale Unit (self-hosted) by using a specific configuration file. (This configuration file might not be in the same location as the executable file for the installer.) This command skips the prerequisite check and moves on to the installation steps. We recommend that you use this command only for testing and development purposes.

```
StoreSystemSetup_V72.exe -S -C "C:\Temp\StoreSystemSetup_V72_Houston.xml" -SkipPrerequisiteCheck
```

## Mass deployment of Modern POS

### Before you begin

To use this functionality, you must be using version 7.3 or later. It's assumed that the configuration of all stores, registers, and devices, and other configurations in the headquarters have already been completed. If any configuration is still required, complete it before you follow the instructions in this topic.

### Examples of commands for silent mass deployment

This section shows examples of commands that are used for self-service mass deployment of Modern POS, even Modern POS with offline and the installer without offline support. Examples of Windows PowerShell scripts are also included to help users do the installations.

#### Silently install Modern POS

The following command silently installs (or updates) Modern POS. It has the standard command structure that is used for silent servicing of components that are currently installed. The structure uses the basic values of `<InstallerName>.exe` and the command for silent installation, `-S`.

This command uses the configuration file that is in the same location as the executable file for the installer, if a configuration file exists there. It should not be used if multiple configuration files are available.

```
ModernPOSSetup_V73.exe -S
```

#### NOTE

A configuration file isn't required for Modern POS. However, the Modern POS application that is installed can't be activated in the appropriate manner unless the associated configuration file can be read from.

#### Silently install Modern POS by using a specific configuration file

The following command silently installs the current installation of Modern POS by using a specific configuration file. This configuration file might not be in the same location as the executable file for the installer, or multiple configuration files might be available.

```
ModernPOSSetup_V72.exe -S -C "C:\Temp\ModernPOSSetup_V73_Houston-3.xml"
```

#### Silently install Retail hardware station

#### NOTE

The `-SkipMerchantInfo` delimiter is required to install Retail hardware station. The Merchant Account Information Utility that is opened at the end of a GUI-based installation of Hardware station no longer has to be used. Because of feature functionality, when Modern POS is paired to Hardware station, it also pushes the latest merchant account information to the component.

The following command silently installs (or updates) Retail hardware station. It has the standard command structure that is used for silent servicing of components that are currently installed. The structure uses the basic values of `<InstallerName>.exe` and the command for silent installation, `-S`. It also uses the `-SkipMerchantInfo` delimiter to skip the download of merchant account information through the utility. This command uses the configuration file that is in the same location as the executable file for the installer.

```
HardwareStationSetup_V10.exe -S -SkipMerchantInfo
```

#### NOTE

A configuration file is required to silently deploy Retail hardware station.

#### Silently install Retail hardware station by using a specific configuration file

The following command silently installs the current installation of Retail hardware station by using a specific configuration file. This configuration file might not be in the same location as the executable file for the installer, or multiple configuration files might be available.

```
HardwareStationSetup_V10.exe -S -SkipMerchantInfo -C "C:\Temp\HardwareStationSetup_V10__20-19-35.xml"
```

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Commerce component events for diagnostics and troubleshooting

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains where to find events from Commerce-specific components. To enable diagnostics and troubleshooting, Commerce components, which include self-hosted components such as the Retail Modern POS and cloud-hosted components, such as Commerce Scale Unit and E-Commerce modules, log their events locally to Event Viewer (or to the browser developer tools console such as F12). Events are also logged in the Microsoft Dynamics Lifecycle Services (LCS) log search experience.

## Viewing events in Event Viewer

You can use Event Viewer to view events for components that are installed on computers that run Microsoft Windows, if you have physical access to the computer where the events are logged. For more information about Event Viewer, see [Event Viewer](#) on TechNet. You can also use Event Viewer to view events remotely from computers that you have access to. For more information about how to use Event Viewer to view events remotely, see [Work with Event Logs on a Remote Computer](#) on TechNet. Typically, Event Viewer is used for troubleshooting in the following use cases:

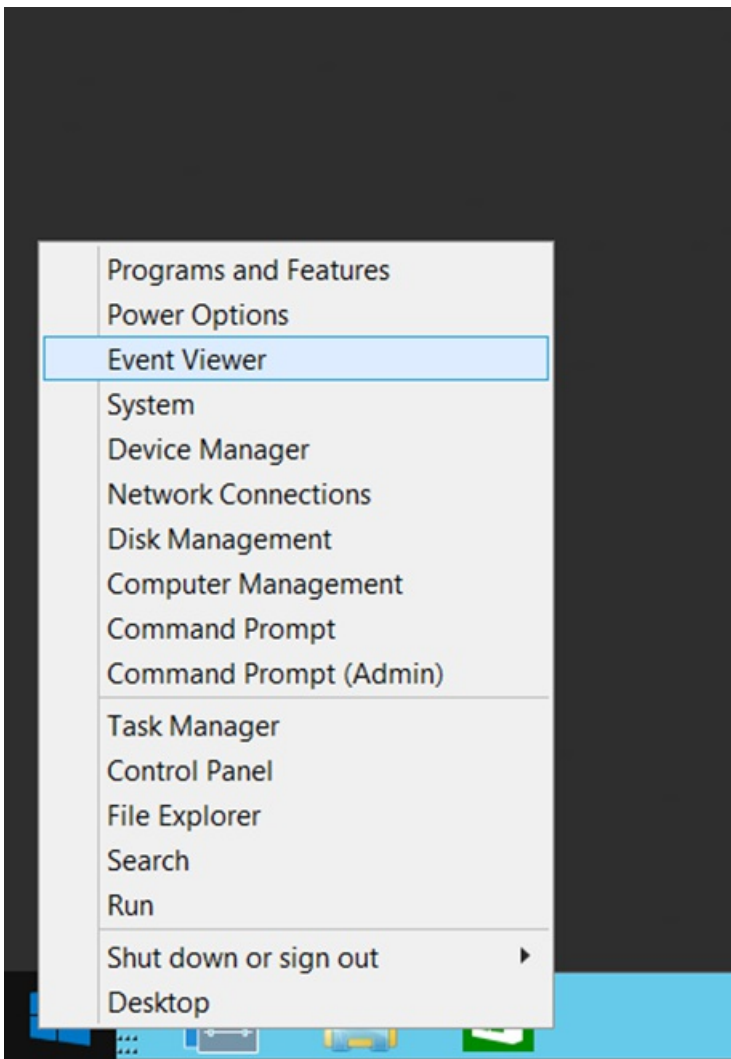
- Development on a developer topology or on a downloadable virtual hard disk (VHD) that provides access to Event Viewer.
- Client components, when you're running a conference room pilot and have access to Event Viewer for that computer.

However, for most other cases, and especially when you don't have access to Event Viewer for the computer, you can use Log Search on LCS. For E-Commerce modules, events are currently available only in browser developer tools (such as F12). Log Search is discussed later in this topic. This section applies to the following components:

- Commerce Scale Unit
- Retail Modern POS
- Retail Hardware Station

### Find Commerce-specific events in Event Viewer

To start Event Viewer on a computer, right-click the **Start** button, and then click **Event Viewer**.



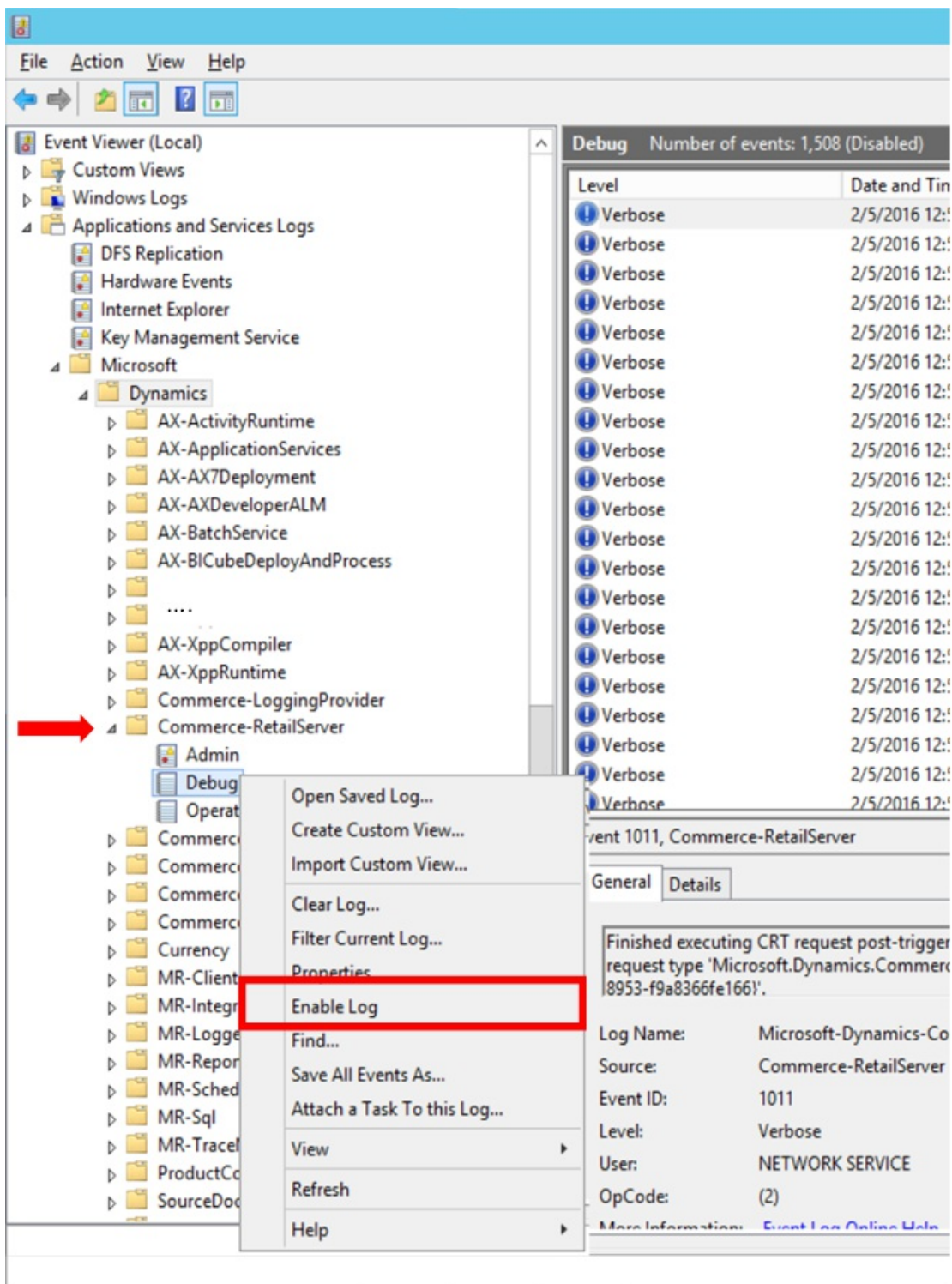
All Commerce-specific event logs can be found under the following path in Event Viewer: Application and Services Logs\Microsoft\Dynamics We provide the following Commerce-specific event logs:

- **Commerce-RetailServer** – This log contains events that are logged by the Commerce Scale Unit components.
- **Commerce-ModernPos** – This log contains events that are logged by Retail Modern POS. These events include events from the TypeScript and C# (CRT) layer.
- **Commerce-LoggingProvider** – This log contains events that are logged by all other Commerce components that aren't included in the list earlier in this article.

### **Enable debug event logs**

Currently, some of the events that are logged by various components are sent to debug event logs. These events are verbose events that are logged at very high rates and are useful only for detailed debugging scenarios. Follow this step to enable the debug event logs in Event Viewer.

- Right-click a debug log, and then click **Enable Log**.

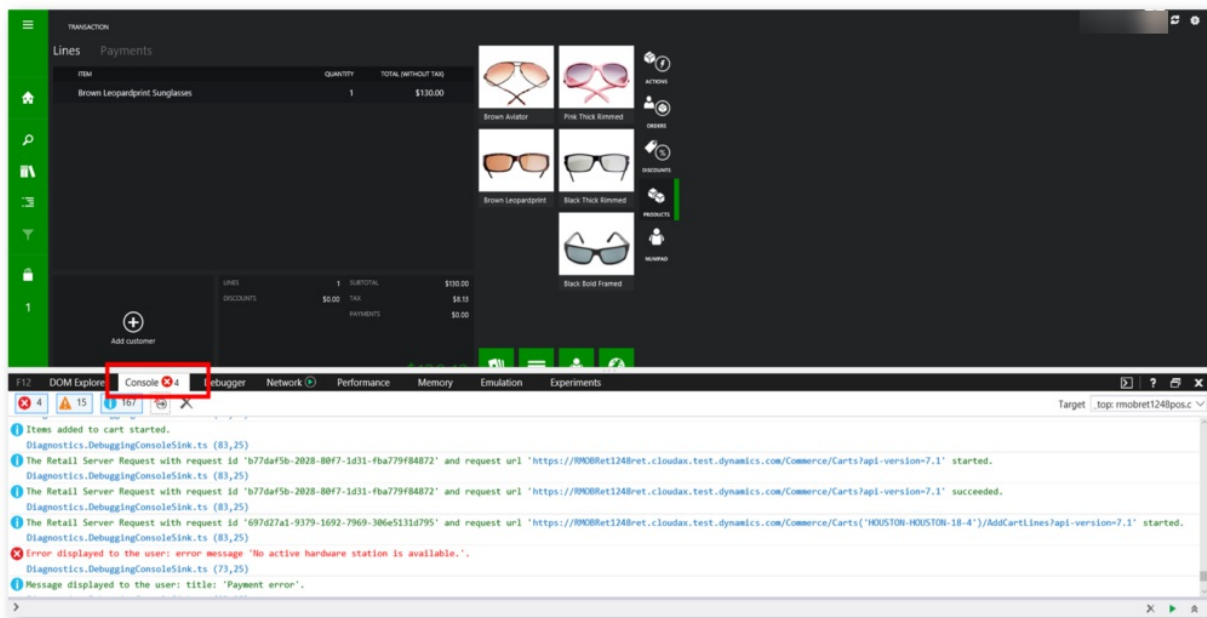


## Viewing events by using the (F12) browser developer tools console

Because Retail Cloud POS and E-Commerce modules are browser-based components, you can use the browser developer tools console to view events for it. For information about the Microsoft browser developer tools console, see [Using the Console to view errors and debug](#). To use the browser developer tools for Retail Cloud POS or E-Commerce modules, you must use a supported browser version.

### View events in the browser developer tools console

1. Start your browser, and navigate to Retail Cloud POS or your E-Commerce website.
2. Press F12, and then click the **Console** tab.
3. As you perform operations on Retail Cloud POS or on your E-Commerce website, events are logged in the console. You can filter by event severity to view events that have different severity levels.



## Correlating events

This section explains how to correlate events from various Commerce components.

### Data flow between a POS client and Commerce Scale Unit

The diagram that follows shows the data flow between a point of sale (POS) client and the Commerce Scale Unit.

#### POS client startup

When a user starts a POS client, a new AppSessionID is generated. The AppSessionID is used to log every event that is instrumented in the POS client. All events that are logged to Event Viewer and App Insights have this ID.

#### User sign-in

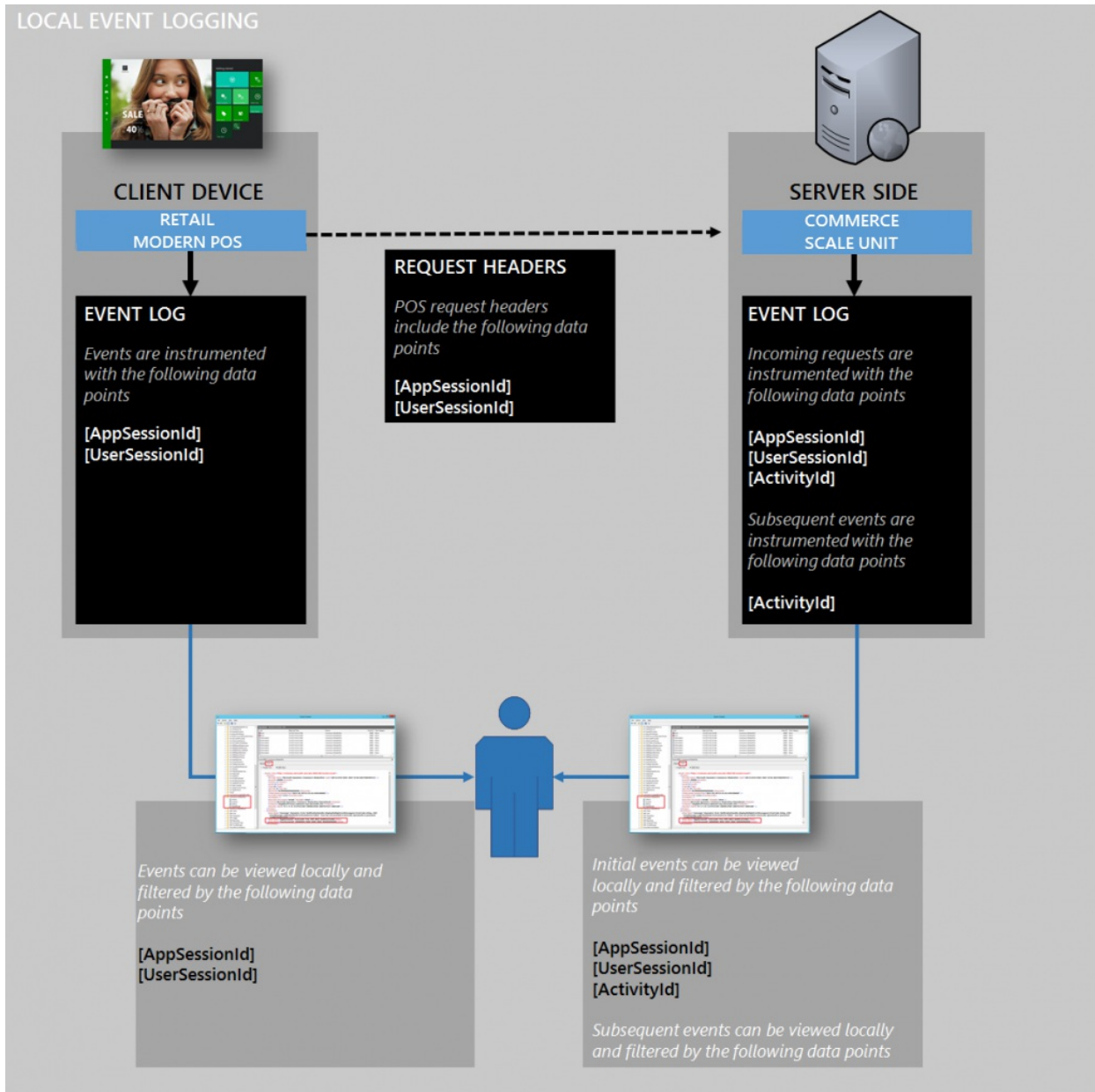
When a user signs in to a POS client, a new UserSessionID is generated. The UserSessionID is used to log every event that is instrumented in the POS client. All user events that are logged to Event Viewer have this ID. This ID is maintained for as long as the user is signed in. When the current user signs out and a new user sign in, a new UserSessionID is generated.

#### POS client calls to Commerce Scale Unit

Whenever a POS client makes a call to the Commerce Scale Unit, the AppSessionID and UserSessionID are sent as headers. The Commerce Scale Unit then logs an event for the incoming request (Event ID 5000). This event includes those two IDs and also an ActivityID. The ActivityID is then also used for all related events. The AppSessionID, UserSessionID, and ActivityID are available in the event log where Commerce Scale Unit is hosted. They are also available in LCS Log Search.

#### Request activity on Commerce Scale Unit

Every event that is logged as part of a Commerce Scale Unit request has the same ActivityID as the initial event that was logged for the initial incoming request event (Event ID 5000). These events are available in both Event Viewer and LCS Log Search.

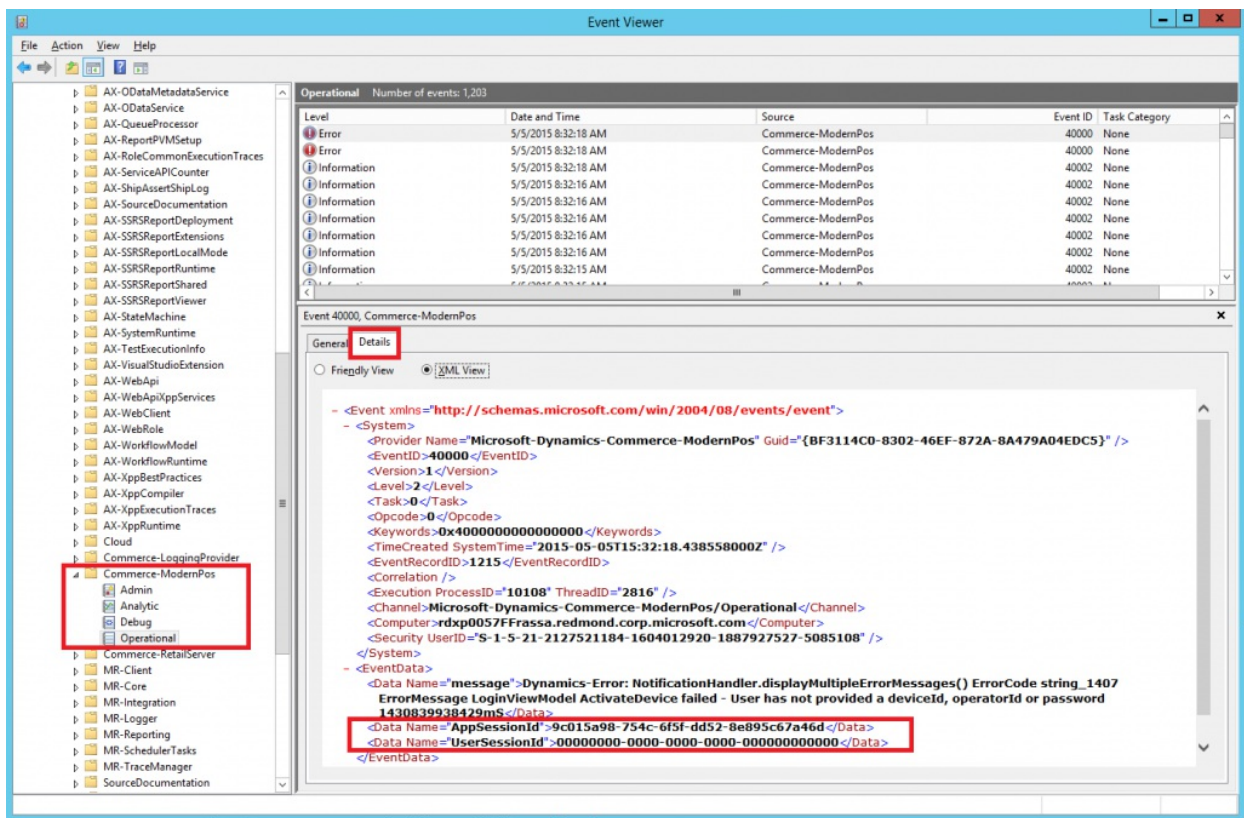


### Finding Retail Modern POS events in Event Viewer

Every event that is logged by Retail Modern POS includes the following data points:

- **AppSessionID** – A unique ID that is generated when the app is first started. It's included with every event that is logged.
- **UserSessionID** – A unique ID that is generated when a user signs in to Retail Modern POS. It's included with every event that is logged, for as long as the user remains signed in. When a new user signs in, a new UserSessionID is created.

You can find the AppSessionID and UserSessionID values on the **Details** tab in Event Viewer on the machine where Retail Modern POS is installed.



## Finding incoming Commerce Scale Unit request events in Event Viewer

To correlate data for incoming Commerce Scale Unit requests in Event Viewer, you must first enable the Analytic channel. To enable the Analytic channel, follow these steps.

1. In Event Viewer, in the left pane, select **Commerce-RetailServer**.
2. Click **View > Enable Analytic and Debug log**. A new node for the Analytic channel appears under the **Commerce-RetailServer** logging provider.
3. Right-click the **Analytic** node, and then click **Enable log**.

In Event Viewer, all incoming Commerce Scale Unit requests are logged to the Analytic channel of the Commerce-RetailServer source as event 5000. These events also have the AppSessionID and UserSessionID that were described earlier. Every event also has a unique ActivityID that is instrumented for every logged event for the same request.

## Using LCS Log Search

LCS Log Search lets you view data from all the components from a single portal. You can access events from both cloud-hosted components (such as Commerce Scale Unit) and in-store components (such as Retail Modern POS and Retail Hardware Station). Event data from all cloud-hosted and in-store components flows to LCS Log Search, where it's indexed and made searchable. Data is typically available within 5 minutes after it's logged. For POS clients and Retail Hardware Station, all events are locally queued in persistent storage and then uploaded in batches after the queue is filled. This behavior enables network traffic to be optimized. It also enables events to be saved even when there is no Internet connectivity. After connectivity is restored, all pending events are uploaded.

LCS Log Search is available for the HA production topology. It can be used for the following Commerce components:

- Retail Modern POS
- Retail Cloud POS
- Commerce Scale Unit (running on Retail Cloud Scale Unit)

LCS Log Search does not include logs from the following Commerce components:

- Commerce layout designer
- Commerce receipt designer
- Self-service installer for Retail Modern POS
- Self-service installer for Retail Hardware Station
- Commerce Scale Unit (running on Retail Store Scale Unit)
- Retail Hardware Station

### Access LCS Log Search

To access LCS Log Search, follow these steps.

1. Go to [Lifecycle Services](#).
2. Sign in by using the credentials that are associated with your project.
3. On the project page, select the correct project.
4. On the **Project details** page, select the correct environment.
5. On the **Environment details** page, click **Environment Monitoring**.
6. On the **Environment monitoring** page, click **View raw logs**.
7. On the **Log Search** page, select one of the following queries:
  - **Commerce client events** query, which includes events from Retail Modern POS, Retail Cloud POS, and Commerce Scale Unit (running on Retail Cloud Scale Unit)
  - **All logs** query, which includes data from Commerce Scale Unit, Commerce Data Exchange, and Commerce Data Exchange: Real-time Service

You can filter by the following criteria to refine your query:

- Start and end dates and times (in Coordinated Universal Time [UTC])
- Device ID
- POS user
- POS application session ID
- POS user session ID
- Severity level

The screenshot displays the 'Raw logs' section of the 'Environment monitoring' page in Lifecycle Services. The interface includes a search bar with the query name 'Commerce error events' and several filter options on the left, such as 'Start date (UTC)', 'End date (UTC)', 'Device ID', 'POS User', 'POS Application Session ID', 'POS User Session ID', and 'Filter Severity Logs'. The main area contains a table with columns for various identifiers and internal data. The table shows three rows of log entries, each with a severity level of 1.61. The first row's internal data includes fields like 'OfflineAvailability', 'Unknow', 'Tana-443d-5b3c', '885932ac220', 'Application', 'POS', 'UserSessionId', '00000000-0000-0000-0000-000000000000', 'DeviceNum', '917a-492e-86b3', 'e242e18726c9', 'TerminalId', '2cbe-f7d6-c560', '149fad12c789', 'UserId', 'Invalid parameter (at index 0) cannot be null or undefined.', 'errorUrl', 'https'. The second row's internal data includes 'Tana-443d-5b3c', '885932ac220', 'UserSession', '0000-0000-0000-000000000000', 'Application', 'POS', 'TenantId', '52bf87264-917a-492e-86b3', 'e242e18726c9', 'UserId', 'Invalid parameter (at index 0) cannot be null or undefined.', 'errorUrl', 'https'. The third row's internal data includes '1f781371-cbaa-11e5-ba36-1.61', '965a5a962867', 'EventSeverity', 'Error', 'App', 'Tana-443d-5b3c', '885932ac220', 'UserSession', '0000-0000-0000-000000000000', 'Application', 'POS', 'TenantId', '52bf87264-917a-492e-86b3', 'e242e18726c9', 'UserId', 'Invalid parameter (at index 0) cannot be null or undefined.', 'stackTrace', 'https'. The page footer shows 'Page 1 of 1' and 'View 1 - 5 of 6'.

## E-Commerce events

The following events are logged by the E-Commerce website, and can be consumed for troubleshooting directly in the browser, or programmatically by partner extensions for analytics, experimentation, or other purposes.

### Button and link clicks

Button and link clicks for the following types of elements on an E-Commerce website are logged as telemetry events.

- Header
  - Navigation hierarchy
  - Cart icon
  - Sign-in
  - Search icon
  - Wishlist icon
- Content block action links (This represents the hero, tile, and feature modules for marketing content.)
- Video player
- Product cards
- Footer links
- Breadcrumbs
- Promo banner
- Add to cart button
- Checkout button
- Place order button

The schema for **Click** action is:

```
Click
IPayload = {
  contentCategory: Name of element clicked on,
  contentAction: {
    pname: Name of page,
    mname: name of module,
    etext: Text of element clicked on,
    recid: Product ID if a product was clicked on,
    etype: 'click',
  }
};
```

### Page views

Page view events are logged for each page view operation.

The schema for a **PageView** action is:

```
PageView
IPageViewInfo = {
  title;
}
```

### Cart operations

The following **Cart** related events are logged.

- Add item to cart.
- Update item in cart.
- Remove item from cart.



- Checkout.
- Product Page view.

The schema for **Cart** events is:

```
/**
 * Defines the telemetry properties to track for a Cart object
 * @property products      {IProductInfo[]} - Array of product information
 * @property orderId      {string}         - ID for the order
 * @property cartId       {string}         - ID for the current cart object
 * @property cartVersion  {string}         - Version number for the current cart object
 */
export interface ICartInfo {
  products: IProductInfo[];
  orderId: string;
  cartId: string;
  cartVersion: string;
}
```

## Purchase

When an order is submitted, a Purchase event is logged. The schema for a **Purchase** event is:

```
/**
 * Defines the telemetry properties to track for a Purchase event
 * @property id           {string}         - Transaction ID
 * @property affiliation  {string}         - Origin of this transaction (e.g. Online Store)
 * @property revenue     {number}         - Revenue from this transaction
 * @property tax         {number}         - Tax amount
 * @property shippingCost {number}         - Shipping cost
 * @property products    {IProductInfo[]} - List of products in this transaction
 */
export interface IProductTransaction {
  id: string;
  affiliation?: string;
  revenue?: number;
  tax?: number;
  shippingCost?: number;
  products?: IProductInfo[];
}
```

## Product details

Product details are logged for **Cart** and **Purchase** operations. The schema for **Product** details is:

```
/**
 * Defines the telemetry properties to track for a Product object
 * @property productChannelId {string} - Product channel ID
 * @property productChannelName {string} - Product channel name
 * @property productCategoryId {string} - Product category ID
 * @property productCategoryName {string} - Product category name
 * @property productId {string} - Product ID
 * @property productName {string} - Product name
 * @property productSku {string} - Product SKU
 * @property productPrice {string} - Product price
 * @property productQuantity {string} - Product quantity
 * @property productCurrency {string} - Product currency code
 */
export interface IProductInfo {
  productChannelId: string;
  productChannelName: string;
  productCategoryId: string;
  productCategoryName: string;
  productId: string;
  productName: string;
  productSku: string;
  productPrice: string;
  productQuantity: string;
  productCurrency: string;
}
```

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Apply updates to cloud environments

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic describes how you can use Microsoft Dynamics Lifecycle Services (LCS) to automatically apply updates to cloud environments.

## IMPORTANT

Updates are applied using deployable packages. Applying updates causes system downtime. All relevant services will be stopped, and you won't be able to use your environments while the package is being applied. You should plan accordingly.

## Supported environments

All environments deployed through Lifecycle Services are supported.

## NOTE

If you have a build environment, you can only use LCS to apply Binary updates and Data upgrade packages. You can't use LCS to apply an Application Deployable package.

For other environments (listed below), you must use Remote Desktop Protocol (RDP) to connect to the environment and install from the command line. For information about manual package deployment, see [Install deployable packages from the command line](#).

- Local development environments (Downloadable virtual hard disk [VHD])
- Multi-box dev/test environments in Microsoft Azure (Partner and trial projects)

## Key concepts

Before you begin, you should understand *deployable packages*, *runbooks*, and the *AXInstaller*. A deployable package is a unit of deployment that can be applied in any environment. A deployable package can be a binary update to the platform or other runtime components, an updated application (AOT) package, or a new application (AOT) package. The AXInstaller creates a runbook that enables installing a package. For more details, see [Packages, runbooks, and the AXUpdateInstaller in depth](#) at the end of this topic.

## Supported package types

- **AOT deployable package** – A deployable package that is generated from application metadata and source code. This deployable package is created in a development or build environment.
- **Application and Platform Binary update package** – A deployable package that contains dynamic-link libraries (DLLs) and other binaries and metadata that the platform and application depend on. This is a package released by Microsoft. This is available from the **All binary updates** tile from LCS.
- **Platform update package** – A deployable package that contains dynamic-link libraries (DLLs) and other binaries and metadata that the platform depend on. This is a package released by Microsoft. This is available from the **Platform binary updates** tile from LCS.
- **Commerce deployable package** – A combination of various packages that are generated after the Commerce code is combined.
- **Merged package** – A package that is created by combining one package of each type. For example, you can

merge one binary update package and one AOT package, or one AOT package and one Commerce deployable package. The packages are merged in the Asset library for the project in LCS.

#### NOTE

A binary package and a Commerce deployable package can't be included in the same merged package.

For information about how to download an update from LCS and what you see in the tiles based on your environment version, see [Download updates from Lifecycle Services \(LCS\)](#).

If your environment is on an application version 8.1 and later, then the **Platform Update package** does not apply to your environment. Starting with 8.1 and later releases, **Application and Platform Binary update package** is the one that applies since application and platform will be combined into a single cumulative package and will be released by Microsoft. Also note that you will no longer be applying granular X++ hotfixes and will get all application and platform updates together. This means that on the environment details page, clicking on **View detailed version information** will not have details on the granular hotfixes or KBs applied as there is no way to apply them.

## Prerequisite steps

- **Make sure that the package that should be applied is valid.** When a package is uploaded to the Asset library, it isn't analyzed. If you select the package, the package status appears in the right pane as **Not Validated**. A package must pass validation before it can be applied in an environment by using the following procedures. The status of the package will be updated in the Asset library to indicate whether the package is valid. We require validation to help ensure that production environments aren't affected by packages that don't meet the guidelines.

There are three types of validations:

- Basic package format validations
  - Platform version checks
  - Types of packages
- **Make sure that the package is applied in a sandbox environment before it's applied in the production environment.** To help ensure that the production environment is always in a good state, we want to make sure that the package is tested in a sandbox environment before it's applied in the production environment. Therefore, before you request that the package be applied in your production environment, make sure that it has been applied in your sandbox environment by using the automated flows.
  - **If you want to apply multiple packages, create a merged package that can be applied first in a sandbox environment and then in the production environment.** Application of a single package in an average environment requires about 5 hours of downtime. To avoid additional hours of downtime when you must apply multiple packages, you can create a single combined package that contains one package of each type. If you select a binary package and an application deployable package in the Asset library, a **Merge** button becomes available on the toolbar. By clicking this button, you can merge the two packages into a single package and therefore reduce the total downtime by half.
  - **Make sure that the Application binary update package is applied to your dev/build environment before it is applied to your sandbox and production environment** - If the application binary package is applied directly to your Tier 2+ sandbox environment but is not applied on your dev/build environment, the next time you move an AOT package from dev/build box (which does not have the same application binaries as your sandbox environment) to sandbox, some of the application binaries will be overwritten with what is in your dev/build environment. This could result in a regression of the version of your sandbox environment.

# Apply a package to a non-production environment by using LCS

Before you begin, verify that the deployable package has been uploaded to the Asset library in LCS.

1. For a binary update, upload the package directly to the Asset library. For information about how to download an update from LCS, see [Download updates from Lifecycle Services \(LCS\)](#). For an application (AOT) deployable package that results from an X++ hotfix, or from application customizations and extensions, create the deployable package in your development or build environment, and then upload it to the Asset library.
2. Open the **Environment details** view for the environment where you want to apply the update.
3. Click **Maintain > Apply updates** to apply an update.
4. Select the package to apply. Use the filter at the top to find your package.
5. Click **Apply**. Notice that the status in the upper-right corner of the **Environment details** view changes from **Queued** to **In Progress**, and an **Environment updates** section now shows the progress of the package. You can refresh the page to check the status.
6. Continue to refresh the page to see the status updates for the package application request. When the package has been applied, the environment status changes to **Deployed**, and the servicing status changes to **Completed**.

# Apply a package to a production environment by using LCS

In a production environment, customers can schedule a downtime for when they want the update to be applied.

## IMPORTANT

An important prerequisite for applying a package to a production environment is that the package must be successfully applied to at least one sandbox environment in the same project.

1. After the update is successfully applied in a sandbox environment, go to the project's asset library. On the **Asset library** page, select the **Software deployable package** tab, select the package that you want to move to production, and click **Release candidate**. This indicates that this package is ready for production deployment.
2. Open the **Environment details** view for the production environment where you want to apply the package.
3. Select **Maintain > Apply updates** to apply the package.
4. Select the package to apply in your production environment, and then click **Schedule** to submit a request to apply it.

## NOTE

The list of packages includes only the packages that have been successfully signed off in the sandbox environment, and that have been marked as release candidates.

5. Specify the date and time to schedule the package application. Click **Submit**, and then click **OK** to confirm. Note that your environments will be unavailable to perform business while the package is being applied.
6. At the scheduled downtime, package deployment will start.
7. After the environment is serviced, you can monitor the status. The **Servicing status** field indicates the status of package application. Additionally, a progress indicator shows the number of steps that have

been run, out of the total number of steps that are available.

8. After the deployment is successfully completed, the **Servicing status** field is set to **Completed**.
9. If package application isn't successfully completed, Microsoft will investigate the issue. The **Servicing status** field will indicate that package application has failed. The environment will be rolled back to a good state.

## Troubleshoot package deployment failures

If package deployment fails, see the [Troubleshoot package application issues](#) topic.

## Applying updates and extensions

If you are updating a Tier-2 Sandbox or Production environment on application version 8.1.2.x or newer and have initialized Cloud Scale Unit, you will also need to update Commerce channel components. For more information, see [Update Retail Cloud Scale Unit](#).

If you're using components (such as Modern POS), after you've applied updates and extensions in your environment, you must also update your in-store components. For more information, see [Configure, install, and activate Modern POS \(MPOS\)](#).

## Packages, runbooks, and the AXUpdateInstaller in depth

Deployable packages, runbooks, and the AXUpdateInstaller are the tools you use to apply updates.

**Deployable package** – A deployable package is a unit of deployment that can be applied in an environment. A deployable package can be a binary update to the platform or other runtime components, an updated application (AOT) package, or a new application (AOT) package. Deployable packages downloaded from LCS or created in a development environment cannot be applied across product types. For example, a Finance deployable package cannot be applied in a Commerce app environment, and vice versa. If you have an existing customization for a Finance and Operations app that is compatible with the Commerce app, and you would like to apply it to a Commerce environment, you will need to re-package your source code in a Commerce development environment, and conversely if moving in the other direction.

AXDeployablePackage\_20160212\_22\_57\_44.zip - ZIP archive, unpacked size 1,221,43; ← Zip format

Name	Size	Pack...	Type
..			File folder
ALMService			File folder
AOSService			File folder
BIService			File folder
DevToolsService			File folder
DIXFService			File folder
MRApplicationService			File folder
MROneBox			File folder
MRProcessService			File folder
PerfSDK			File folder
ReportingService			File folder
RetailCloudPos			File folder
RetailSDK			File folder
RetailSelfService			File folder
RetailServer			File folder
RetailStorefront			File folder
SCMSelfService			File folder
TestAssets			File folder
UserSID			File folder
AutoTriggerETWManifestRefresh.ps1	10,1...	6,065	Window...
AXUpdateInstaller.exe	18,6...	9,365	Applicati...
DefaultServiceModelData.xml	13,4...	724	XML Do...
DefaultTopologyData.xml	1,199	430	XML Do...
HotfixInstallationInfo.xml	2,999	443	XML Do...
Microsoft.Dynamics.AX.AXInstallationInfo.dll	26,3...	12,7...	Applicati...
Microsoft.Dynamics.AX.AXUpdateInstallerBase.dll	42,7...	18,7...	Applicati...
Switch.dll	40,6...	18,9...	Applicati...
System.Management.Automation.dll	2,68...	896...	Applicati...

← Changed files/ folder

← Update Installer

← Modules information

← Topology information

**Runbook** – The deployment runbook is a series of steps that are generated in order to apply the deployable package to the target environment. Some steps are automated, and some steps are manual. AXUpdateInstaller lets you run these steps one at a time and in the correct order.

- Generated based on topology of deployments with multiples VMs
- Contains step by step information for applying deployable package
- Provides sequence of steps across VMs in multi-box/ HA environment
- Integration for apply automation scripts at each step
  - Stop/ start AOS service, batch service
  - Report deployment, DB sync, ...

```
<?xml version="1.0" encoding="UTF-8"?>
- <RunbookData xmlns:xsi="http://www.w3.
  <RunbookID>AXDeployablePackage_20
  + <RunbookTopologyData>
  + <RunbookServiceModelData>
  + <RunbookStepList>
  + <RunbookLogs>
</RunbookData>
```

```
<Name>AX topology</Name>
<MachineList>
- <Machine>
  <Name>AOS-77edc66f7a1</Name>
  - <ServiceModelList>
    <string>AOSService</string>
    <string>DIXFService</string>
    <string>RetailCloudPos</string>
    <string>RetailSelfService</string>
    <string>RetailServer</string>
    <string>SCMSelfService</string>
  </ServiceModelList>
</Machine>
+ <Machine>
- <Machine>
  <Name>BI-4bb1b0a48fa5</Name>
  - <ServiceModelList>
    <string>ReportingService</string>
  </ServiceModelList>
</Machine>
- <Machine>
  <Name>BI-3c0207c4482e</Name>
```

```
- <Step>
  <ID>1</ID>
  <Description>Stop script for service model: AOSService on machine: AOS-77edc66f7a1</Description>
  <MachineName>AOS-77edc66f7a1</MachineName>
  <ServiceModelName>AOSService</ServiceModelName>
  - <ScriptToExecute>
    <FileName>AutoStopAOS.ps1</FileName>
    <Automated>true</Automated>
    <Description>Stop AOS service and Batch service</Description>
    <RetryCount>0</RetryCount>
  </ScriptToExecute>
  <StartTime>2016-02-17T01:14:45.2397318+00:00</StartTime>
  <EndTime>2016-02-17T01:14:48.6772116+00:00</EndTime>
  <StepState>Completed</StepState>
</Step>
+ <Step>
```

**AXUpdateInstaller** – When you create a customization package from Microsoft Visual Studio or a Microsoft binary update, the installer executable is bundled together with the deployable package. The installer generates the runbook for the specified topology. The installer can also run steps in order, according to the runbook for a specific topology.

## Additional resources

## Install deployable packages from the command line

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Initialize seed data in new Commerce environments

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes the data that's created as part of the initialization process for Dynamics 365 Commerce.

After the Commerce solution has been deployed through Microsoft Dynamics Lifecycle Services (LCS), you must initialize the Commerce configuration to create the basic configuration data.

## IMPORTANT

Before you initialize the commerce configuration, make sure that you've specified a language and a postal address for each legal entity where you will set up stores. This step must be completed for each legal entity that you use for commerce.

To initialize the configuration, follow these steps.

1. Start the Commerce client.
2. Click **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**.
3. Click **Initialize**.

Initialization creates the following default configuration data:

- Commerce scheduler jobs and subjobs
- Commerce channel schema
- Commerce distribution schedules
- Default screen layouts, which include button grids, images, and themes
- Time zone information
- Point-of-sale (POS) operations
- POS permissions
- Channel reports
- Attribute metadata
- Entity validation templates
- Batch job to purge Commerce Data Exchange session history

Additionally, logging that is related to the payment card industry (PCI) is enabled for the Commerce database.

## NOTE

There is an option to separately configure the Commerce scheduler. This option lets you reset the Commerce scheduler configuration to its default settings.

After initialization is completed, you must configure additional commerce data. Here are some examples:

- Commerce parameters
- Commerce scheduler parameters
- Commerce channels
- Registers and devices
- Assortments

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Channels overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic presents an overview of channels in Microsoft Dynamics 365 Commerce. It includes information about the tasks that you must complete both before and after you set up each channel.

## Types of Channels

Dynamics 365 Commerce supports three different channel types: retail, call center, and online channels.

### **Retail channels**

Retail channels represent standard brick-and-mortar stores. Each store can have its own point of sale (POS) registers, income and expense accounts, and staff.

### **Call center channels**

Call center channels represent call center order and customer management.

### **Online channels**

Online channels represent online e-Commerce storefronts. Once an online channel is created, a site must be created using the Microsoft Dynamics 365 Commerce Site Builder tool or other third-party e-Commerce solution.

## Channel setup basics

Channel set up is performed in the Commerce tool. Each channel can have its own payment methods, price groups, product hierarchies, assortments, and set of products. After you create a channel, you assign the products that you want it to carry and sell. Each channel type has a unique set of features that may need to be configured. For example, a retail channel needs assigned employees, registers, and customers. Once a new channel is created, it needs to be assigned to an organization hierarchy.

## Channel setup prerequisites

Before you can set up a channel, you must complete some prerequisite tasks based on the channel type. For more information, see [Channel setup prerequisites](#).

## Set up a channel

After you complete the prerequisite tasks, for further setup instructions, use the following links.

- [Set up a retail channel](#)
- [Set up a call center channel](#)
- [Set up an online channel](#)

## Additional resources

[Channel setup prerequisites](#)

[Set up a retail channel](#)

[Set up an online channel](#)

[Set up a call center channel](#)

## Set up organization hierarchies

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Channel setup prerequisites

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic presents an overview of channel setup prerequisites in Microsoft Dynamics 365 Commerce.

## Overview

Before a Dynamics 365 Commerce channel can be created, several prerequisite tasks must be completed. The following lists of prerequisite tasks are organized by channel type.

### NOTE

Some documentation is still being written, and links will be updated as new content is published.

## Initialization

- [Initialize seed data](#)

## Global prerequisites required for all channel types

- [Define and configure your legal entity structure](#)
- [Configure your organizational hierarchy](#)
- [Set up a warehouse](#)
- [Configure sales tax](#)
- [Set up an email notification profile](#)
- [Set up number sequences](#)
- [Set up a default customer and address book](#)

## Retail channel prerequisites

- [Info codes and info code groups](#)
- [Set up a retail functionality profile](#)
- [Set up an employee address book](#)
- [Set up a screen layout](#)
- [Set up a hardware station](#)

## Call Center channel prerequisites

- [Call center parameters](#)
- [Call center order and refund payment methods](#)
- [Call center modes of delivery and charges](#)

## Online channel prerequisites

- [Create an online functionality profile](#)

## Additional resources

[Channels overview](#)

[Organizations and organizational hierarchies overview](#)

[Set up organization hierarchies](#)

[Create legal entities](#)

[Set up a retail channel](#)

[Set up an online channel](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Organizations and organizational hierarchies overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

An organization is a group of people who are working together to carry out a business process or achieve a goal. Organizational hierarchies represent the relationships between the organizations that make up your business.

## Organizations

You can define the following types of internal organizations: legal entities, operating units, and teams.

All internal organizations are types of the **Party** entity. Therefore, these organizations use the address book to store address and contact information. A party, which can be either a person or an organization, can belong to one or more address books.

### Legal entities

A legal entity is an organization that has a registered or legislated legal structure. Legal entities can enter into legal contracts and are required to prepare statements that report on their performance.

A company is a type of legal entity. Currently, companies are the only kind of legal entity that you can create, and every legal entity is associated with a company ID. This association exists because some functional areas in the program use a company ID, or DataArealD, in their data models. In these functional areas, companies are used as a boundary for data security. Users can access data only for the company that they are currently logged on to.

### Operating units

An operating unit is an organization that is used to divide the control of economic resources and operational processes in a business. People in an operating unit have a duty to maximize the use of scarce resources, improve processes, and account for their performance.

The types of operating units include cost centers, business units, value streams, departments, and commerce channels. The following table provides more information about each type of operating unit.

OPERATING UNIT TYPE	DESCRIPTION	PURPOSE
Cost center	An operating unit in which managers are accountable for budgeted and actual expenditures.	Used for the management and operational control of business processes that span legal entities.
Business unit	A semi-autonomous operating unit that is created to meet strategic business objectives.	Used for financial reporting that is based on industries or product lines that the organization serves independently of legal entities.
Value stream	An operating unit that controls one or more production flows.	Commonly used in lean manufacturing to control the activities and flows that are required to supply a product or service to consumers.

OPERATING UNIT TYPE	DESCRIPTION	PURPOSE
Department	An operating unit that represents a category or functional part of an organization that performs a specific task, such as sales or accounting.	Used to report on functional areas. A department may have profit and loss responsibility, and may consist of a group of cost centers.
Commerce channel	An operating unit that represents a brick and mortar store, an online store or an online marketplace.	Used for the management and operational control of one or more stores within or across legal entities.

### Teams

A team is an organization in which the members share a common responsibility, interest, or objective. Teams cannot be used in organizational hierarchies.

## Organizational hierarchies

Set up organizational hierarchies to view and report on your business from different perspectives. For example, you can set up a hierarchy of legal entities for tax, legal, or statutory reporting. Set up a hierarchy that is based on operating units to report financial information that is not legally required, but that is used for internal control. For example, you can create a purchasing hierarchy to control purchasing policies, rules, and business processes.

Each hierarchy is assigned a purpose. The purpose of a hierarchy determines the types of organizations that can be included in the hierarchy. The purpose also determines which application scenarios a hierarchy can be used in.

Organizations in a hierarchy can share parameters, policies, and transactions. An organization can inherit or override the parameters of its parent organization. However, shared master data, such as products and address books, applies to the whole organization and cannot be overridden for individual organizations. Creating organizations and hierarchies requires careful planning. For more information, see [Plan your organizational hierarchy](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Plan your organizational hierarchy

2/18/2021 • 13 minutes to read • [Edit Online](#)

Before you set up organizations and organization hierarchies, make sure that you plan how your business will be modeled. The organization model has a significant effect on the implementation and business processes.

Organizational hierarchies represent the relationships between the organizations that make up a business. Therefore, the most important consideration when you model organizations is the structure of your business. We recommend that you define organization structures based on feedback from executives and senior managers from functional areas, such as finance and accounting, human resources, operations, purchasing, and sales and marketing.

When you are planning hierarchies, it is also important to consider the relationship between the organizational hierarchy and financial dimensions. You can set up multiple organizational hierarchies to represent different views of your business. By using financial dimensions, you can create reports based on these views. Work with your partner to create hierarchies that address both organizational and statutory reporting needs.

## NOTE

Although you can use financial dimensions to represent legal entities without creating the legal entities, financial dimensions aren't designed to address the operational or business needs of legal entities. The interunit accounting functionality is designed to address only the accounting entries that are created by each transaction.

## IMPORTANT

You shouldn't decide how to model organizations based only on the information in this article. This documentation is a guide. You can work with your Partner for additional guidance. Your Partner has gained experience in various industries and across the customer base.

## Decide whether to model internal organizations as legal entities or operating units

You must have at least one legal entity to represent your business. A legal entity can enter legal contracts and is required to prepare financial statements that report on its performance.

Legal entities can be used for transactional business or for consolidation. This means that a legal entity in Finance and Operations does not necessarily represent a real entity in your business. For example, a company that participates in transactions can own subsidiary legal entities. In this scenario, a legal entity is required for transactions, and a virtual legal entity is required to consolidate the results and balances of the subsidiary legal entities.

Internal organizations in your business, such as regional offices, can be represented as additional legal entities, or as operating units of the main legal entity. An operating unit is not required to be a legally defined organization. Operating units are used to control economic resources and operational processes in the business. For example, departments and cost centers are operating units.

Some functionality works differently depending on whether the organization is a legal entity or an operating unit. Carefully consider the functionality described below as you make your decision.

### Master data

**If the organization is modeled as a legal entity**

Some master data, such as customers, payment terms, tax authorities, and site-specific stock ordering, must be set up for each legal entity. Some master data, such as users, products, and most human resources data, is shared among all legal entities.

**If the organization is modeled as an operating unit**

Master data is shared among operating units.

**Module parameters****If the organization is modeled as a legal entity**

Parameters for modules, such as Accounts receivable parameters, Accounts payable parameters, and Cash and bank management parameters, must be set per legal entity. Because the module setup for legal entities is separate, each subsidiary can comply with local statutory requirements and business practices. For example, a professional services legal entity and a manufacturing legal entity can have different module parameters even though they report to the same parent company.

**If the organization is modeled as an operating unit**

Module parameters are shared among operating units.

**Data security****If the organization is modeled as a legal entity**

Most data is automatically secured by company ID. A company ID is a unique identifier for the data that is associated with a legal entity. A company can be associated with only one legal entity, and a legal entity can be associated with only one company. Users can access data only for the companies that they have access to. You do not need to customize to secure data by company ID.

**If the organization is modeled as an operating unit**

Data can be secured per operating unit by creating customized data security policies. Data security policies are used to limit access to data. For example, assume that a user is allowed to create purchase orders only in a particular operating unit. Data security policies can be created to prevent the user from accessing purchase order data from any other operating unit. The volume of transactions and the number of security policies can affect performance. When you design security policies, keep performance in mind.

**Ledgers****If the organization is modeled as a legal entity**

Each legal entity requires a ledger that provides a chart of accounts, accounting currency, reporting currency, and fiscal calendar. A balance sheet can be created only for a legal entity. Main accounts, dimensions, account structures, charts of accounts, and account rules can be used by more than one legal entity.

**If the organization is modeled as an operating unit**

An operating unit can't have its own ledger information. If your internal organizations do not require unique ledgers, you can model them as operating units. Ledger information will be set up for the parent legal entity in the hierarchy. Income statements can be created for operating units within a legal entity or for the parent legal entity.

**Fiscal calendars****If the organization is modeled as a legal entity**

Each legal entity has its own fiscal calendar. If your internal organizations use different fiscal years and fiscal calendars, you must model the organizations as legal entities.

**If the organization is modeled as an operating unit**

Operating units must share a fiscal calendar. If your internal organizations can use the same fiscal years and fiscal calendars, you can model the organizations as operating units.

**Consolidation****If the organization is modeled as a legal entity**

You must consolidate the financial results for regional offices into a single, consolidated company in order to

prepare financial statements.

**If the organization is modeled as an operating unit**

Consolidation is not required, because data is already shared among operating units.

**Centralized payments**

**If the organization is modeled as a legal entity**

Centralized payments must be set up so that invoices for all child legal entities can be paid to or from a single parent legal entity.

**If the organization is modeled as an operating unit**

Centralized payments are not required because all invoices are recorded in a single legal entity.

**Intercompany transactions**

**If the organization is modeled as a legal entity**

Intercompany sales orders, purchase orders, payments, or receipts can be applied to one another. You are not required to use journal vouchers. You can view intercompany transactions at the sub-ledger level (Accounts receivable, Accounts payable). The following examples illustrate how intercompany transactions are handled.

*Example 1: Headquarters provides services to regional offices and must charge the costs of those services to the regional offices*

If you model the regional office as a legal entity, you have the following options:

- Headquarters creates a journal entry to cross-charge the regional office for the expense. The transactions cannot be aged.
- Headquarters sends a purchase order for the services to the regional office. A sales order is automatically created in the legal entity for the regional office, with intercompany sub-ledger transactions.

*Example 2: Headquarters procures and pays for service that is delivered to a regional office*

If you model the regional office as a legal entity, you have the following options:

- The invoice and payment follow the regulatory requirements of headquarters. Headquarters can create a journal entry to cross-charge the regional office for the expense. The transactions cannot be aged.
- The invoice and payment follow the regulatory requirements of headquarters. Headquarters can create an intercompany sub-ledger transaction.

**If the organization is modeled as an operating unit**

Intercompany transactions among operating units are supported only through journal vouchers. An operating unit cannot issue or receive a purchase order, sales order, or invoice from another operating unit in the same legal entity. You cannot view intercompany transactions at the sub-ledger level (Accounts receivable, Accounts payable). The following examples illustrate how intercompany transactions are handled.

*Example 1: Headquarters provides services to regional offices and must charge the costs of those services to the regional offices*

If you model the regional office as an operating unit, headquarters enters an expense transaction and codes it to the regional office.

*Example 2: Headquarters procures and pays for service that is delivered to a regional office*

If you model the regional office as an operating unit, the invoice and payment follow the regulatory requirements of headquarters. The invoice can be coded to the regional office. On the income statement, use a balancing financial dimension to report costs for the regional office.

**Local tax requirements**

**If the organization is modeled as a legal entity**

A legal entity is subject to the tax laws of the tax authority in the country/region where the legal entity is registered. For example, a legal entity that is registered in Denmark is subject to Danish tax laws and regulations. A legal entity can belong to only one country/region. The country/region that you select for the primary address of the legal entity controls the country/region-specific features that are available to that legal entity. For example, if the primary address of the legal entity is in Denmark, features that are related to Danish tax laws and regulations become available. Therefore, if your organizations are in different countries/regions and require

different local tax options, you must set up the organizations as separate legal entities.

**If the organization is modeled as an operating unit**

Operating units use the country context of the parent legal entity. Operating units in the same legal entity cannot have different country/region-specific requirements. If your organizations are in the same country/region and use the same tax options, you can set them up as operating units.

**Statutory reporting for a country/region**

**If the organization is modeled as a legal entity**

For countries/regions that are supported, most statutory reports can be created. For information about which reports are available for each country/region, see the [Microsoft Dynamics Localization Portal](#). (A CustomerSource logon is required.)

**NOTE**

A posting layer in the general ledger allows you to make adjusting entries to a parent company that uses a different accounting standard than the child company. For example, for a company that uses generally accepted accounting practices in the United Kingdom (UK GAAP), you can make adjusting entries in the posting layer. These entries can be consolidated into a parent company that uses generally accepted accounting principles (GAAP) in the United States. The adjusting entries do not affect UK GAAP reporting.

**If the organization is modeled as an operating unit**

Statutory reports must be created by using another application. You must ensure that data is captured in Finance and Operations apps to support the requirements of each operating unit, where they differ from the requirements of headquarters.

**Currency**

**If the organization is modeled as a legal entity**

If your organizations must use different functional currencies, you must model the organizations as legal entities. Functional currencies are set up per legal entity. However, you can enter transactions in multiple currencies.

**If the organization is modeled as an operating unit**

If your organizations can use a single functional currency, you can model the organizations as operating units. Operating units must share a functional currency. However, you can enter transactions and create reports in multiple currencies.

**Year-end closing**

**If the organization is modeled as a legal entity**

If laws and accounting practices differ among the countries/regions where your organizations are located, you may require different year-end procedures per organization. This means that you must model the organizations as legal entities. Each legal entity has its own year-end procedures.

**If the organization is modeled as an operating unit**

If laws and accounting practices are the same among the countries/regions where your organizations are located, you may use a single set of year-end procedures. This means that you can model the organizations as operating units. All operating units must use the same year-end closing procedure.

**Number sequences**

**If the organization is modeled as a legal entity**

Number sequences for some references can be set up per legal entity. Some number sequences can be shared.

**If the organization is modeled as an operating unit**

Number sequences for some references can be set up per operating unit. Some number sequences can be shared.

## Products

### If the organization is modeled as a legal entity

Product definitions are shared, and they must be released to individual legal entities before they can be included in transactions. Each legal entity has its own set of released products that can be included in transaction documents. If your internal organizations must use different sets of products, you must model the organizations as legal entities.

#### NOTE

Even though product definitions are shared, in each legal entity where a product has been released, you can specify different sales, purchase, and stocking parameters for the item at each inventory site.

### If the organization is modeled as an operating unit

All operating units share the same set of products. If your internal organizations can share the same set of products, you can model the organizations as operating units.

## Inquiry and reporting

### If the organization is modeled as a legal entity

You must manually change companies to enter transactions and perform inquiries in multiple legal entities. Because of data security boundaries, consolidated inquiry and reporting can be resource intensive and time-consuming.

### If the organization is modeled as an operating unit

You do not need to change companies to access data from multiple operating units. Consolidated inquiry and reporting and individual regional inquiry is easier and faster.

## Best practices for modeling organizations and hierarchies

Consider the following best practices when you implement an organization hierarchy:

- Create a department to model the intersection between a legal entity and a business unit. You can then roll up data from a department to a legal entity for statutory reporting, and from a department to a business unit for internal reporting. Departments can serve as profit centers. If you use departments, you do not have to use legal entities and business units as dimensions in the account structure. You can use just departments as a dimension. However, you must use both cost centers and departments as dimensions in the account structure if cost centers are used only as cost accumulators, and departments are used for revenue recognition.
- Model multiple hierarchies for operating units if you have complex requirements for reporting profit and loss.
- In a single legal entity, do not model multiple hierarchies for the same hierarchy purpose.
- Do not create a hierarchy for every purpose. Usually, you can use one hierarchy for multiple purposes. For example, one hierarchy of operating units can be assigned to all policy-related purposes.
- Create balanced hierarchies. In a hierarchy, all nodes that are the same distance from the root node are defined as a level. In a balanced hierarchy, only one type of operating unit can occur at each level, and the distance from the root node to each level is consistent. If there are intermediate levels between a department and a legal entity or a business unit, placeholder organizations may be required to create a balanced hierarchy.
- Do not model a separate hierarchy of operating units if the structure for legal entities is also your operating structure. A mixed hierarchy of legal entities and operating units may serve both purposes.
- Before you model major restructuring scenarios, use the hierarchy's effective dates to perform an impact analysis and a validation test.
- Use draft mode to change a hierarchy before you publish a new version in a production environment.
- Limit the number of people who have permissions to add or remove organizations from a hierarchy in a

production environment. A smaller number reduces the chance that costly mistakes can occur and corrections must be made.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create legal entities

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create legal entities in Microsoft Dynamics 365 Commerce, which must be created and configured before creating channels.

## Overview

A legal entity is an organization that has a registered or legislated legal structure. Legal entities can enter into legal contracts and are required to prepare statements that report on their performance.

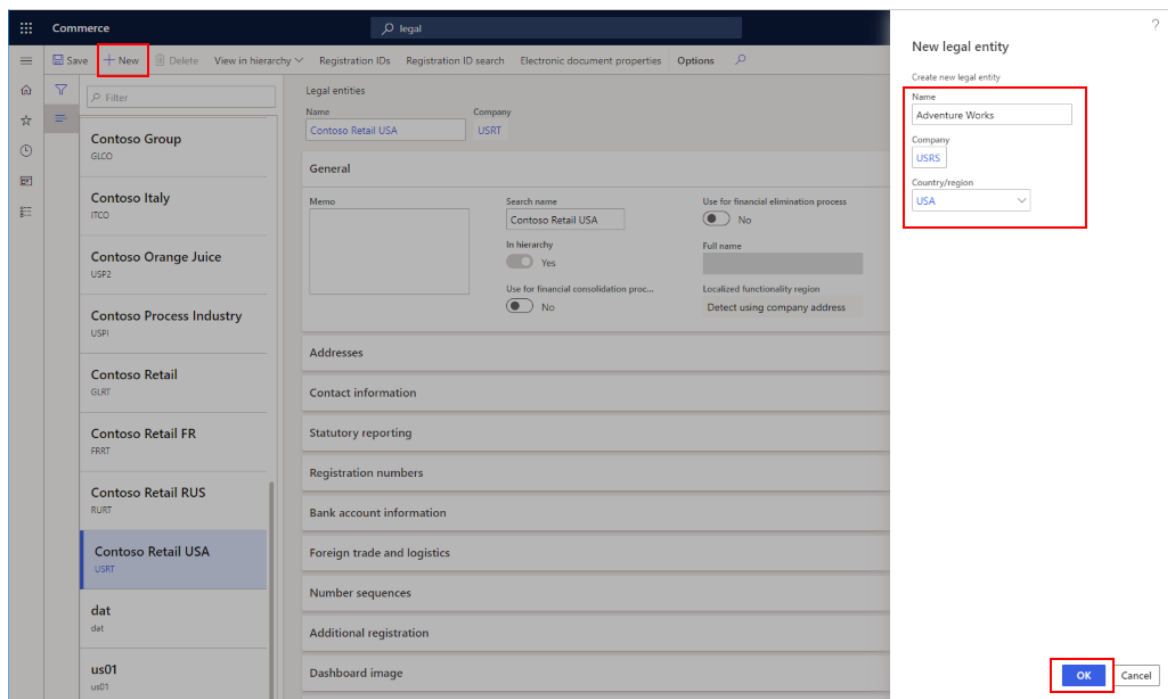
A company is a type of legal entity. Currently, companies are the only kind of legal entity that you can create, and every legal entity is associated with a company ID. This association exists because some functional areas in the program use a company ID, or *DataAreald*, in their data models. In these functional areas, companies are used as a boundary for data security. Users can access data only for the company that they are currently logged on to.

When creating a channel, you must specify which legal entity that channel belongs to.

## Create a new legal entity

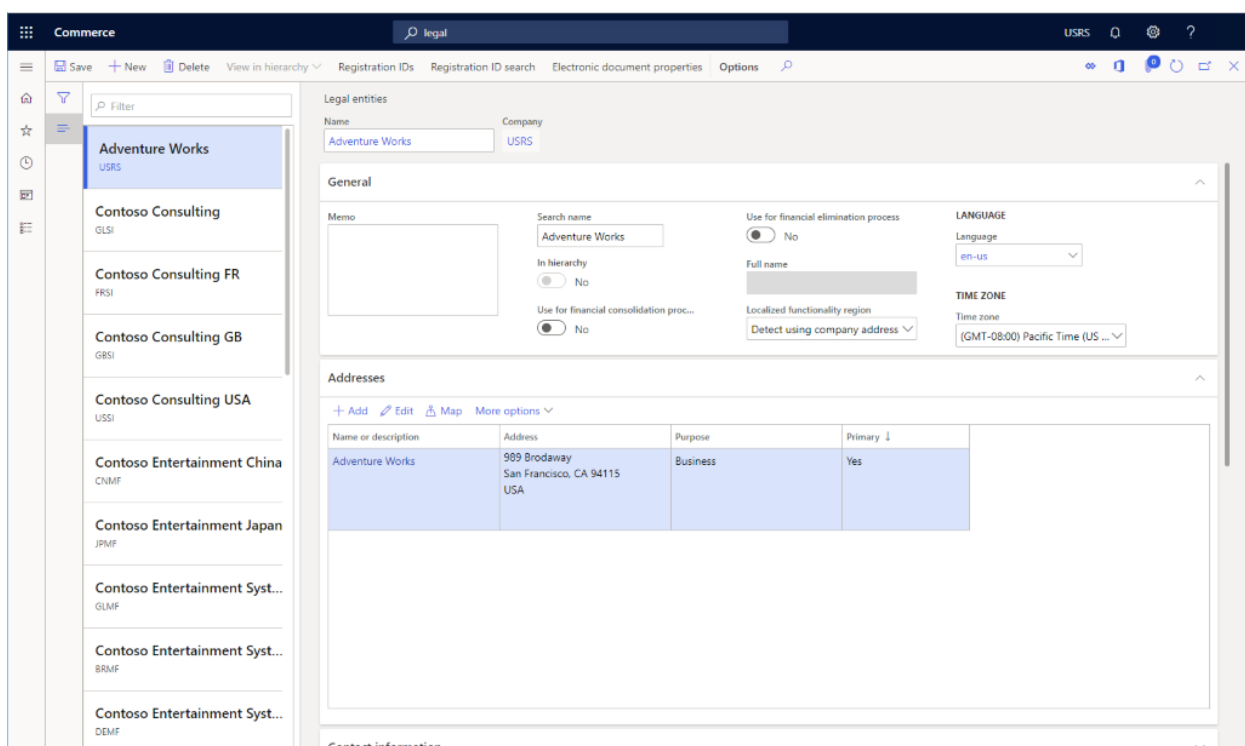
To create a new legal entity in Dynamics 365 Commerce, follow these steps.

1. In the navigation pane, go to **Modules > Headquarters setup > Legal entities**.
2. On the action pane, select **New**. The **New legal entity** pane appears on the right.
3. In the **Name** field, enter a value.
4. In the **Company** field, enter a value.
5. In the **Country/region** field, enter or select a value.
6. Select **OK**.



7. In the **General** section, provide the following general information about the legal entity:
  - a. Enter a search name, if a search name is required. A search name is an alternate name that can be used to search for this legal entity.
  - b. Select whether this legal entity is being used as a consolidation company.
  - c. Select whether this legal entity is being used as an elimination company.
  - d. Select the **default language** for the entity.
  - e. Select the **time zone** for the entity.
8. In the **Addresses** section, select **Edit** to enter address information, such as the street name and number, postal code, and city.
9. In the **Contact information** section, enter information about methods of communication, such as email addresses, URLs, and telephone numbers.
10. In the **Statutory reporting** section, enter the registration numbers that are used for statutory reporting.
11. In the **Registration numbers** section, enter any information required by the legal entity.
12. In the **Bank account information** section, enter bank accounts and routing numbers for the legal entity.
13. In the **Foreign trade and logistics** section, enter shipping information for the legal entity.
14. In the **Number sequences** section, you can view the number sequences that are associated with the legal entity. This will be empty to start with.
15. In the **Dashboard image** section, view or change the logo and dashboard image associated with the legal entity.
16. In the **Tax registration** section, enter the registration numbers that are used to report to tax authorities.
17. In the **Tax 1099** section, enter 1099 information for the legal entity.
18. In the **Tax information** section, enter tax information for the legal entity.
19. Select **Save**.

The following image shows details of an example legal entity.





# Additional resources

[Organizations and organizational hierarchies overview](#)

[Plan your organizational hierarchy](#)

[Organization hierarchies](#)

[Channels overview](#)

[Channel setup prerequisites](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up organization hierarchies

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to set up organization hierarchies in Microsoft Dynamics 365 Commerce.

## Overview

Before creating channels, you'll want to ensure you have set up your organization hierarchies.

You can use organization hierarchies to view and report on your business from various perspectives. For example, you can set up one hierarchy for tax, legal, or statutory reporting. You can then set up another hierarchy to report financial information that is not legally required, but that is used for internal reporting.

Before you create an organization hierarchy, you must create organizations. For more information, see [Create legal entities](#) or [Create operating units](#).

For more information, see the following topics.

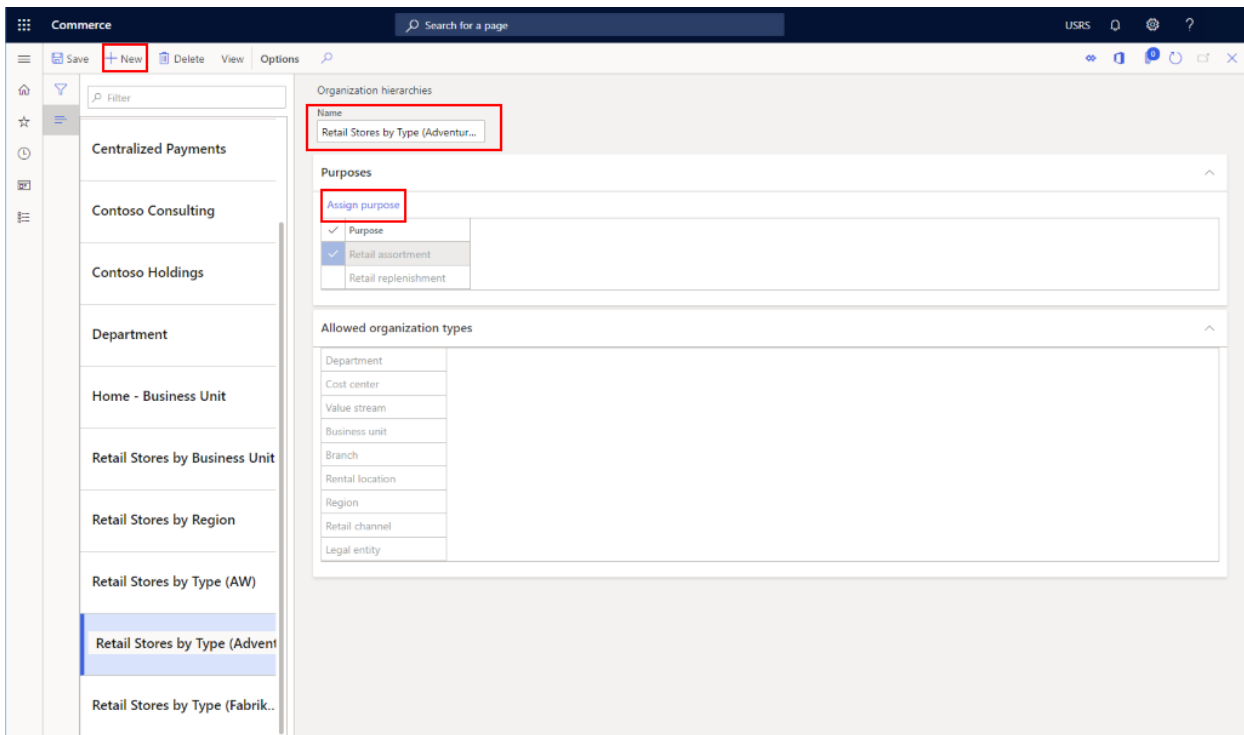
- [Organizations and organizational hierarchies overview](#)
- [Plan your organization hierarchy](#)
- [Create an organization hierarchy](#)

## Create an organizational hierarchy

To create an organizational hierarchy, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Channel Setup > Organization hierarchies**.
2. On the action pane, select **New**.
3. In the **Name** field, enter a value.
4. In the **Purpose** section, select **Assign purpose**.
5. In the list, find and select the desired record. Select a purpose to assign to your organization hierarchy.
6. In the **Assigned hierarchies** section, select **Add**.
7. In the list, mark the selected row. Find the hierarchy you just created.
8. Select **OK**.

The following image shows an example organizational hierarchy created for a fictitious "Adventure Works" set of stores.

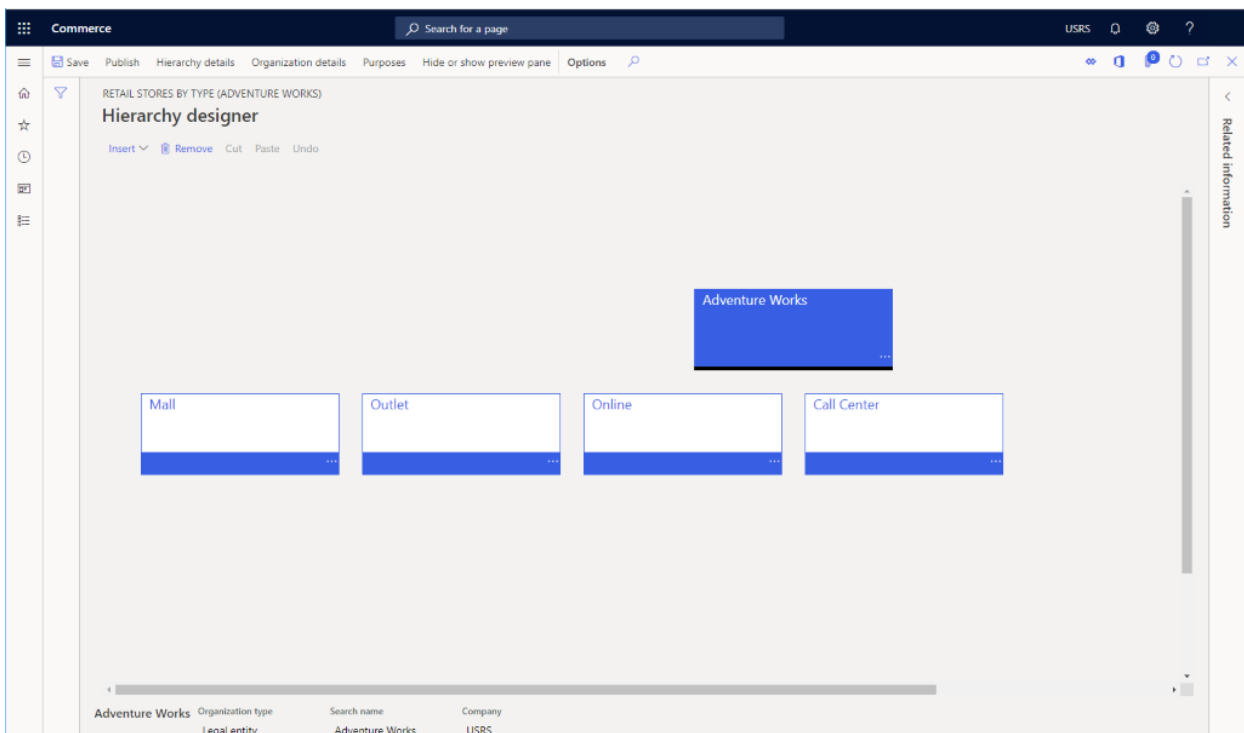


### Add organizations to a hierarchy

To add organizations to a hierarchy, follow these steps.

1. In the list, find and select the desired record. Select your hierarchy.
2. On the action pane, select **View**.
3. Add organizations, as necessary.
4. To add an organization, select **Edit** and then select **Insert**. When you are done making changes you can save a draft and publish the changes.

The following image shows a legal entity added at the hierarchy root with four cost centers added for "Mall", "Outlet", "Online" and "Call Center" channels. Various retail, call center and online channels can then be added to each.



# Additional resources

[Organizations and organizational hierarchies overview](#)

[Plan your organizational hierarchy](#)

[Create legal entities](#)

[Create operating units](#)

[Channels overview](#)

[Channel setup prerequisites](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create an operating unit

2/18/2021 • 2 minutes to read • [Edit Online](#)

An operating unit is an organization that is used to divide the control of economic resources and operational processes in a business. People in an operating unit have a duty to maximize the use of scarce resources, improve processes, and account for their performance. The types of operating units include cost centers, business units, departments, and value streams. Use the following procedure to create an operating unit. The demo data company used to create this procedure is USMF.

1. Go to **Navigation pane > Modules > Organization administration > Organizations > Operating units**.
2. Click **New** to open the drop dialog.
3. In the list, find and select the desired record. Select the type of operating unit you want to create.
4. In the list, click the link in the selected row.
5. In the **Name** field, type a value.
  - Expand the **General** section, if necessary.
  - Provide general information about the operating unit, such as an identification number, DUNS number, and manager.
  - Expand the **Addresses** section, if necessary.
  - Enter address information, such as the street name and number, postal code, and city. Click **Add** to enter a new address record, or click **Edit** to modify an existing address record.
  - Expand the **Contact information** section, if necessary.
  - Enter information about methods of communication, such as email addresses, URLs, and telephone numbers. To enter a new communication record, click **New**. To modify an existing communication record, click **More options > Advanced**.
6. Optionally, change the **Operating unit number** as needed. Note that this number is a unique identifier for the corresponding **Party** record and cannot be the same as any other operating unit.
7. Select **Save**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Design the relationships between organizational units

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through how to design the relationship between organizational units. You must create a new organization purpose before defining the relationship, or you can use the existing organization purpose. The demo data company used to complete this procedure is USRT. This task is intended for the administrator role.

1. Go to Organization administration > Organizations > Organization hierarchies.
2. Click New.
3. In the Name field, type a value.
4. Click Assign purpose.
5. In the list, find and select the desired record.
6. Click Add.
7. In the list, find and select the desired record.
8. Click OK.
  - You can select as many organization purposes as required for your organization.
9. In the list, find and select the desired record.
10. Click Set as default.
11. Close the page.
12. Click Save.
13. Click View.
14. Click Edit.
15. Click Insert.
16. Click Business unit.
17. In the list, find and select the desired record.
18. In the list, click the link in the selected row.
19. Click Insert.
20. Click Commerce channel.
21. In the list, find and select the desired record.
22. In the list, click the link in the selected row.
  - You can add as many organization units as is required.
23. Click Save.
24. Click Close.
25. Click Publish to open the drop dialog.
26. In the Effective date field, enter a date and time.
27. In the Effective date field, enter a date and time.
28. In the Describe changes field, type a value.
29. Click Publish.
30. Click Close.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Warehouse set up

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to set up a warehouse to be used with a new channel in Microsoft Dynamics 365 Commerce.

## Overview

Each Commerce channel requires a configured warehouse to be associated with it. The following procedures provide the minimum configuration required to set up a warehouse for a Commerce channel. For more information regarding warehouse setup, please see the [Warehouse management overview](#).

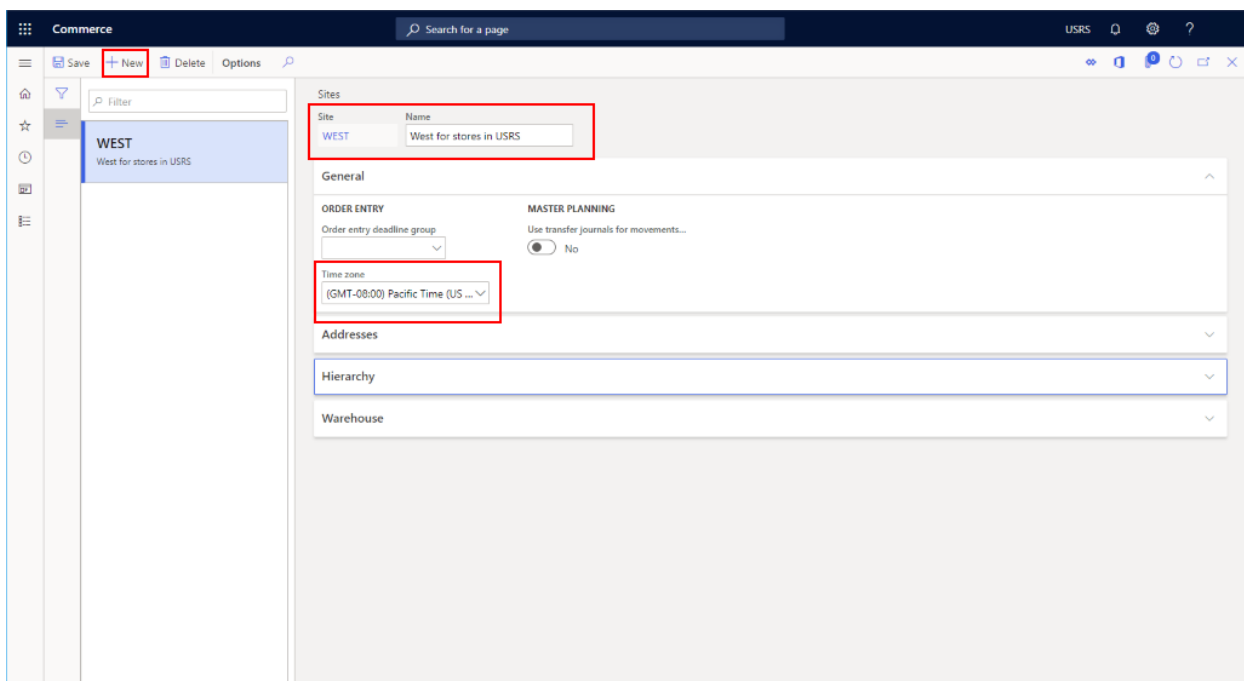
## Configure a warehouse site

Before setting up a warehouse, you need to configure a warehouse site.

To configure a warehouse site, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Channel setup > Sites**.
2. On the action pane, select **New**.
3. In the **Site** field, enter a value.
4. In the **Name** field, enter a value.
5. In the **General** section, set the appropriate **Time zone**.
6. In the **Addresses** section, enter an address.
7. On the action pane, select **Save**.

The following image shows an example warehouse site.



## Set up a warehouse

To set up a warehouse, follow these steps.



1. In the navigation pane, go to **Modules > Retail and commerce > Channel setup > Warehouses**.
2. On the action pane, select **New**.
3. In the **Warehouse** field, enter a value. If this is a 1:1 mapping to a store, consider using the store name or the name of a regional distribution center.
4. In the **Name** field, enter a value.
5. In the **Site** drop-down list, select the warehouse site created previously.
6. In the **Type** field, select **Default**.
  - If you want to set a **Quarantine warehouse**, first you'll need to follow these steps to create an additional warehouse where the **Type** is set to **Quarantine**.
  - If you want to set a **Transit warehouse**, first you'll need to follow these steps to create an additional warehouse where the **Type** is set to **Transit**.
7. On the action pane, select **Save**.

## Set up inventory aisles

To set up inventory aisles, follow these steps.

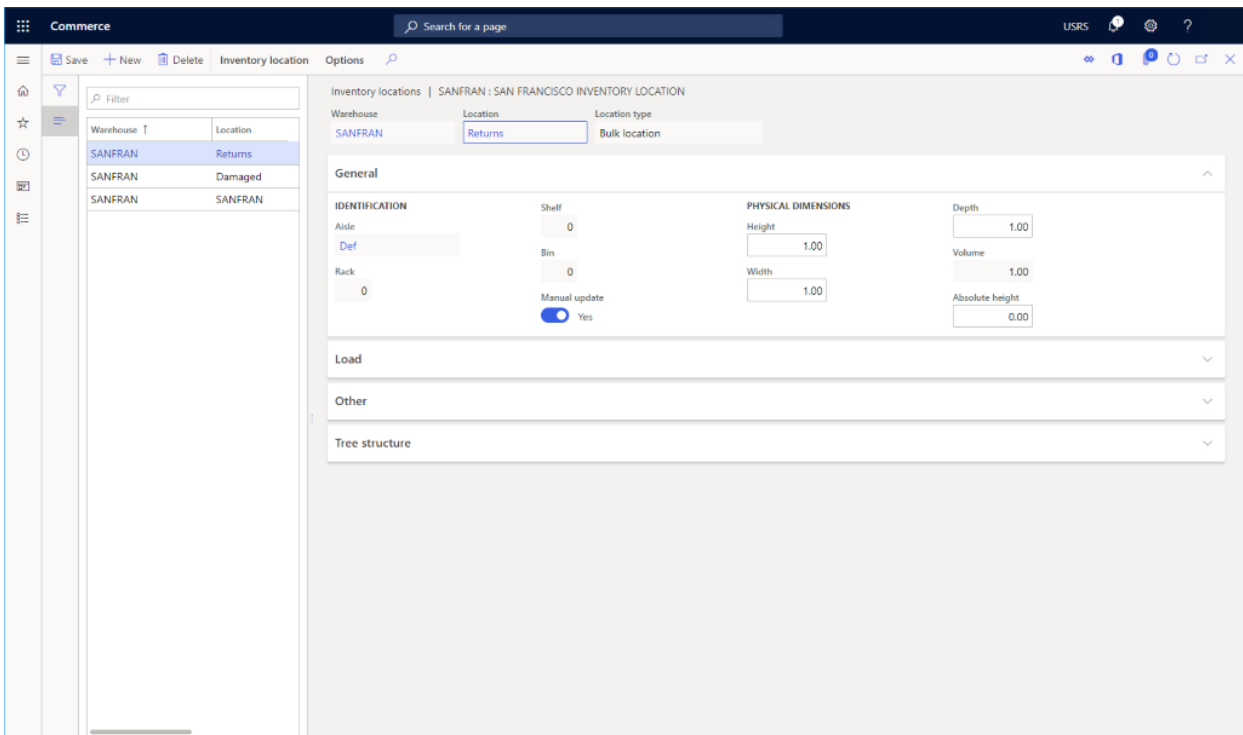
1. In the navigation pane, go to **Modules > Retail and commerce > Channel setup > Location setup > Inventory aisles**.
2. On the action pane, select **New**.
3. In the **Warehouse** drop-down list, select the warehouse created previously.
4. In the **Aisle** field, enter a name (for example, "Def").
5. In the **Name** field, enter a name (for example, "Default aisle").
6. On the action pane, select **Save**.

## Set up warehouse inventory locations

To set up warehouse inventory locations for standard, damaged, and returned inventory, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Channel setup > Warehouses**.
2. Select the warehouse you created previously.
3. On the action pane, select **Edit**.
4. On the action pane, select **Warehouse**, and then select **Inventory location**.
5. On the action pane, select **New**. The **Warehouse** drop-down list should default to your new warehouse.
  - a. In the **Aisle** box, enter the name of the aisle you specified earlier.
  - b. Set **Manual update** to **Yes**
  - c. In the **Location** box, enter the name of the warehouse.
  - d. On the action pane, select **Save**.
6. On the action pane, select **New**. The **Warehouse** drop-down list should default to your new warehouse.
  - a. In the **Aisle** box, enter the name of the aisle you specified earlier.
  - b. Set **Manual update** to **Yes**
  - c. In the **Location** box, enter "Damaged".
  - d. On the action pane, select **Save**.
7. On the action pane, select **New**. The **Warehouse** drop-down list should default to your new warehouse.
  - a. In the **Aisle** box, enter the name of the aisle you specified earlier.
  - b. Set **Manual update** to **Yes**
  - c. In the **Location** box, enter "Returns".
  - d. On the action pane, select **Save**.

The following image shows a San Francisco warehouse inventory location setup.

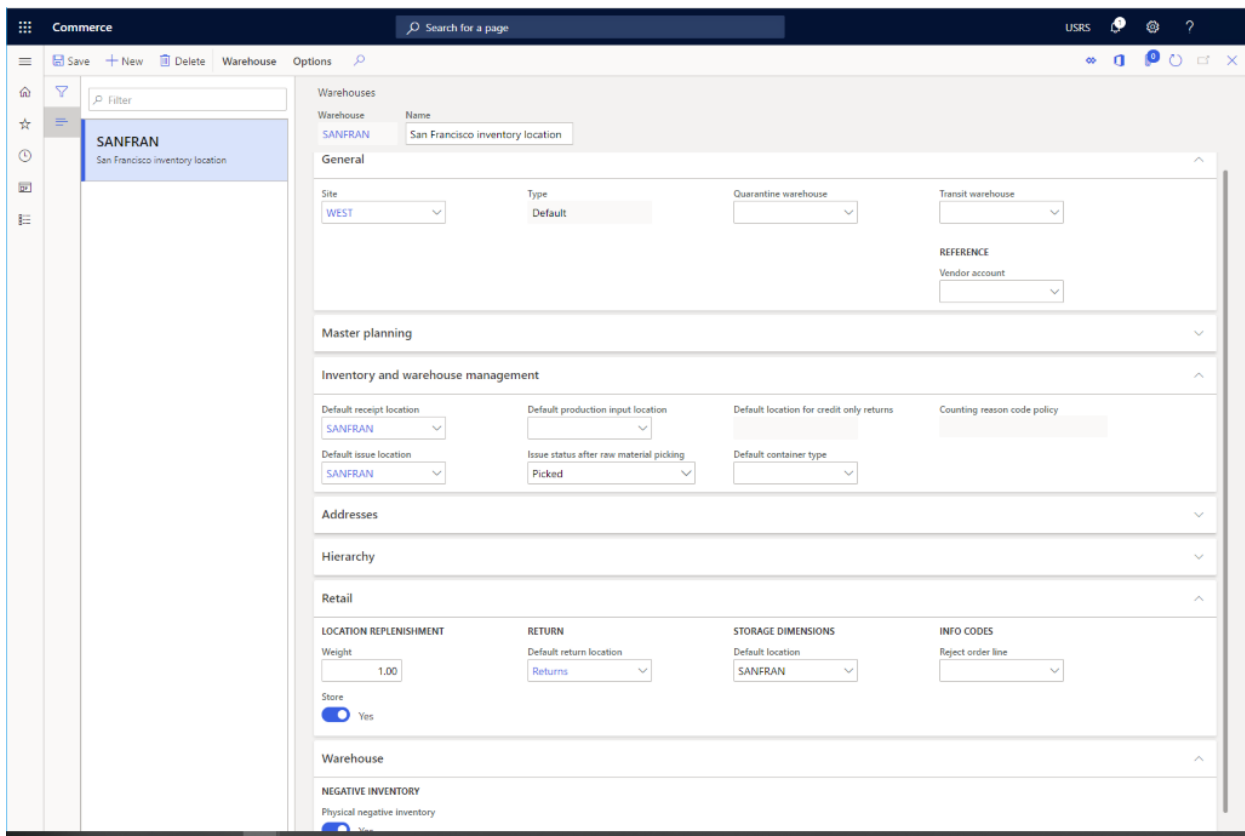


## Complete warehouse setup

To complete warehouse setup, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Channel setup > Warehouses**.
2. Select the warehouse you previously created.
3. On the action pane, select **Edit**.
4. Under **Inventory and warehouse management**:
  - a. Set **Default receipt location** to the default location created above.
  - b. Select **Default issue location** to the default location created above.
5. Under the **Addresses** section, enter a warehouse address.
6. Under the **Retail** section:
  - a. In the **Default return location** box, enter the returns location created previously.
  - b. Set **Store** to **Yes**.
  - c. Set **Weight** to **1.00**.
  - d. In the **Storage Dimensions** box, enter the default location created previously.
7. Under the **Warehouse** section, set **Physical negative inventory** to **Yes**.
8. On the action pane, select **Save**.

The following image shows details for a configured warehouse.



## Additional resources

[Warehouse management overview](#)

[Channels overview](#)

[Channel setup prerequisites](#)

[Set up a retail channel](#)

[Set up an online channel](#)

[Set up a call center channel](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Sales tax overview

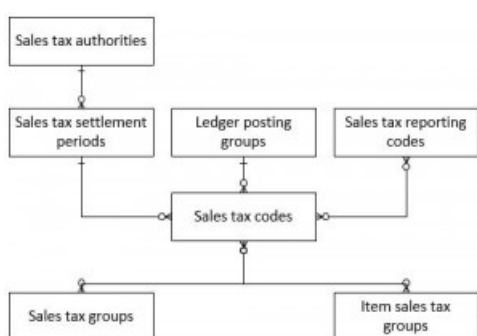
2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic provides an overview of the sales tax system. It explains the elements of the sales tax setup and how they work together.

## Overview

The sales tax framework supports many types of indirect taxes, such as sales tax, value-added tax (VAT), goods and services tax (GST), unit-based fees, and withholding tax. These taxes are calculated and documented during purchase and sales transactions. Periodically, they must be reported and paid to tax authorities.

The following diagram shows the entities of the tax setup and how they are related.



For every sales tax that a company must account for, a sales tax code must be defined. A sales tax code stores the tax rates and calculation rules for the sales tax.

Every sales tax code must be linked to a sales tax settlement period. Sales tax settlement periods define the intervals at which sales tax must be reported and paid to the sales tax authority. Every sales tax settlement period must be assigned to a sales tax authority. A sales tax authority represents the entity that sales tax is reported and paid to. It also defines the layout for the sales tax report. Sales tax authorities can be related to vendor accounts. For more information, see [Set up sales tax settlement periods](#).

Every sales tax code must also be linked to a ledger posting group. A ledger posting group specifies the main accounts that amounts for the sales tax codes will be posted to.

Optional sales tax reporting codes can also be defined. These can be assigned on sales tax codes for the various amount types that are calculated for the sales tax code. The **Sales tax payment by code** report shows totals per sales tax reporting code for a given sales tax settlement period and interval.

Every transaction that sales tax needs to be calculated and posted for must have a sales tax group and an item sales tax group. Sales tax groups are related to the party (for example, customer or vendor) of the transaction, whereas item sales tax groups are related to the resource (for example, item or procurement category) of the transaction. Tax groups contain a list of tax codes. The tax codes that are present in both the sales tax group and item sales tax group for a transaction are the tax code that apply to that transaction.

The following table describes the entities and the sequence for the tax setup.

SETUP ACTIVITY	REQUIRED/OPTIONAL AND DESCRIPTION
Create main accounts.	Required. Before you can set up the sales tax functionality, the main accounts that the company uses to pay and record taxes must be created.

SETUP ACTIVITY	REQUIRED/OPTIONAL AND DESCRIPTION
Set up ledger posting groups for sales tax.	Required. Ledger posting groups define the main accounts for recording and paying sales taxes. For more information, see <a href="#">Set up Ledger posting groups for sales tax</a> .
Set up sales tax authorities.	Required. Sales tax authorities are the entities that tax must be reported and paid to. For more information, see <a href="#">Set up sales tax authorities</a> .
Set up sales tax settlement periods.	Required. Sales tax settlement periods contain information about when and how often sales tax must be reported and paid. They are related to a sales tax authority.
Set up sales tax reporting codes.	Optional. Sales tax reporting codes can be assigned to sales tax codes to report amounts for multiple sales tax codes under one sales tax reporting code. For more information, see <a href="#">Set up sales tax reporting codes</a> .
Set up sales tax codes.	Required. Sales tax codes contain the tax rates and calculation rules for each sales tax. Sales tax codes are related to a sales tax settlement period and a ledger posting group. For more information, see <a href="#">Set up sales tax codes</a> .
Set up sales tax groups.	Required. Sales tax groups contain a list of sales codes that apply for the party (customer or vendor) of a transaction. For a given transaction, the intersection of sales tax codes in the sales tax group and the item sales tax group determines the sales tax codes that apply to that transaction.
Set up item sales tax groups.	Required. Item sales tax groups contain a list of sales codes that apply for the resource (product, service, and so on) of a transaction. For a given transaction, the intersection of sales tax codes in the sales tax group and the item sales tax group determines the sales tax codes that apply to that transaction. For more information, see <a href="#">Set up sales tax groups and item sales tax groups</a> .
Set up sales tax parameters on the application parameter pages.	Required. Different areas, such as General ledger, Accounts receivable, and Accounts payable, must set up parameters for correct calculation of indirect taxes. Although most of these parameters have default values, they must be modified to fit each company's requirements.

## Sales tax on transactions

On every transaction (sales/purchase document lines, journals, and so on), you must enter a sales tax group and an item sales tax group to calculate sales tax. Default groups are specified in master data (for example, customer, vendor, item, and procurement category), but you can manually change the groups on a transaction if you must. Both groups contain a list of sales tax codes, and the intersection of the two lists of sales tax codes determines the list of applicable sales tax codes for the transaction.

On every transaction, you can look up the calculated sales tax by opening the **Sales tax transaction** page. You can look up the sales tax for a document line or for the whole document. For certain documents (for example, vendor invoice and general journals), you can adjust the calculated sales tax if the original document shows deviant amounts.

# Sales tax settlement and reporting

Sales tax must be reported and paid to tax authorities at regulated intervals (monthly, quarterly, and so on). You can settle tax accounts for the interval and offset the balances to the tax settlement account, as specified in the ledger posting groups. You can access this functionality on the **Settle and post sales tax** page. You must specify the sales tax settlement period that sales tax should be settled for.

After the sales tax has been paid, the balance on the sales tax settlement account should be balanced against the bank account. If the sales tax authority that is specified on the sales tax settlement period is related to a vendor account, the sales tax balance is posted as an open vendor invoice and can be included in the regular payment proposal.

## Conditional sales tax

Conditional sales tax is a sales tax that is paid proportionally to the actual amount that is paid on an invoice. Conversely, standard sales tax is calculated at invoicing time. Conditional sales tax must be paid to the sales tax authority when the payment is posted, not when the invoice is posted. When the invoice is posted, the transaction must be reported on the sales tax book report. However, the transaction must be excluded from the sales tax payment report.

If you select the Conditional sales tax check box in the General ledger parameters form, no sales tax can be deducted until you have paid the invoice. This is a legal requirement in some countries/regions.

### NOTE

When you select the Conditional sales tax check box, you must set up sales tax codes and sales tax groups, and also create ledger posting groups, to support the functionality. |

### Example

You settle sales taxes each month. On June 15, you create a customer invoice of 10,000, plus sales tax.

- The sales tax is 25 percent, or 2,500.
- The invoice payment is due July 30.

You typically would have to settle and pay 2,500 to the tax authority when the invoice is posted in June, even though you have not received the payment from the customer.

However, if you are using a conditional sales tax, you settle with the tax authority when you receive the payment from the customer on July 30.

### Postdated check

If you use postdated check as the payment method, when the payment is created, the bank account isn't cleared. In some countries, the VAT becomes 'realized' liability when the payment clears the bank, which means the postdated check is settled. You can enable it by selecting **Realize the conditional tax when postdated checks are drawn** in **Cash and bank management > Setup > Cash and bank management parameters > Postdated checks**.

For more information, see [Set up withholding tax](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Sales tax calculation methods in the Origin field

2/18/2021 • 6 minutes to read • [Edit Online](#)

This article explains the options in the Origin field on the sales tax codes page and how sales tax is calculated based on the selected option for a sales tax code.

For each sales tax code that you create in the Sales tax codes page, you must select the method of calculation to apply to the tax base amount in the Origin field.

## Percentage of net amount

The Percentage of net amount calculation method is the default value in the Origin field. The sales tax is calculated as a percentage of the purchase or sales amount, excluding any other sales taxes.

### Example

The tax rate is 25%. The invoice line shows a quantity of 10 items at 1.00 each, and the customer is allowed a 10% line discount. Net amount:  $(10 \times 1.00) - 10\% = 9.00$  Sales tax:  $9.00 \times 25\% = 2.25$  Total amount:  $9.00 + 2.25 = 11.25$

## Percentage of gross amount

If you select the Percentage of gross amount method, the sales tax is calculated as a percentage of the gross sales amount. The gross amount is the line net amount plus all taxes and fees for the line except the one tax with Origin = Percentage of gross amount.

### Example

The tax authority has imposed special duties on an item. The duty amounts are added to the net amount before sales tax is calculated. Given the following sales tax codes:

- DUTY 1 = 10%, using the Percentage of net amount calculation method
- DUTY 2 = 20%, using the Percentage of net amount calculation method
- SALESTAX = 25%, using the Percentage of gross amount calculation method

If the net amount is 10.00, then DUTY 1 is 1.00 ( $10.00 \times 10\%$ ) and DUTY 2 = 2.00 ( $10.00 \times 20\%$ ). The amounts would be as follows: Gross amount: Net amount + DUTY 1 amount + DUTY 2 amount ( $10.00 + 1.00 + 2.00$ ) = 13.00 SALESTAX =  $13.00 \times 25\% = 3.25$  Total DUTIES and SALESTAX:  $1.00 + 2.00 + 3.25 = 6.25$  Total amount:  $10.00 + 6.25 = 16.25$

#### NOTE

Only one tax code with Origin = Percentage of gross amount can be used for a transaction. If more than one such tax code is determined for a transaction an error will be displayed that sales tax cannot be calculated.

## Percentage of sales tax

When you select Percentage of sales tax in the Origin field, sales tax is calculated as a percentage of the sales tax that is selected in the Sales tax on sales tax field. The sales tax that is selected in the Sales tax on sales tax field is calculated first. The second sales tax is then calculated based on the first sales tax amount.

### Example

Given the following sales tax codes:

- DUTY 1 = 10%, using the Percentage of net amount method
- DUTY 2 = 20%, using the Percentage of sales tax method, with Duty 1 in the Sales tax on sales tax field
- SALESTAX = 25%, using the Percentage of gross amount method

Net amount: 10.00 DUTY 1:  $10.00 \times 10\% = 1.00$  DUTY 2:  $1.00 \times 20\% = 0.20$  Gross amount:  $10.00 + 1.00 + 0.20 = 11.20$  SALESTAX:  $11.20 \times 25\% = 2.80$  Total DUTIES and SALESTAX:  $1.00 + 0.20 + 2.80 = 4.00$  Total amount:  $10.00 + 4.00 = 14.00$

#### NOTE

Multilevel tax on tax calculations are not possible. A tax cannot be calculated based on a tax which already is calculated based on another tax. Multiple single level tax on tax codes can be calculated on a transaction.

## Amount per unit

When you select Amount per unit in the Origin field, sales tax is calculated as a fixed amount per unit multiplied with the quantity entered on the document line. A unit has to be selected in the Unit field. The amount per unit is specified in the Sales tax code values page.

### Example

Sales tax code is set up as: USD 1.20 per unit = box On a sales invoice line 25 boxes of an item are sold Sales tax is calculated as  $25 \times 1.20 = 30.00$

#### NOTE

If the transaction is entered in different unit than the unit specified on the sales tax code, it is converted automatically based on the unit conversions that are set up in the Unit conversions page.

### Amount per unit, additional option

On the Calculation tab, you can select whether an amount per unit calculated tax is calculated before other tax codes and added to the net amount before other tax codes with Origin = Percentage of net amount are calculated.

### Examples

Assume we calculate 2 tax codes on a transaction:

- DUTY: Origin = Amount per unit and a sales tax, the value is set to 5.00 per unit = pcs
- SALESTAX: Origin = as shown in the examples below, the value is set to 25%

We sell 1 piece of an item at a unit price of 10.00

#### Example 1

SALESTAX: Origin = Percentage of gross amount method The Calculate before sales tax option has no effect, because SALESTAX is calculated as a percentage of the gross amount. DUTY:  $1 \times 5.00 = 5.00$  Gross amount:  $10.00 + 5.00 = 15.00$  SALESTAX:  $15.00 \times 25\% = 3.75$  Total sales tax:  $5.00 + 3.75 = 8.75$  Total amount:  $10.00 + 8.75 = 18.75$

#### Example 2

SALESTAX: Origin = Percentage of net amount The Calculate before sales tax option is not selected for the DUTY calculation. Net amount: 10.00 DUTY:  $1 \times 5.00 = 5.00$  SALESTAX:  $10.00 \times 25\% = 2.50$  Total sales tax:  $5.00 + 2.50 = 7.50$  Total amount:  $10.00 + 7.50 = 17.50$

#### Example 3

SALESTAX: Origin = Percentage of net amount The Calculate before sales tax option is selected for the DUTY calculation. Net amount: 10.00 DUTY:  $1 \times 5.00 = 5.00$  SALESTAX:  $(10.00 + 5.00) \times 25\% = 3.75$  Total sales tax: 5.00



+ 3.75 = 8.75 Total amount: 10.00 + 8.75 = 18.75

#### Example 4

The result of Example 3 and Example 1 is the same, because there is only one duty. Assume that you have two DUTIES, and only one of them is included in the net amount for the sales tax calculation: DUTY 1: 5.00, using the Amount per unit method, and the Calculate before sales tax option is selected DUTY 2: 2.50, using the Amount per unit method, and the Calculate before sales tax option is not selected Sales tax: 25%, using the Percentage of net amount method Net amount: 10.00 DUTY 1:  $1 \times 5.00 = 5.00$  DUTY 2:  $1 \times 2.50 = 2.50$  Net amount subject to sales tax:  $10.00 + 5.00 = 15.00$  SALESTAX:  $15.00 \times 25\% = 3.75$  Total sales taxes, including duties:  $5.00 + 2.50 + 3.75 = 11.25$  Total amount:  $10.00 + 11.25 = 21.25$  The 25% SALESTAX is calculated for the sum of the net amount (10.00) + DUTY 1 (5.00) = 15.00. DUTY 2 is added to the tax amount after the sales tax is calculated.

## Calculated percentage of net amount

The Calculated percentage of net amount handles tax calculation differently depending on the setting of the Amounts include sales tax parameter for the document or journal.

#### Example 1

Document / journal is set to Amounts include sales tax = Yes Transaction line amount: 10.00 Tax rate: 25% Sales tax: Transaction line amount x tax rate ( $10.00 \times 25\%$ ) = 2.50 Tax base amount (origin amount): Transaction line amount - Sales tax ( $10.00 - 2.50$ ) = 7.50

#### Example 2

Document / journal is set to Amounts include sales tax = No Transaction line amount: 10.00 Tax rate: 25% Sales tax:  $(\text{Transaction line amount} \times \text{tax rate}) / (100 - \text{tax rate})$  ( $10.00 \times 25\% / (100\% - 25\%) = 3.33$  Tax base amount (origin amount): Transaction line amount = 10.00

## Additional resources

[Sales tax rates based on the Marginal base and Calculation methods](#)

[Whole amount and Interval calculation options for sales tax codes](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Sales tax assignment and overrides

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates how to assign sales tax groups to commerce channels. It also walks through the process of creating a new sales tax override and assigning it to an existing sales tax override group. This procedure uses the USRT company in demo data.

1. Go to Retail and Commerce > Channels > Stores > All stores.
2. In the list, click the Channel ID link for "Houston."
3. Click Edit.
  - The "Sales tax group" field contains the list of sales tax groups for the current company. The currently assigned group is a generic "Texas" sales tax group. There are also sales tax groups for "Washington" and "Washington, King County." Sales tax groups can include applicable taxes for multiple municipalities.
  - The "Sales tax override" field is where sales tax override groups can be mapped to the channel. Sales tax override groups can be used to group together sales tax overrides that work for multiple stores. Rather than manually assigning sales tax overrides one by one, the group can be created and assigned directly to the channels to save time.
4. Click Save.
5. Close the page.
6. Go to Retail and Commerce > Channel setup > Sales taxes > Sales tax overrides.
7. Click New.
8. In the Sales tax override field, provide a name for your new override.
9. In the Description field, provide a description of the override.
10. Set the status to "Enable."
11. Expand or collapse the Override section.
12. In the Type field, select an option.
  - Item sales tax groups can be used to override taxes for specific items that belong to the group. For example, food items are typically taxed differently from hard goods, and would likely have their own sales tax group. Sales tax groups are groups of taxes that are applicable to a particular channel. For example, if a channel sells both retail and business-to-business, different items sales tax groups may be used. All the applicable taxes would be mapped to the sales tax group.
  - Now you can select the "From" and "To" taxes or "From tax group" and "To tax group" to create your sales tax override. The "From" field indicates the tax or tax group to be overridden. Overriding by Item sales tax group provides different options than overriding by sales tax group. Sales tax overrides can be set up to override taxes on entire transactions or on particular lines in the transaction.
13. Click Save.
14. Close the page.
15. Go to Retail and Commerce > Channel setup > Sales taxes > Sales tax override groups.
  - In this step you will assigned the newly created sales tax override to the sales tax override group assigned to the Houston channel.
16. Click Edit.
17. Expand or collapse the Setup section.
18. Click Add.
19. In the Sales tax override field, click the drop-down button to open the lookup.
20. Select the previously created sales tax override from the list.

21. In the list, click the link in the selected row.

22. Click Save.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Whole amount and Interval calculation options for sales tax codes

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article explains the options for the Calculation method field on sales tax codes and how sales tax is calculated for intervals and whole amounts.

You can set up a sales tax code to be calculated based on a whole amount or an interval amount. In the Sales tax codes page, use the Calculation method field on the Calculation FastTab to select how to calculate a sales tax code.

- Whole amount – The tax rate is applied to the whole taxable amount.
- Interval – The taxable amount is divided into parts, each of which falls in a range that has a specific sales tax rate. The part of the amount that falls in a given interval is taxed according to the tax rate for that interval. The sales tax is the sum of the tax amounts that are calculated for each amount interval.

## NOTE

The Interval option is available only when you select Line in the Calculation method field in the Sales tax area of the General ledger parameters page.

Intervals are set up in the Sales tax code values page by entering Minimum and Maximum limit amounts per tax rate. For taxes to be calculated on all taxable amounts, regardless of which calculation method is selected, intervals must follow these rules:

- The first interval must have a Minimum limit of zero.
- The last interval must have a Maximum limit of zero, which indicates infinity.
- The Maximum limit of an interval must be the Minimum limit of the next interval.

If an amount is the Maximum limit of the previous interval and the Minimum limit of the next interval, the sales tax rate of the first interval will be applied to the amount. If an amount falls outside the intervals that are defined by upper and lower limits, a sales tax rate of zero will be applied.

## Example: Whole amount method of calculation

In the Sales tax code values page, sales tax rates are set up in the following intervals:

Minimum limit	Maximum limit	Tax rate
0.00	50.00	30%
50.00	100.00	20%
100.00	0.00	10%

The sales tax is calculated on the whole taxable amount.

TAXABLE AMOUNT (PRICE)	CALCULATION	SALES TAX
35.00	$35.00 * 0.30$	10.50
50.00	$50.00 * 0.30$	15.00
85.00	$85.00 * 0.20$	17.00
305.00	$305.00 * 0.10$	30.50

## Example: Interval method of calculation

In the Values page, sales tax rates are set up in the following intervals:

Minimum limit	Maximum limit	Tax rate
0.00	50.00	30%
50.00	100.00	20%
100.00	0.00	10%

The sales tax is the sum of the tax amounts that are calculated for each amount interval.

TAXABLE AMOUNT (PRICE)	CALCULATION	SALES TAX
35.00	$35.00 * 0.30$	10.50
50.00	$50.00 * 0.30$	15.00
85.00	$(50.00 * 0.30 = 15.00) + (35.00 * 0.20 = 7.00)$	22.00
305.00	$(50.00 * 0.30 = 15.00) + (50.00 * 0.20 = 10.00) + (205 * 0.10 = 20.50)$	45.50

For more information, see [Sales tax rates based on the Marginal base and Calculation methods](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Calculation of tax exemption

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes functionality for tax exemption calculations in the point of sale (POS) and call center.

## Key terms

TERM	DESCRIPTION
B2B	An abbreviation for "business to business." It's used to indicate sales between businesses, as opposed to sales between a retailer and an individual.
VAT	Value-added tax that is included in the price of a product.

## Adjust prices for tax exemptions when the price includes tax

Microsoft Dynamics 365 Commerce version 10.0.13 and later includes a feature called **Enable tax exemption for the 'price includes sales tax' scenario**. When this feature is enabled, an option called **Calculate price inclusive tax exempt** appears on the **General** FastTab for store and call center settings. If this option is set to **Yes**, prices in tax-inclusive scenarios are adjusted when the transaction or specific taxes in the transaction should be exempted. When store-based taxes are used, you can apply these exemptions by using tax overrides. When customer-based taxes are used at the store, the exemptions are automatically applied based on the customer's tax settings.

This setting is also supported for orders that are created in the call center and stores.

The screenshot shows the 'Stores' configuration page for 'BELLEVU: Bellevue'. The 'General' tab is selected. The 'Calculate price inclusive tax exempt' option is highlighted with a red box and is set to 'Yes'. Other settings include 'Retail Channel Id' (000022), 'Name' (Bellevue), 'Store number' (BELLEVU), 'Operating unit number' (057), 'Legal entity' (usr), 'Warehouse' (BELLEVU), 'Shipping warehouse' (BELLEVU), 'Store time zone' ((GMT-08:00) Pacific Time (US ...)), 'Channel profile' (Default), 'Live channel database', 'Offline profile' (AX7), 'Sales tax group' (WA), 'Prices include sales tax' (Yes), 'Use destination-based tax' (No), 'Use customer-based taxes' (No), 'Sales tax override group' (Default), 'Customer address book' (RetailCust), and 'Employee address book' (Bellevue:USRTWest,...).

## Set up price reductions for tax exemptions

The following steps show how to test this capability in demo data scenarios. The setup steps for other data sets are similar.

1. Go to **Retail and Commerce > Channels > Stores > All stores**.
2. Select the **San Francisco** store. To open the store details, select the **Retail Channel Id** value for the store.
3. Select **Edit**.

4. On the **General** FastTab, set the **Calculate price inclusive tax exempt** option to **Yes**.
5. Set the **Price includes tax** option to **Yes**.
6. Select **Save**.
7. Enter **Sales tax groups** in the search field to open the **Sales tax groups** page.
8. Select **New**, and enter a name for the sales tax group.
9. On the **Setup** FastTab, select **Add**.
10. On the drop-down list in the **Sales tax code** column, select **RP\_CAST**, and then select the **Exempt** check box.
11. Select **Save**.
12. Enter **Sales tax overrides** in the search field to open the **Sales tax overrides** page.
13. Select **New**, and enter a name for the sales tax override.
14. Set the status to **Enable**.
15. In the **Override type** field, select **Sales tax group**.
16. In the **From** field, select **Any tax group**.
17. In the **To** field, select **Specified tax group**.
18. In the **From tax group** field, select **CA**.
19. In the **To tax group** field, select the sales tax group that created earlier.
20. Select **Save**.
21. Enter **Sales tax override groups** in the search field to open the **Sales tax override groups** page.
22. Select the **Default** group, and then select **Edit**.
23. Select **Add**, and then select the sales tax override that you created earlier.
24. Select **Save**.
25. Enter **Distribution schedules** in the search box to open the **Distribution schedules** page.
26. Select schedule job **9999**, and then select **Run now**.
27. When the jobs have completed synchronization, open the POS.

**NOTE**

Tax details for the channel might be cached. Then close the point of sale application and relaunch it to observe the changes after they have synchronized to the channel database.

28. Add item **91050** to a transaction.
29. Select **Tax overrides**, and then select **Override transaction tax**.
30. Select the sales tax override that you created earlier. The tax is reduced to 0 (zero), and the price for the line items is reduced to reflect the tax exemption.

Alternatively, you can set the **Use customer based tax** option for the store to **Yes** and then assign the sales tax group that you create directly to the customer. Then, when the customer is added to a transaction, the prices are

reduced to reflect that customer's tax-exempt status.

### Check customers for exemptions when tax is exclusive of price

Some retail verticals, such as liquor stores, sell goods to individuals and other businesses in cash-and-carry transactions. However, in many cases, transactions that involve different customer segments have different requirements for taxation purposes. For example, when a liquor store sells goods to some businesses, sales taxes that are usually associated with the items that are sold might be exempt for those specific businesses. However, in all other cases, regular sales tax should apply.

To support this scenario, set the **Calculate customer tax exempt** option for the store to **Yes**. Then, when a customer is added to a transaction, the POS checks the taxes that are applicable to that customer. If the customer's tax settings have a tax code that is marked as **Exempt**, but tax is applicable to the transaction, the tax is treated as exempt for the transaction and isn't added to the transaction.

The **Calculate customer tax exempt** option applies to stores where the price doesn't include tax. The exemption calculation is also supported if the **Use destination based tax** option for the store is set to **Yes**.

### Set up tax exemption calculations for customers

The following steps show how to test this capability in demo data scenarios. The setup steps for other data sets are similar.

1. Go to **Retail and Commerce > Channels > Stores > All stores**.
2. Select the **San Francisco** store. To open the store details, select the **Retail Channel Id** value for the store.
3. Select **Edit**.
4. On the **General** FastTab, set the **Calculate customer tax exempt** option to **Yes**.
5. Select **Save**.
6. Enter **Sales tax groups** in the search field to open the **Sales tax groups** page.
7. Select **New**, and enter a name for the sales tax group.
8. On the **Setup** FastTab, select **Add**.
9. On the drop-down list in the **Sales tax code** column, select **RP\_CAST**, and then select the **Exempt** check box.
10. Select **Save**.
11. Go to **Retail and Commerce > All customers**.
12. Select account ID **004009** for **Matthew Tolley**.
13. Select **Edit**.
14. On the **Invoice and delivery** FastTab, in the **Sales tax group** field, select the sales tax group that you created earlier.
15. Select **Save**.
16. Enter **Distribution schedules** in the search field to open the **Distribution schedules** page.
17. Select schedule job **9999**, and then select **Run now**.
18. When the jobs have completed synchronization, open the POS.
19. Add item **91050** to a transaction. The total that is due is **\$75.06**.
20. Add **Matthew Tolley** to the transaction. Taxes are recalculated to reflect this customer's exemption.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure sales tax for online orders

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of sales tax group selection for different online order types.

Your e-commerce channel may want to support options like delivery or pickup for online orders. The sales tax applicability is based on the option selected by your online users. When a site customer chooses to buy an item online and gets it shipped to an address, the sales tax is determined based on the customer's shipping address tax group setting. When a customer opts to pick up a purchased item at a store, the sales tax is determined based on the pickup store's tax group setting.

## Orders shipped to a customer address

In general, taxes for online orders that ship to customer addresses are defined by the destination. Every sales tax group has a retail destination-based tax configuration in which your business can define destination details such as county/region, state, county, and city in a hierarchical form. When an online order is placed, the Commerce tax engine uses the delivery address of every line item in the order, and finds sales tax groups with matching destination-based tax criteria. For example, for an online order with a line item delivery address to San Francisco, California, the tax engine will find the sales tax group and sales tax code for California and then calculate tax for each line item accordingly.

## Customer-based tax groups

In Commerce headquarters, there are two places where customer tax groups are configured:

- **Customer's profile**
- **Customer's shipping address**

### **If a customer's profile has a tax group configured**

A customer's profile record in headquarters may have a sales tax group configured, however for online orders the sales tax group configured in a customer's profile will not be used by the tax engine.

### **If a customer's shipping address has a tax group configured**

If a customer's shipping address record has a tax group configured and an online order (or line item) is shipped to the customer's shipping address, the tax group configured in the customer's address record will be used by the tax engine for tax calculations.

#### **Configure a tax group for a customer's shipping address record**

To configure a tax group for a customer's shipping address record in Commerce headquarters, follow these steps.

1. Go to **All customers**, and then select the desired customer.
2. On the **Addresses** FastTab, select the desired address, and then select **More options > Advanced**.
3. Under the **General** tab on the **Manage addresses** page, set the sales tax value as needed.

#### **NOTE**

The tax group is defined using the delivery address of the order line and the destination-based taxes are configured at the tax group itself. For more information, see [Set up taxes for online stores based on destination](#).

## Order pickup in store

For order lines with pickup in store or curbside pickup specified, the tax group from the selected pickup store will be applied. For details about how to configure the tax group for a given store, see [Set other tax options for stores](#).

### NOTE

When an order line is picked up at a store, a customer's address tax settings (if set up) will be ignored by the tax engine and the pickup store's tax configuration will be applied.

## Additional resources

[Sales tax overview](#)

[Sales tax calculation methods in the Origin field](#)

[Sales tax assignment and overrides](#)

[Whole amount and Interval calculation options for sales tax codes](#)

[Calculation of tax exemption](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up an email notification profile

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create an email notification profile in Microsoft Dynamics 365 Commerce.

## Overview

Before creating channels, you'll want to set up a profile to ensure that email notifications can be sent out for various events, such as order creation, order shipping status, and payment failure.

For additional email configuration information, see [Configure and send email](#).

## Create an email notification profile

To create an email notification profile, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Headquarters setup > Commerce email notification profile**.
2. On the action pane, click **New**.
3. In the **Email notification profile** field, enter a name to identify the profile.
4. In the **Description** field, enter a relevant description.
5. Set the **Active** switch to **Yes**.

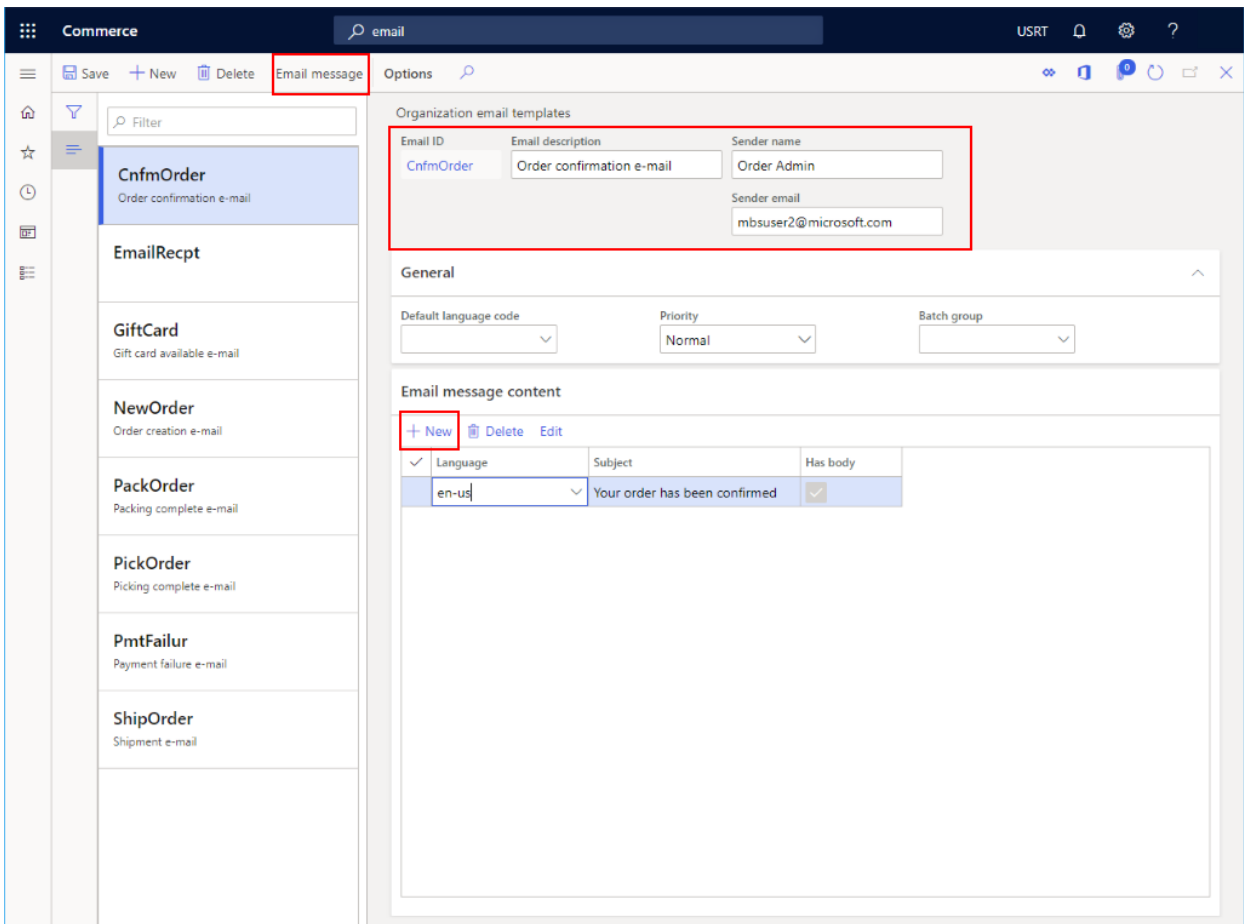
### Create an email template

Before an email notification can be created, you must create an organization email template which contains the senders email information and the email template.

To create an email template, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Headquarters setup > Parameters > Organization email templates**.
2. On the action pane, select **New**.
3. In the **Email ID** field, enter an ID to help identify this template.
4. In the **Sends name** field, enter the senders name.
5. In the **Email Description**, enter a meaningful description.
6. In the **Sender email**, enter the senders email address.
7. In the **General** section, fill out any optional information needed (such as the email priority).
8. Expand the **Email message content** section and select **New** to create the template content. For each content item, select the language and provide the email subject line. If the email will have a body, ensure that the **Has body** box is checked.
9. On the action pane, select **Email message** to provide an email body template.

The following image shows some example email template settings.

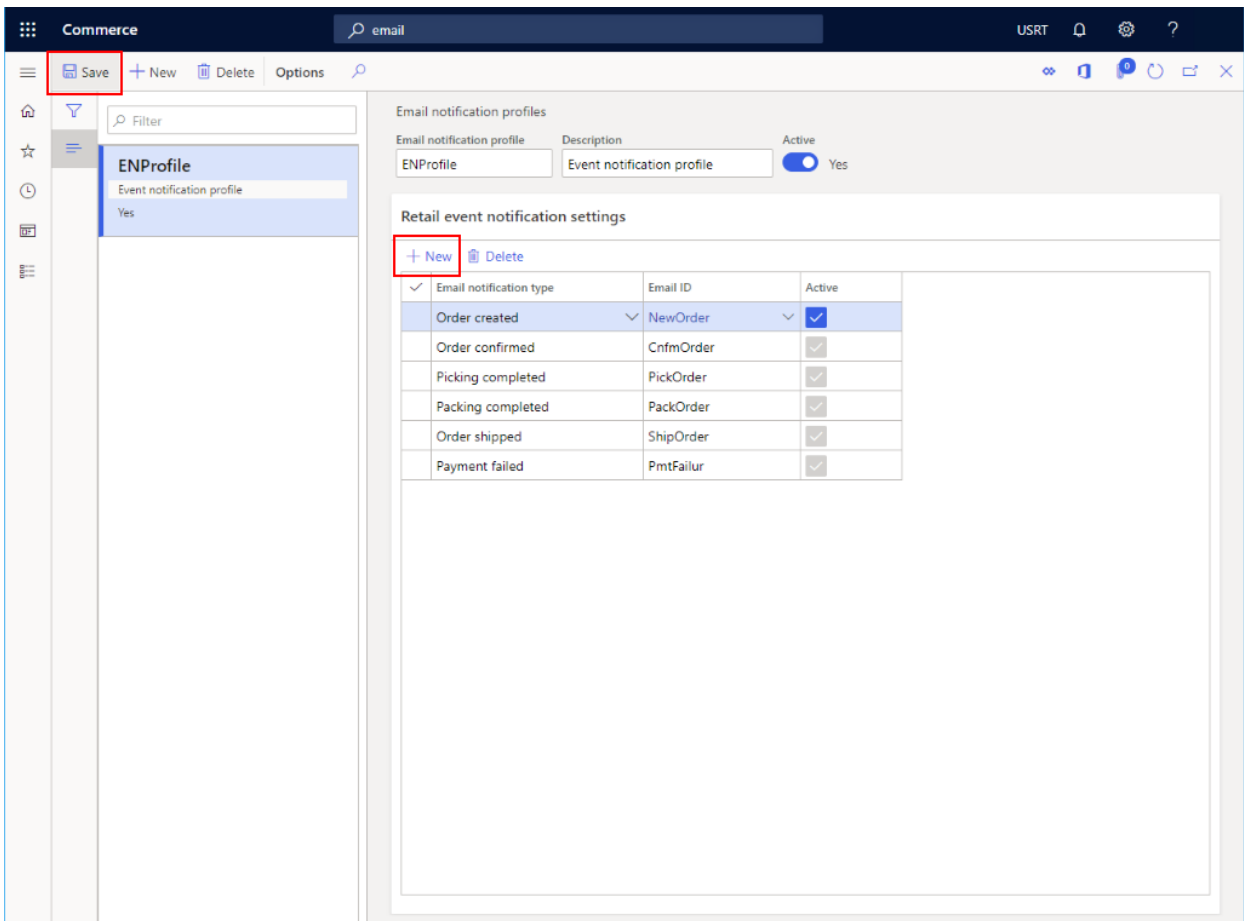


## Create an email event

To create an email event, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Headquarters setup > Commerce email notification profile**.
2. In the list, find and select the desired record.
3. Select the email template from the **Email ID** drop-down list.
4. Select the appropriate **Email notification type** from the drop-down list.
5. Select the **Active** check box.
6. On the action pane, select **Save**.

The following image shows some example event notification settings.



### Next steps

Before you can send mails, you must configure your outgoing mail service and set up a batch job. For more information, see [Configure and send email](#).

## Additional resources

[Configure and send email](#)

[Channels overview](#)

[Channel setup prerequisites](#)

[Organizations and organizational hierarchies overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Number sequences overview

2/18/2021 • 4 minutes to read • [Edit Online](#)

Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require identifiers. A master data record or transaction record that requires an identifier is referred to as a *reference*.

Before you can create new records for a reference, you must set up a number sequence and associate it with the reference. We recommend that you use the pages in **Organization administration** to set up number sequences. If module-specific settings are required, you can use the parameters page in a module to specify number sequences for the references in that module. For example, in **Accounts receivable** and **Accounts payable**, you can set up number sequence groups to allocate specific number sequences to specific customers or vendors.

When you set up a number sequence, you must specify a scope, which defines which organization uses the number sequence. The scope can be **Shared**, **Company**, **Legal entity**, or **Operating unit**. **Legal entity** and **Company** scopes can be combined with **Fiscal calendar period** to create even more specific number sequences.

Number sequence formats consist of segments. Number sequences with a scope other than **Shared** can contain segments that correspond to the scope. For example, a number sequence with a scope of **Legal entity** can contain a legal entity segment. By including a scope segment in the number sequence format, you can identify the scope of a particular record by looking at its number.

In addition to segments that correspond to scopes, number sequence formats can contain **Constant** and **Alphanumeric segments**. A **Constant** segment contains a set of letters, numbers, or symbols that does not change. An **Alphanumeric** segment contains a set of letters or numbers that increment every time that a number is used. Use a number sign (#) to represent incrementing numbers and an ampersand (&) to represent incrementing letters. For example, the format #####\_2017 creates the sequence 00001\_2017, 00002\_2017, and so on.

## Number sequence examples

The following examples show how to use segments to create number sequence formats. In particular, the examples demonstrate the effects of using scope segments.

### Expense report numbers

In the following example, expense report numbers are set up for the legal entity that is titled **CS**.

- **Area:** Travel and expense
- **Reference:** Expense report number
- **Scope:** Legal entity
- **Legal entity:** CS

SEGMENTS	SEGMENT TYPE	VALUE
Segment 1	Legal entity	CS
Segment 2	Constant	-EXPENSE-

SEGMENTS	SEGMENT TYPE	VALUE
Segment 3	Alphanumeric	####

**Example of formatted number:** CS-EXPENSE-0039

You can set up a similar number sequence format for other legal entities. For example, for a legal entity that is named **RW**, if you change only the value of the legal entity segment, the formatted number is RW-EXPENSE-0039. You can also change the whole number sequence format for other legal entities. For example, you can omit the legal entity scope segment to create a formatted number such as Exp-0001.

### Sales order numbers

In the following example, sales order numbers are set up for the company ID CEU.

- **Area:** Sales
- **Reference:** Sales order
- **Scope:** Company
- **Company:** CEU

SEGMENTS	SEGMENT TYPE	VALUE
Segment 1	Constant	SO-
Segment 2	Alphanumeric	####

**Example of formatted number:** SO-0029

Even though a scope segment is not included in the format, numbering restarts for each company ID. If you use the same format for all company IDs, the same numbers are used in each company. For example, sales order number SO-0029 is used in each company. You can also change the whole number sequence format for other company IDs.

### Purchase requisition numbers

In the following example, purchase requisition numbers are organization-wide.

- **Area:** Purchase
- **Reference:** Purchase requisition
- **Scope:** Shared

SEGMENTS	SEGMENT TYPE	VALUE
Segment 1	Constant	Req
Segment 2	Alphanumeric	####

**Example of formatted number:** Req0052

Because the scope is **Shared**, the number sequence format is used across the organization. You cannot set up different number sequence formats for different parts of the organization.

## Performance considerations for number sequences

Consider the following information about how the configuration of number sequences can affect system performance before you set up number sequences.

## Continuous and non-continuous number sequences

Number sequences can be continuous or non-continuous. A continuous number sequence does not skip any numbers, but numbers may not be used sequentially. Numbers from a non-continuous number sequence are used sequentially, but the number sequence may skip numbers. For example, if a user cancels a transaction, a number is generated, but not used. In a continuous number sequence, that number is recycled later. In a non-continuous number sequence, the number is not used.

Continuous number sequences are typically required for external documents, such as purchase orders, sales orders, and invoices. However, continuous number sequences can adversely affect system response times because the system must request a number from the database every time that a new document or record is created.

If you use a non-continuous number sequence, you can enable **Preallocation** on the **Performance** FastTab of the **Number sequences** page. When you specify a quantity of numbers to preallocate, the system selects those numbers and stores them in memory. New numbers are requested from the database only after the preallocated quantity has been used.

Unless there is a regulatory requirement that you use continuous number sequences, we recommend that you use non-continuous number sequences for better performance.

## Automatic cleanup of number sequences

In case of a power failure, an application error, or other unexpected failure, the system cannot recycle numbers automatically for continuous number sequences. You can run the cleanup process manually or automatically to recover the lost numbers.

Carefully consider server usage when you plan the cleanup process. We recommend that you perform the cleanup as a batch job during non-peak hours.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Set up number sequences on an individual basis

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to set up number sequences on an individual basis. Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require them. A master data or transaction record that requires an identifier is referred to as a reference. Before you can create new records for a reference, you must set up a number sequence and associate it with the reference. You can set up all required number sequences at the same time by using the **Set up number sequences** wizard, or you can create or modify individual number sequences by using the **Number sequences** page.

1. Go to **Navigation pane > Modules > Organization administration > Number sequences > Number sequences**.
2. Select **Number sequence**.
3. In the **Number sequence code** field, type a value.
4. In the **Name** field, type a value.
5. On the **Scope parameters** FastTab, select a scope for the number sequence and select scope values from the drop-down list. The scope defines which organizations use the number sequence. In addition, number sequences that have a scope other than **Shared** can have segments that correspond to their scope. For example, a number sequence with a scope of **Legal entity** can have a legal entity segment. For more information about scopes, see [Number sequence overview](#).
6. Expand the **Segments** section.
  - Define the format for the number sequence by adding, removing, and rearranging segments.
  - Number sequences of all scopes can contain *Constant segments* and *Alphanumeric segments*. Constant segments contain a set of alphanumeric characters that do not change. Use this segment type to add a hyphen or other separators between number sequence segments. Alphanumeric segments contain a combination of number signs (#) and ampersands (&). These characters represent letters and numbers that increment every time that a number from the sequence is used. Use a number sign (#) to indicate incrementing numbers and an ampersand (&) to indicate incrementing letters. For example, the format `#####_2014` creates the sequence `00001_2014`, `00002_2014`, and so on. At least one alphanumeric segment must be present. Scope segments, such as company or legal entity, are not required. However, if you do not include scope segments in the format, numbers for the selected reference are still generated per scope.
7. Expand the **References** section. Select the document type or record to assign this number sequence to. This step is optional for sequences that are defined for special application usage patterns. In these scenarios, a new number is generated by using the value of a number sequence code or ID, without using a reference. An example of a special application usage pattern is a voucher series that is used for specific journal names. However, we do not recommend that you use such patterns.
8. Expand the **General** section. On the General FastTab, specify whether the number sequence is manual, and continuous or non-continuous. In addition, enter the lowest and highest numbers that can be used in the number sequence. We do not recommend changing a non-continuous number sequence to a continuous number sequence. The number sequence will not be truly continuous. This change may also cause duplicate key violations in the database. In addition, continuous number sequences have a larger effect on performance.
9. Click **Save**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up number sequences using a wizard

2/18/2021 • 2 minutes to read • [Edit Online](#)

Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require them. A master data or transaction record that requires an identifier is referred to as a reference. Before you can create new records for a reference, you must set up a number sequence and associate it with the reference. This topic explains how to set up all required number sequences at the same time by using a wizard. The demo data company used to create this procedure is USMF.

1. Go to **Navigation > Modules > Organization administration > Number sequences > Number sequences**.
2. Select **Generate**.
3. Select **Next**.
  - On this page, you can modify the identification code, the lowest value, and the highest value. In addition, you can indicate whether the number sequence must be continuous.
  - Do not select the **Continuous** option if you must preallocate numbers for the number sequence. To add a scope segment to the format of a number sequence, select the format in the list, and then select **Include scope in format**. To remove a scope segment from the format of a number sequence, select the format in the list, and then select **Remove scope from format**. To exclude a number sequence from automatic generation, select the number sequence in the list, and then select **Delete**.
4. Select **Next**.
5. Select **Finish**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a default customer

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a default customer to use when creating a channel in Microsoft Dynamics 365 Commerce.

## Overview

When creating a channel, you will need to provide a default customer. A default customer can easily be created after first creating the customer group and customer address book.

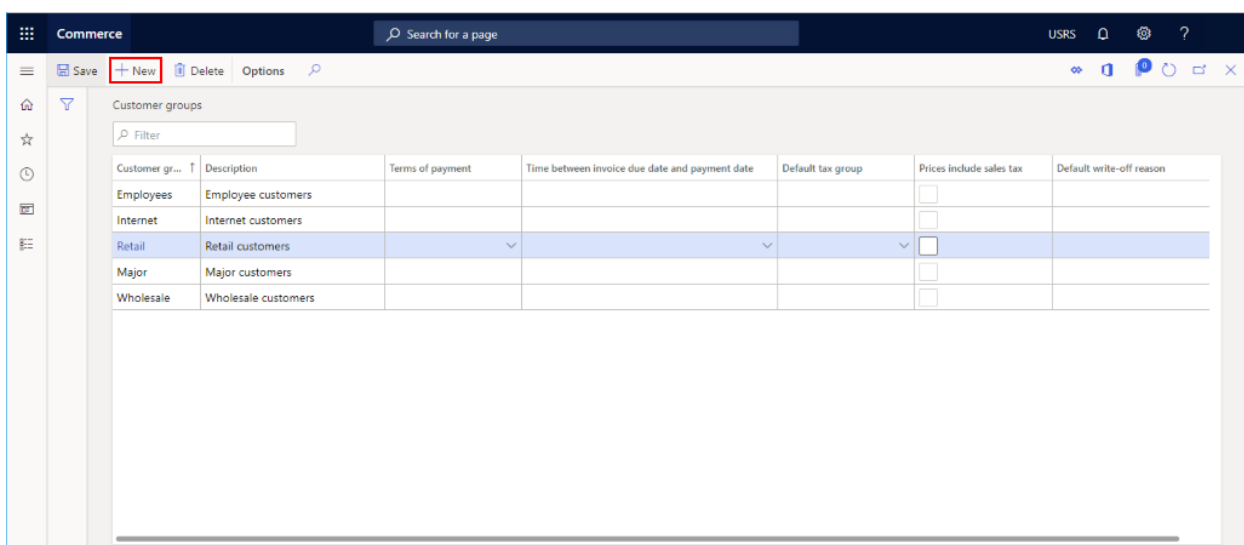
## Create a customer group

If no customer groups exist yet, you can create one. Examples may be groups to represent different customer groups, such as wholesale, retail, Internet, Employees, etc.

To create a customer group, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Customers > Customer groups**.
2. On the action pane, select **New**.
3. In the **Customer group** box, enter a customer group ID.
4. In the **Description** box, enter an appropriate description.
5. In the **Terms of payment** box, enter an appropriate value.
6. In the **Time between invoice due date and payment date** box, enter an appropriate value.
7. In the **Default tax group** box, enter a tax group if applicable.
8. Select the **Prices include sales tax** check box if applicable.
9. In the **Default write-off reason** box, enter an appropriate value, if applicable.

The following image shows several configured customer groups.



Customer gr...	Description	Terms of payment	Time between invoice due date and payment date	Default tax group	Prices include sales tax	Default write-off reason
Employees	Employee customers				<input type="checkbox"/>	
Internet	Internet customers				<input type="checkbox"/>	
Retail	Retail customers				<input checked="" type="checkbox"/>	
Major	Major customers				<input type="checkbox"/>	
Wholesale	Wholesale customers				<input type="checkbox"/>	

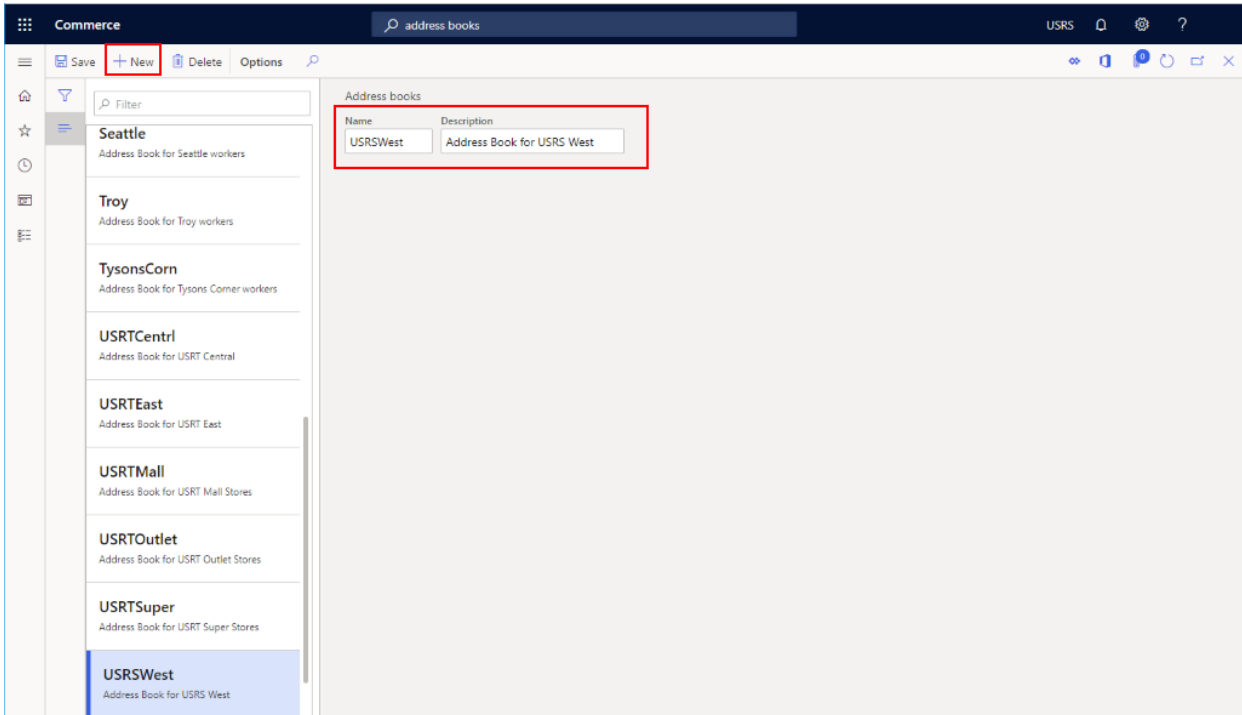
## Create a customer address book

A customer needs to be associated with an address book. If one has not yet been created, then you will need to create one.

To create a customer address book, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Channel setup > Address Books**.
2. On the action pane, select **New**.
3. In the **Name** box, enter a name.
4. In the **Description** box, enter a description.
5. On the action pane, select **Save**.

The following image shows an example address book.



## Create a default customer

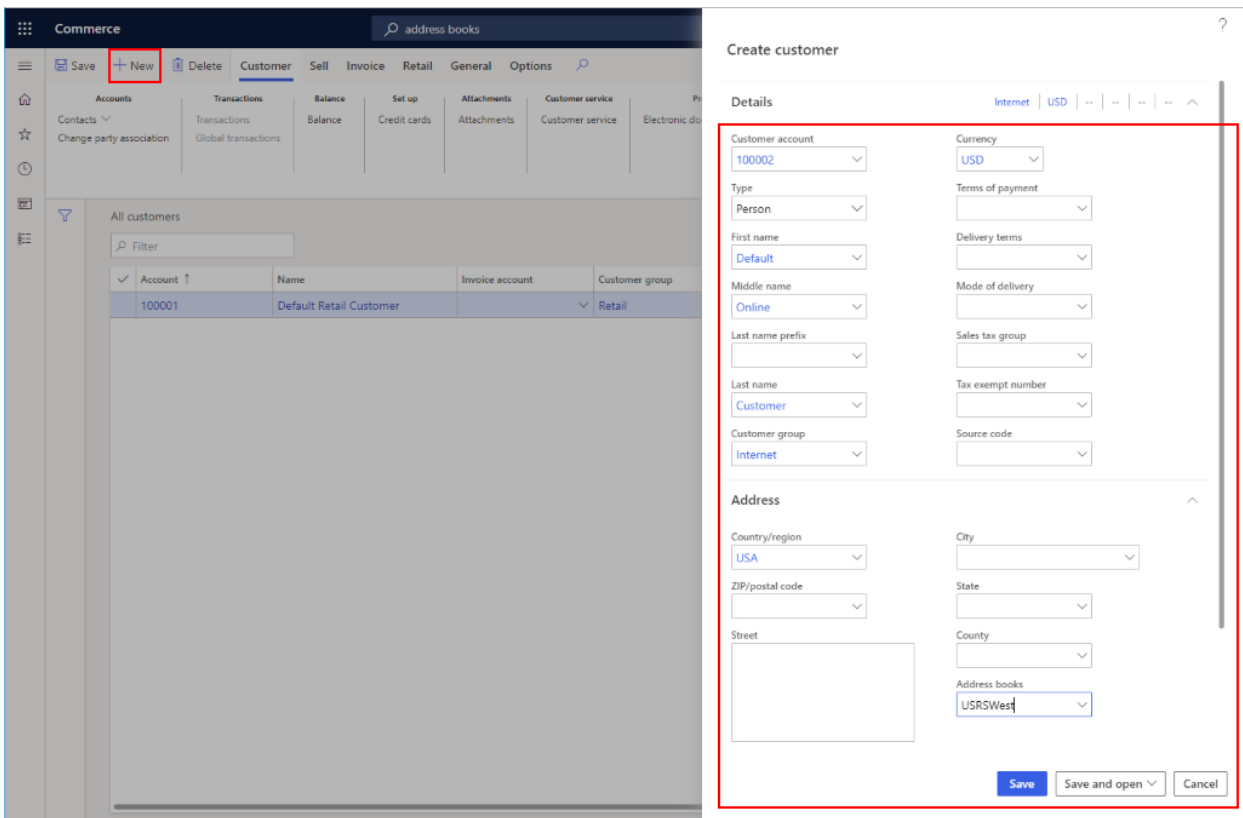
To create a default customer, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Customers > All customers**.
2. On the action pane, select **New**.
3. In the **Type** drop-down list, select "Person".
4. In the **Customer account** drop-down list, select or enter an account number (for example, "100001").
5. In the **First name** drop-down list, select or enter a name (for example, "Default").
6. In the **Middle name** drop-down list, select or enter a name (for example, "Retail").
7. In the **Last name** drop-down list, select or enter a name (for example, "Customer").
8. In the **Currency** drop-down list, select or enter a currency (for example, "USD").
9. In the **Currency** drop-down list, select the customer group created previously.
10. In the **Address books** drop-down list, select an existing customer address book.
11. Select **Save** to save and return to customer details screen for the new customer.

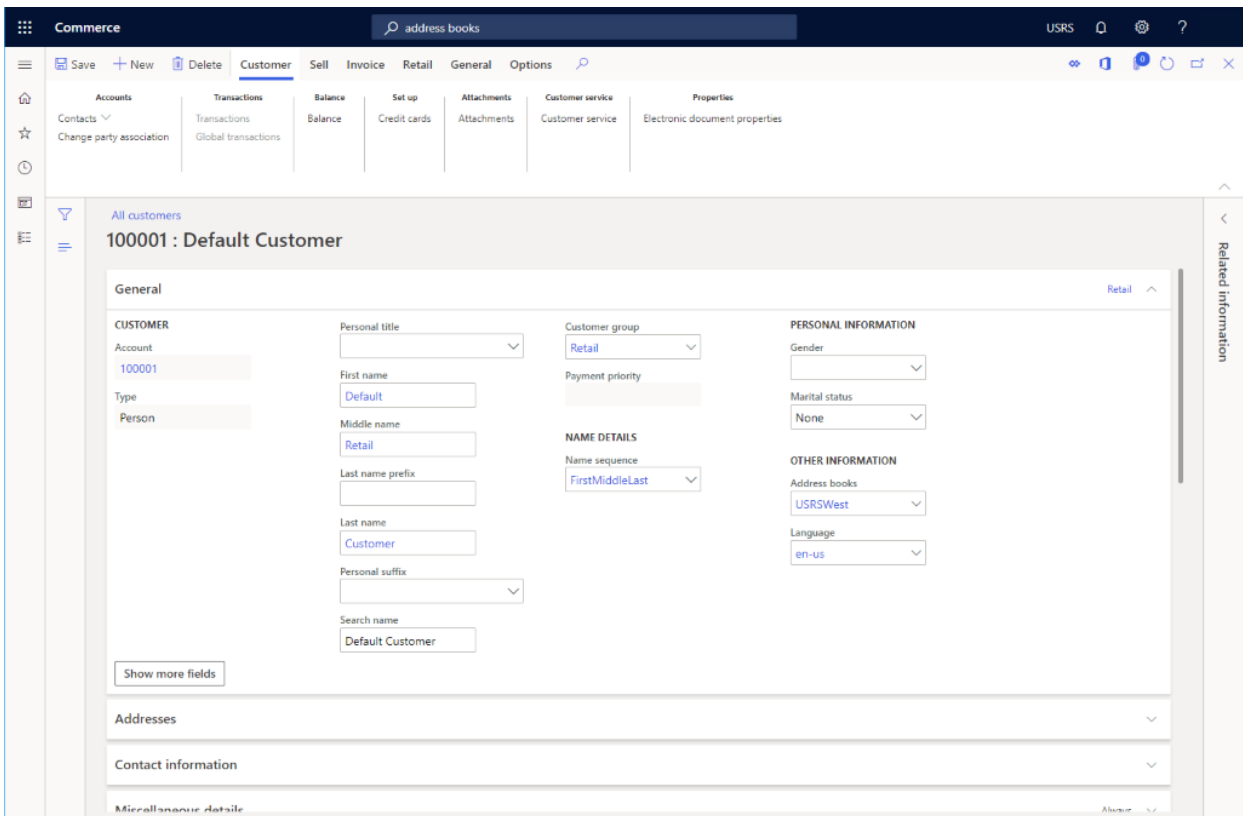
### NOTE

It is not necessary to add an address for a default customer.

The following image shows an example of customer creation.



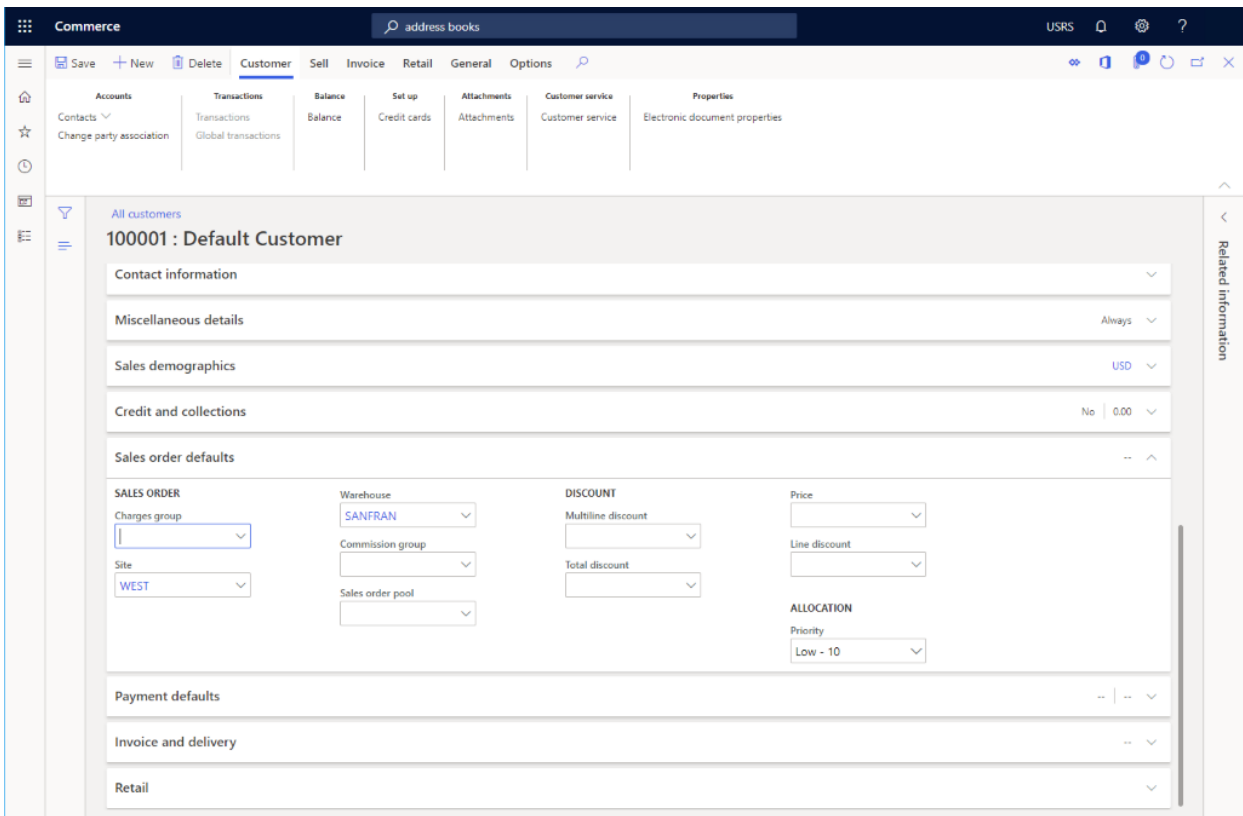
The following image shows a default customer configuration.



Most of the default values on the customer details screen can remain, but two values should be changed.

1. On the customer details screen, expand **Sales order defaults**.
2. In the **Site** drop-down list, select or enter a pre-configured site.
3. In the **Warehouse** drop-down list, and select or enter a pre-configured warehouse.

The following image shows an example customer configuration.



## Additional resources

[Channels overview](#)

[Channel setup prerequisites](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Info codes and info code groups

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article provides an overview about info codes, info code groups, and how to use them.

Info codes provide a way for you to capture data at a point-of-sale (POS) register. You can use info codes to prompt the cashier to enter information during various actions at the POS, such as item sales, item returns, or selecting customers. Cashiers can select input from a list or enter it as a code, number, date, or text. You can assign info codes to predefined store actions, retail items, payment methods, customers, or specific point-of-sale activities. You can use info codes to do the following:

- Capture additional information at transaction time, such as a flight number or the reason for a return.
- Prompt the register cashier to select from a list of prices for specific products.
- Link a subcode to an info code to prompt the cashier for input when performing a specific activity. For example, when a customer returns a product, you can prompt the cashier to ask why the product is being returned. Then you can use subcodes to display a list of reasons that the cashier can choose from.
- Sell a product as a regular sale, discounted sale, or free product.
- Prompt the cashier to enter a value or select from a list of subcodes when the register drawer is opened without performing a sales operation.

## Info codes group

In Commerce, you can create groups of info codes. Info code groups add flexibility by enabling you to define fewer info codes and then use them in more versatile ways. You can use info code groups in the following ways:

- Define fewer info codes and easily re-use them. Info codes that are included in info code groups have no predefined dependencies on other info codes. You can include the same info code in multiple info code groups and then use prioritization to present the same info codes in the order that makes sense in any particular situation.
- Link info codes to other info codes or info code groups to gather information about a product or transaction without having to define a separate info code or linked info code for each scenario.

## Info code examples

### Example 1: Reuse info codes

You can link info codes so that when one info code is triggered, another info code is triggered immediately after it. For example, when you sell certain products, you can prompt the cashier to ask the customer if they want to purchase batteries and a product warranty. For other products, you can prompt the cashier to ask the customer if they want to purchase batteries and collect their postal code. If you create linked info codes for these scenarios, you must set up every variation of the info code so that the cashier is prompted to ask for the right information. If you use info code groups, common info codes, such as asking for batteries, can be set up once and then reused in multiple info code groups. You can also use prioritization in the info code groups to identify the order in which the prompts are displayed.

### Example 2: Link info codes to info code groups

When you sell certain products, for example mobile devices, you always want to collect a specific set of information, such as telephone number, mobile equipment identifier (MEID), and serial number. However, you also want to collect different information for a tablet versus a mobile phone. You can set up an info code group that includes prompts for the telephone number, MEID, and the serial number, and then link the info code group



to an individual info code. When the product-specific info code is triggered, the info code group can be triggered next to enable you to collect the common data without having to define multiple sets of linked info codes for each device.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a retail functionality profile

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a functionality profile in Microsoft Dynamics 365 Commerce.

## Overview

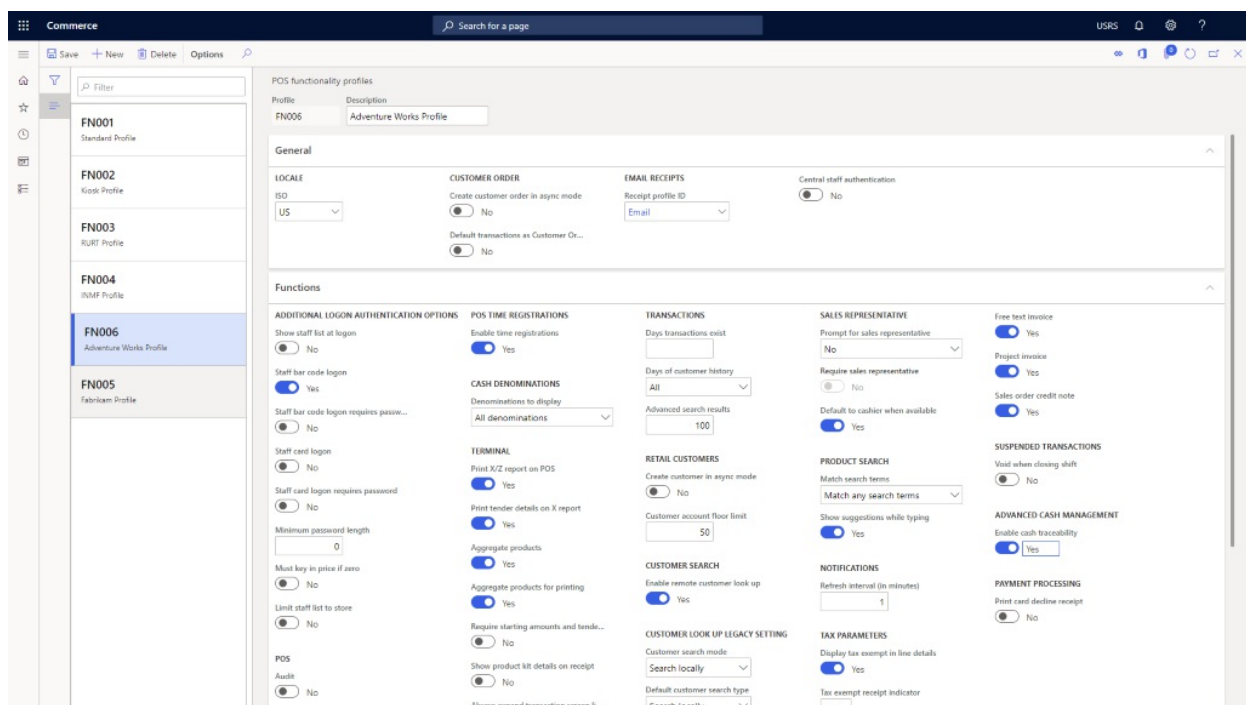
The commerce functionality profile provides various settings used for online channels. Each channel must specify a functionality profile.

## Create a functionality profile

To create a functionality profile, follow these steps.

1. In the navigation pane, go to **Modules > Channel setup > POS profiles > Functionality profiles**.
2. On the action pane, select **New**.
3. In the **Profile** field, enter an ID for the profile ("FN006" in the example image below).
4. In the **Description** field, enter a value ("Adventure Works Profile" in the example image below).
5. In the **General** section, select a country for the ISO locale.
6. In the **General** section, modify other settings, as needed.
7. In the **General** section, select a **Receipt profile ID** for email receipts.
8. In the **Functions** section, modify settings, as needed.
9. In the **Amount** section, modify settings, as needed.
10. In the **Info Codes** section, modify settings, as needed.
11. In the **Receipt numbering** section, modify settings, as needed.

The following image shows an example functionality profile.



## Additional resources

[Info codes and info code groups](#)

[Create new address book](#)

[Screen layout overview](#)

[Configure and install Retail hardware station](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create new address book

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a new address book in Microsoft Dynamics 365 Commerce.

## Overview

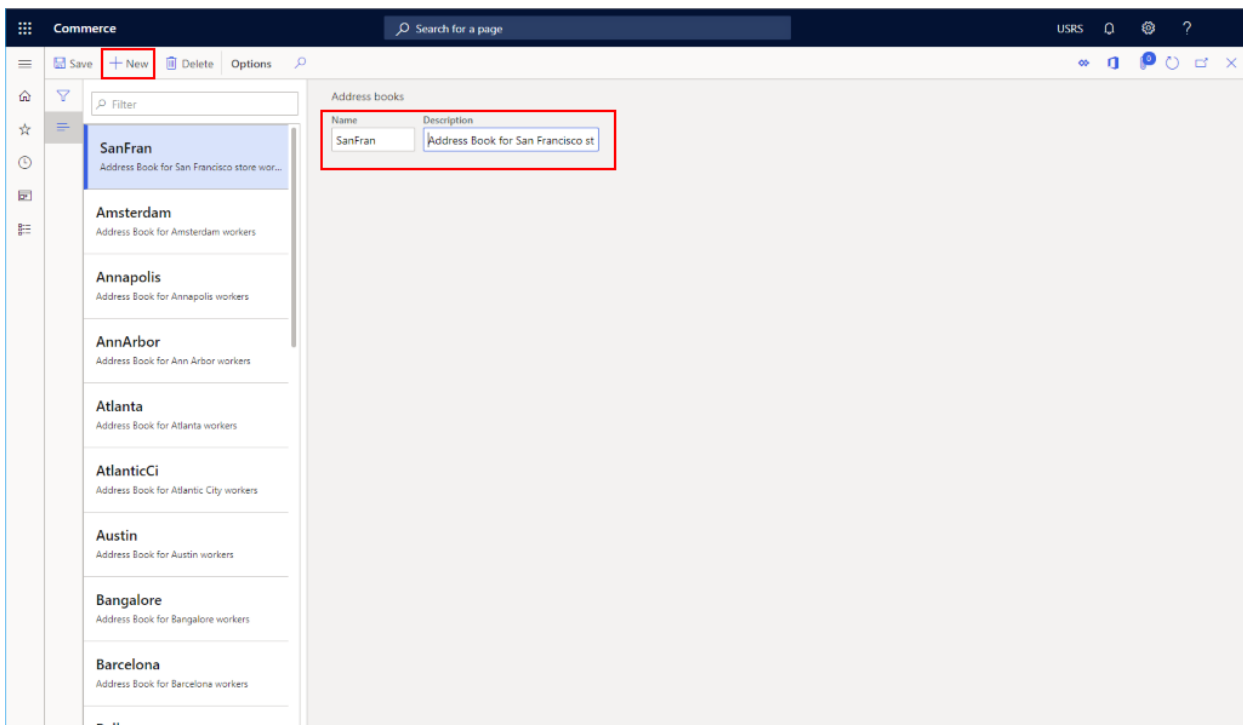
Address books are used in Commerce for various reasons including storing customer lists and employee lists for a channel. Address books can be used for a single channel or shared between channels.

## Create a new address book

To create a new address book, follow these steps.

1. In the navigation pane, go to **Modules > Channel setup > Address books**.
2. On the action bar, select **New**.
3. Enter name and description information.
4. On the action bar, select **Save**.

The following image shows the creation of an employee address book for a retail store.



## Additional resources

[Info codes and info code groups](#)

[Create a retail functionality profile](#)

[Screen layout overview](#)

[Configure and install Retail hardware station](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

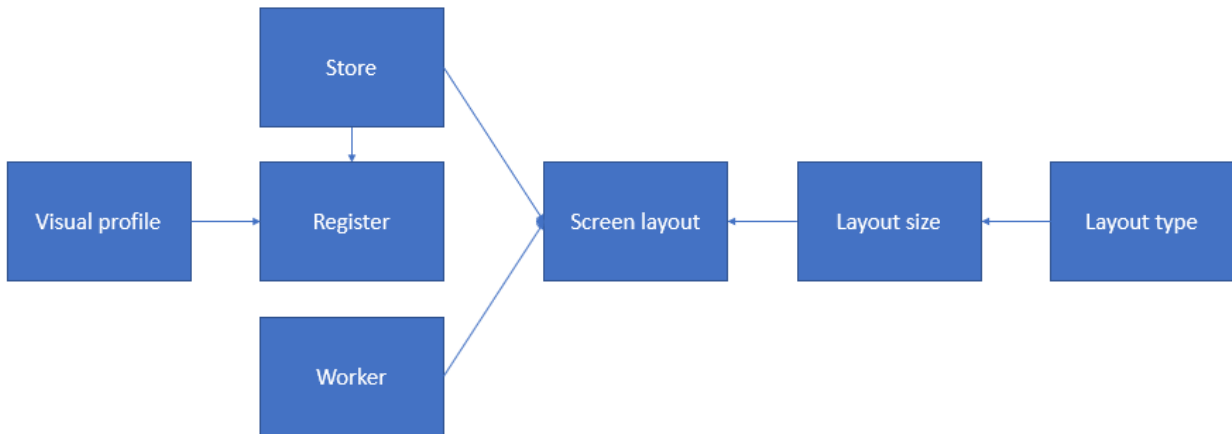
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# POS user interface visual configurations

2/18/2021 • 12 minutes to read • [Edit Online](#)

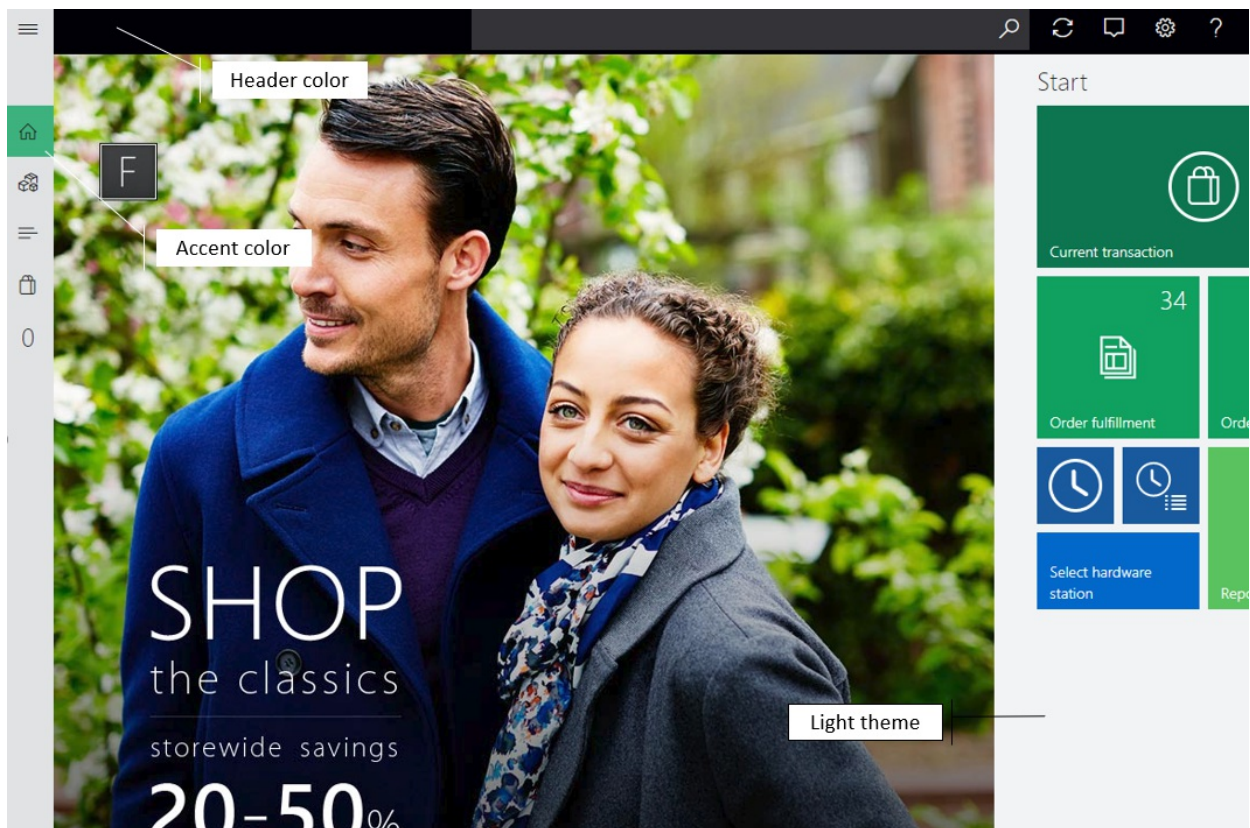
The user interface (UI) of the Microsoft Dynamics 365 Commerce point of sale (POS) can be configured by using a combination of visual profiles and screen layouts that are assigned to stores, registers, and users. This topic provides information about those configuration options.

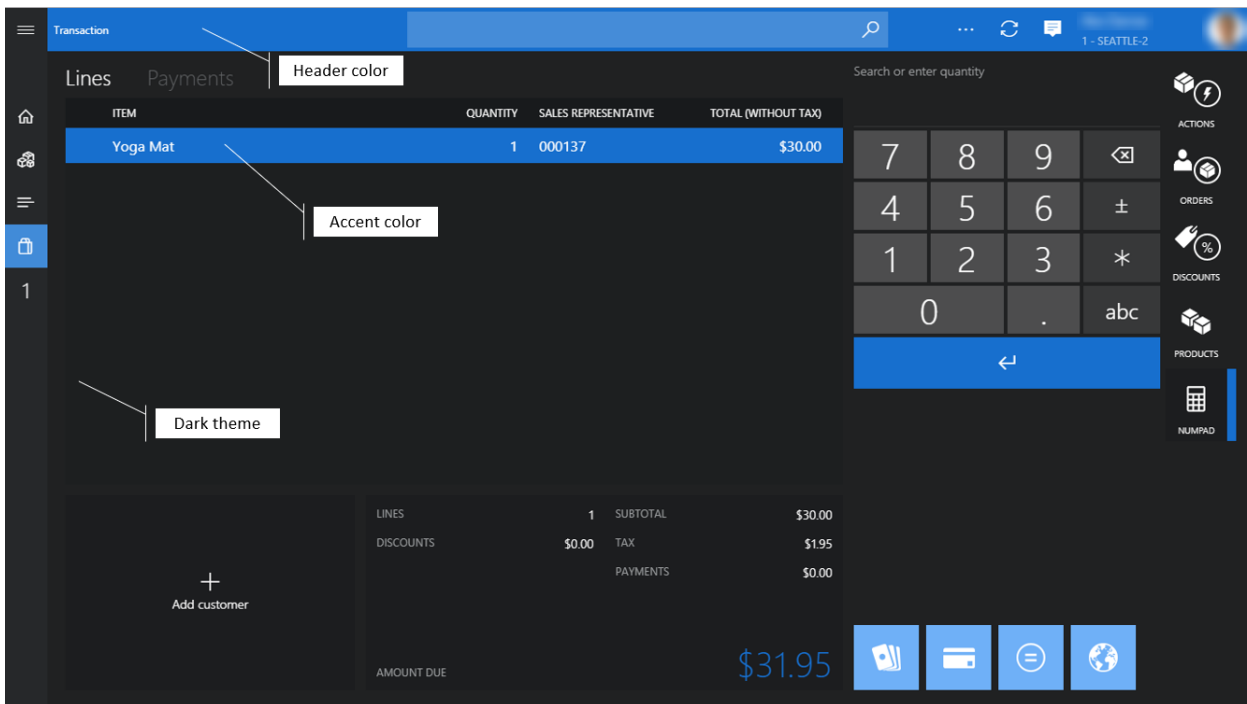
The following illustration shows the relationships among the various entities that make up the configurable aspects of the POS UI.



## Visual profile

Visual profiles are assigned to registers, and they specify the visual elements that are register-specific and shared across users. Every user who signs in to the register sees the same theme, layout, colors, and images.





- **Profile number** – The profile number is the unique identifier of the visual profile.
- **Description** – You can specify a meaningful name that will help identify the correct profile for your situation.
- **Theme** – You can select between the **Light** and **Dark** application themes. The theme affects the font and background colors throughout the application.
- **Accent color** – The accent color is used throughout the POS to differentiate or highlight specific visual elements, such as tiles, command buttons, and hyperlinks. Typically, these elements are actionable.
- **Header color** – You can configure the color of the page header to meet the retailer's branding requirements.
- **Font scheme** – You can select between the **Standard** and **Large** font schemes. The font scheme affects the font size throughout the application. The default selection is **Standard**.
- **Always show application bar labels** – When this option is turned on, the label text is always visible under the application bar buttons.
- **Layout** – You can select between the **Centered** and **Right** layouts. The layout affects the alignment of the sign-in box on the sign-in screen. The default selection is **Centered**.
- **Show date/time** – When this option is turned on, the current date and time are shown in the POS header and on the sign-in screen.
- **Keyboard** – You can select between **Default to OS keyboard** and **Show number pad** to specify the default keyboard that is used for input on the sign-in screen. The number pad is a virtual keyboard that is used primarily for touch-based devices. The default selection is **Default to OS keyboard**.
- **Logo image** – You can specify a logo image that is shown on the sign-in screen. We recommend that you use an image that has a transparent background. The file size should be kept as small as possible, because application behavior and performance can be affected when large files are stored and loaded.
- **Login background** – You can specify a background image for the sign-in screen. The file size of background images should be kept as small as possible.
- **Background** – You can specify a background image that is used instead of the solid theme color throughout the application. As for background images for the sign-in screen, the file size should be kept as small as possible.

#### NOTE

The **Right** layout and date/time display don't apply to the sign-in screen in compact view.

You need to run the **1090 (Registers)** distribution schedule job to synchronize the latest visual profile

configurations to the channel database.

## Screen layouts

Screen layout configurations determine the actions, content, and placement of UI controls on the POS **Welcome** screen and **Transaction** screen.

The screenshot shows the Dynamics 365 Retail POS setup interface for screen layouts. The left sidebar lists various screen layout configurations, with 'F3MGR Fabrikam Manager' selected. The main area displays the configuration for 'F3MGR Fabrikam Manager', showing the default start screen as 'Welcome screen'. Below this, there are three sections: 'LAYOUT SIZES', 'Button grids', and 'Images'. The 'LAYOUT SIZES' section contains a table with the following data:

Name	Layout type	Width	Height
1024x768 - Full	Modern POS - Full	1024	768
1280x720 - Full	Modern POS - Full	1280	720
1366x768 - Full	Modern POS - Full	1366	768
1440x960 - Full	Modern POS - Full	1440	960
480x853 - Compact	Modern POS - Compact	480	853

The 'Button grids' section shows a table with the following data:

Layout zone	Button grid ID	Name
Welcome screen 1	F2W1M	Start
Welcome screen 2	F2W2	Products
Welcome screen 3	F2W3	Inventory
Welcome screen 4	F2W4M	Shift and drawer
Welcome screen 5	F2W5M	Operations
Transaction screen 1	F2T1M	Actions
Transaction screen 2	F2T2	Customer orders

The 'Images' section shows a table with the following data:

Layout zone	Image
Image 1	2506
Image 2	2507
Image 3	2508

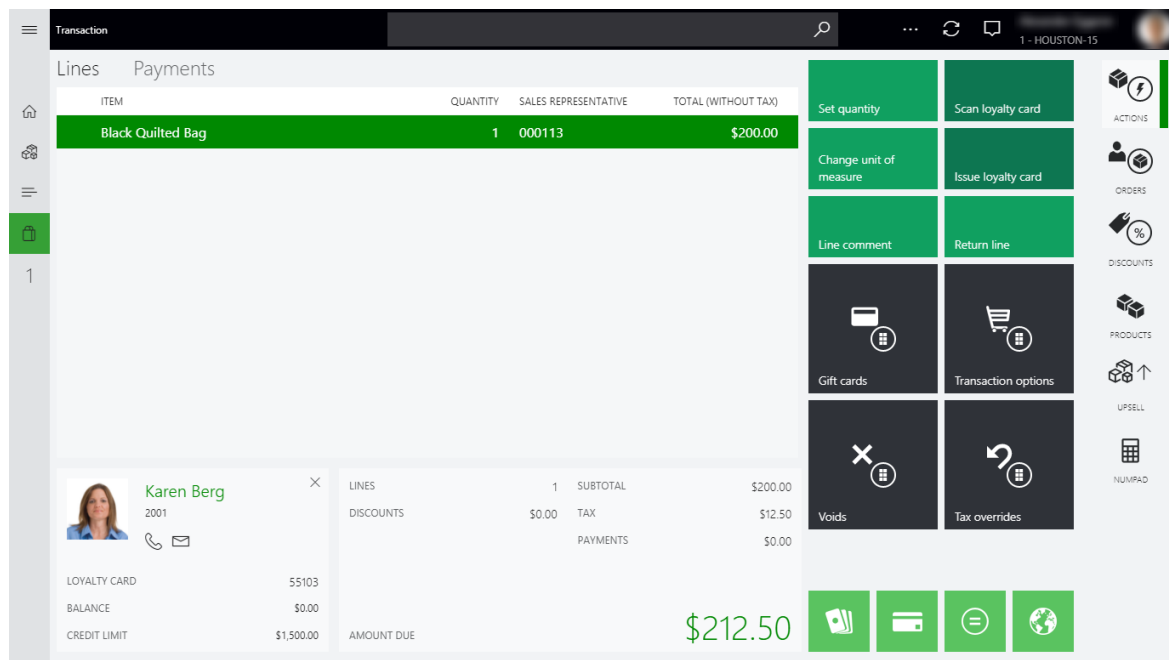
Callouts in the image point to the 'Screen layout' (F3MGR), 'Layout sizes' table, 'Button grids' table, and 'Images' table.

- **Welcome screen** – In most cases, the welcome screen is the page that users see when they first sign in to the POS. The welcome screen can consist of a branding image and button grids that provide access to POS operations. Typically, operations that aren't specific to the current transaction are put on this screen.

The screenshot shows the POS Welcome screen. The background features a large image of a person climbing a rock at sunrise, with the text 'RISE & SHINE! Earlybird specials are back again 20-40% SAVINGS'. The top right corner shows the user's name '1 - SEATTLE-2'. The main content area is a 'Start' screen with several button grids. Callouts in the image point to the 'Welcome screen button grid' and the 'Welcome screen image'.

- **Transaction screen** – The **Transaction** screen is the main screen in the POS for processing sales transactions and orders. The content and layout are configured by using the screen layout designer.





- **Default start screen** – Some retailers prefer that cashiers go directly to the **Transaction** screen after sign-in. The **Default start screen** setting lets you specify the default screen that appears after sign-in for each screen layout.

### Assignment

Screen layouts can be assigned at the store, register, or user level. The user assignment overrides the register and store assignments, and the register assignment overrides the store assignment. In a simple scenario where all users use the same layout, regardless of register or role, the screen layout can be set only at the store level. In scenarios where specific registers or users require specialized layouts, those layouts can be assigned.

Depending on which level the screen layouts are assigned, you need to run the **1070 (Channel configuration)**, **1090 (Registers)**, and/or **1060 (Staff)** distribution schedule jobs to synchronize the latest screen layout configurations to the channel database.

### Layout sizes

Most aspects of the POS UI are responsive, and the layout is automatically resized and adjusted based on the screen size and orientation. However, the POS **Transaction** screen must be configured for every screen resolution that is expected.

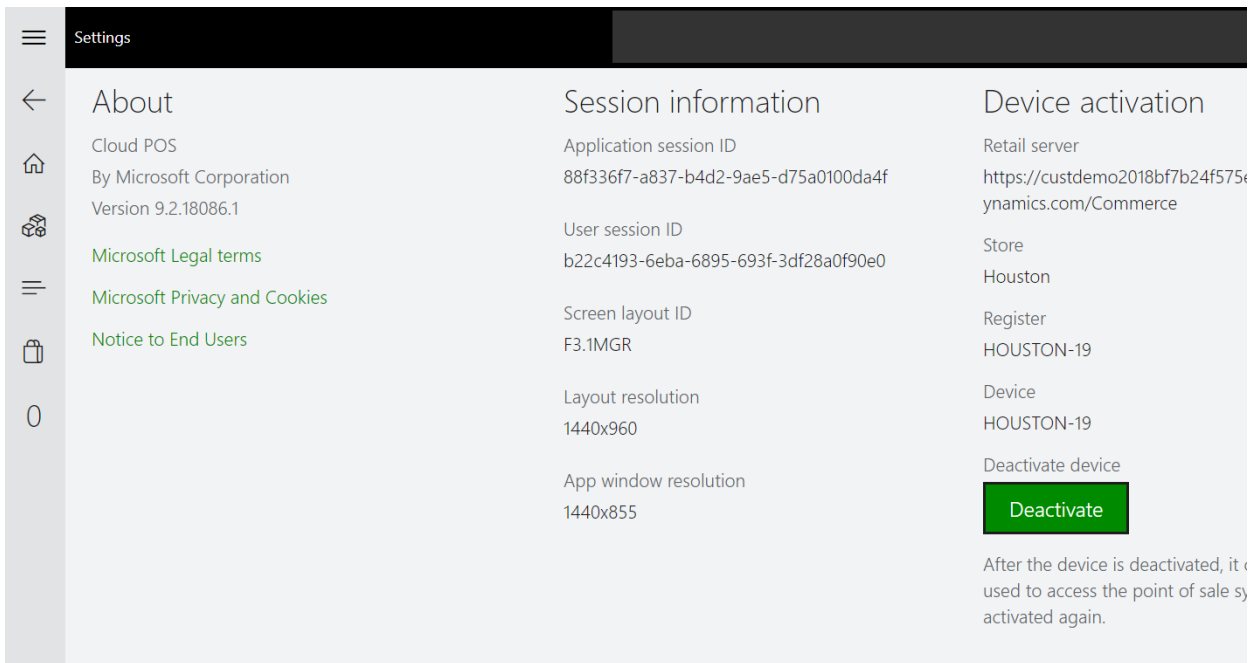
At startup, the POS application automatically selects the closest layout size that is configured for the device. A screen layout can also contain configurations for both landscape and portrait modes, and for both full-size and compact devices. Therefore, users can be assigned to a single screen layout that works across various sizes and form factors that are used in the store.



- **Name** – You can enter a meaningful name to identify the screen size.
- **Layout type** – The POS application can show its UI in various modes to provide the best user experience on a given device.
  - **Modern POS – Full** – Full layouts are typically best for larger displays, such as desktop monitors and tablets. You can select the UI elements to include, specify the size and placement of those elements, and configure their detailed properties. Full layouts support both portrait and landscape configurations.
  - **Modern POS – Compact** – Compact layouts are typically best for phones and small tablets. The design possibilities for compact devices are limited. You can configure the columns and fields for the receipt and totals panels.
- **Width/Height** – These values represent the effective screen size, in pixels, that is expected for the layout. Remember that some operating systems use scaling for high-resolution displays.

**TIP**

You can learn the layout size that is required for a POS screen by viewing the resolution in the app. Start the POS, and go to **Settings > Session information**. POS shows the screen layout that is currently loaded, the layout size, and the resolution of the app window.

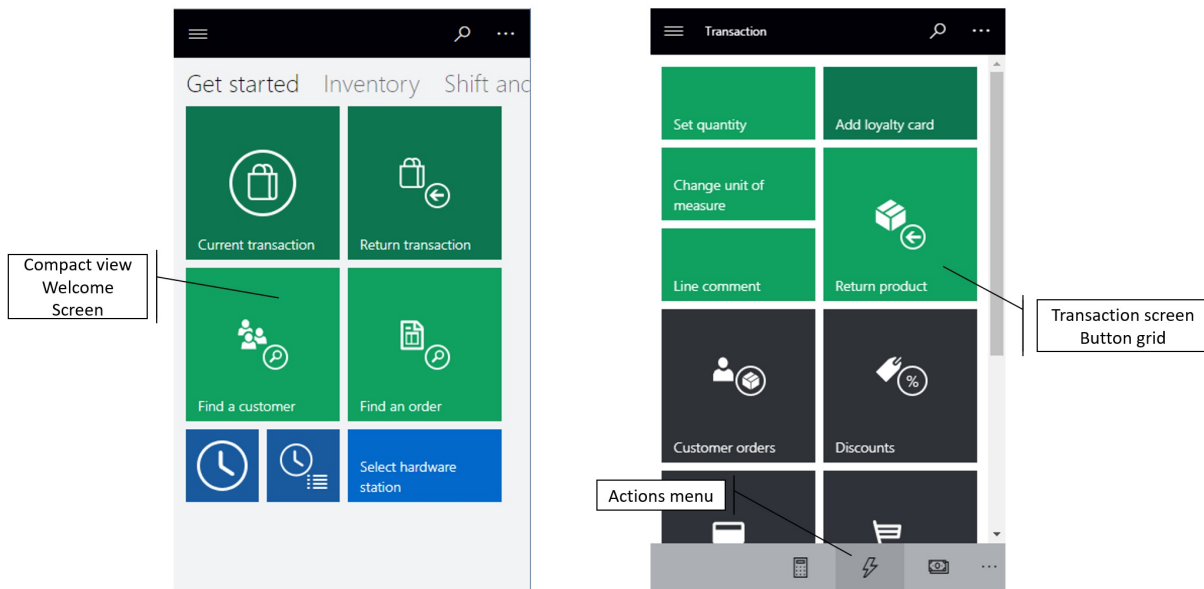


## Button grids

For each layout size in a screen layout, you can configure and assign button grids for the POS welcome screen and **Transaction** screen. Button grids for the welcome screen are automatically laid out from left to right, from the lowest number (Welcome screen 1) to the highest number.

In Full POS layouts, the placement of button grids is specified in the screen layout designer.

In Compact POS layouts, the button grids are automatically laid out from top to bottom, from the lowest number (Transaction screen 1) to the highest number. They can be accessed on the **Actions** menu.



### NOTE

The button sizes in the designer will scale to fit the size of the window, therefore they may not accurately reflect the actual buttons rendered in POS. To best simulate the button grid layout, adjust the designer windows to the same size as the POS.

## Images

For each layout size in a screen layout, you can specify images to include in the POS UI. For Full POS layouts, a

single image can be specified for the welcome screen. This image appears as the first UI element on the left. On the **Transaction** screen, images can be used as tab images or as a logo. Compact POS layouts don't use these images.

## Screen layout designer

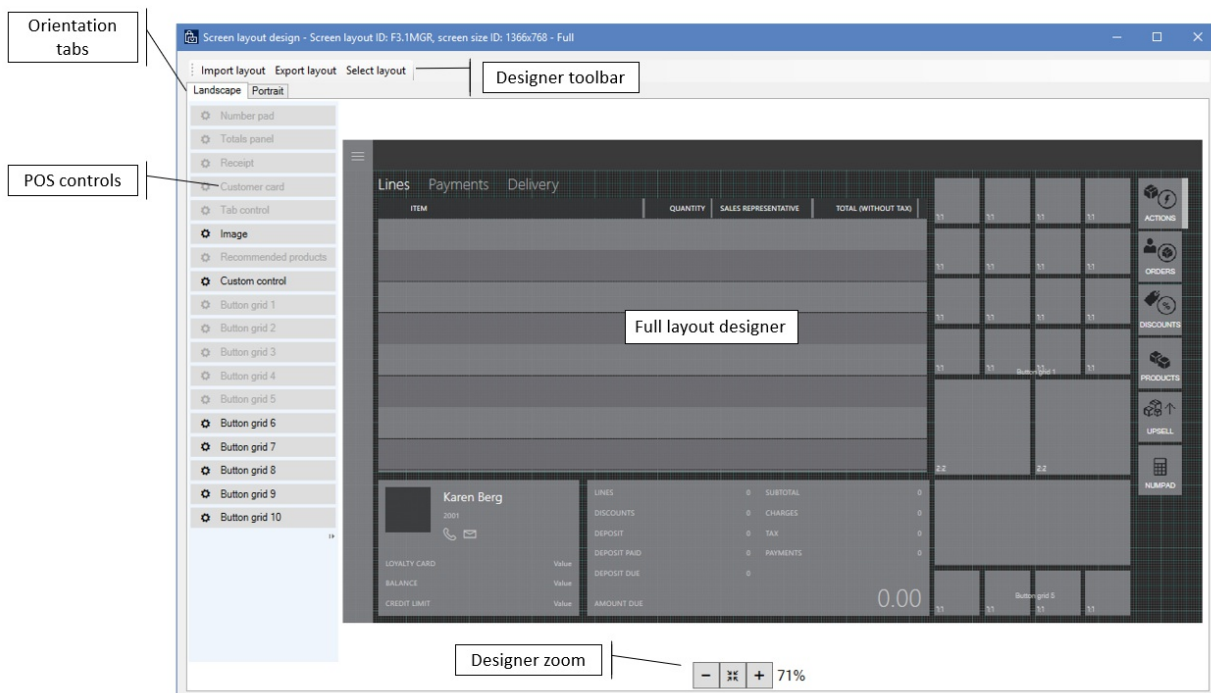
The screen layout designer lets you configure various aspects of the POS **Transaction** screen for each layout size, in both portrait and landscape modes, and for both Full and Compact layouts. The screen layout designer uses the ClickOnce deployment technology to download, install, and start the latest version of the application every time that users access it. Be sure to check the browser requirements for ClickOnce. Some browsers, such as Google Chrome, require extensions.

### IMPORTANT

You must configure a screen layout for each layout size that is defined and that is used by the POS.

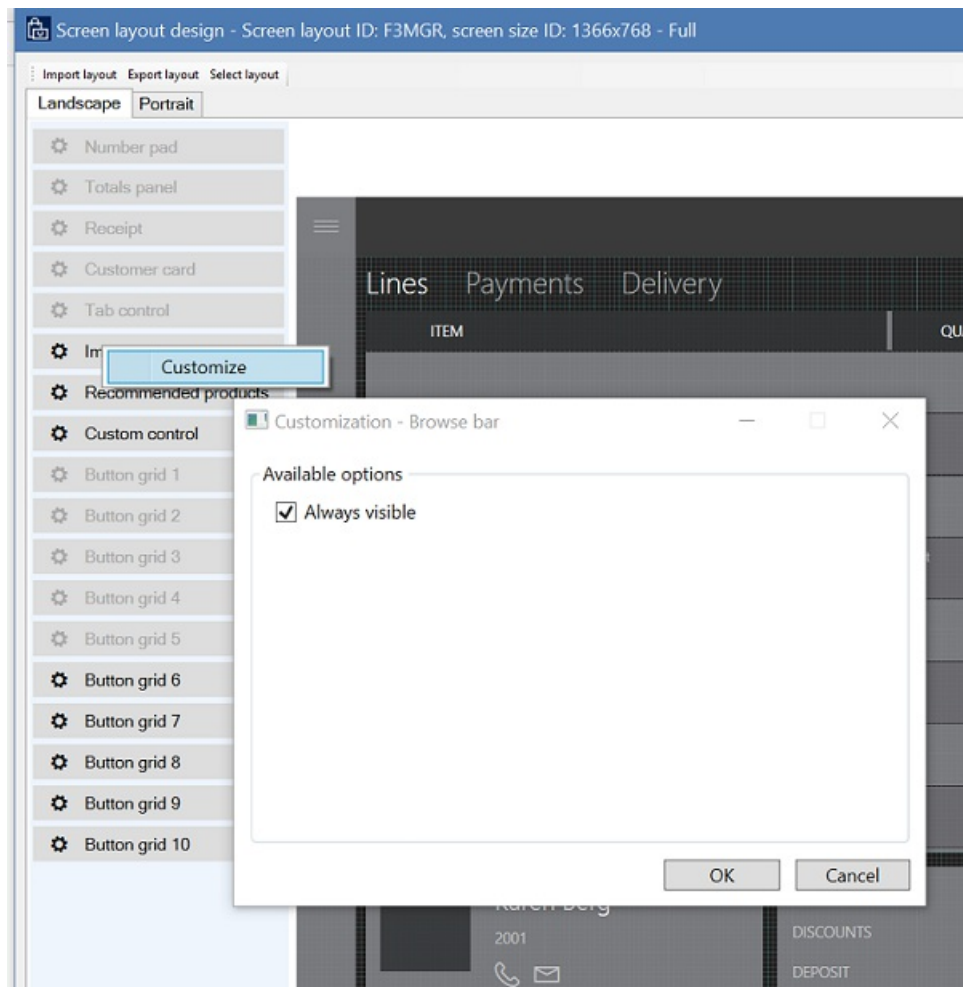
## Full layout designer

The Full layout designer lets users drag UI controls onto the POS **Transaction** screen and configure the settings of those controls.

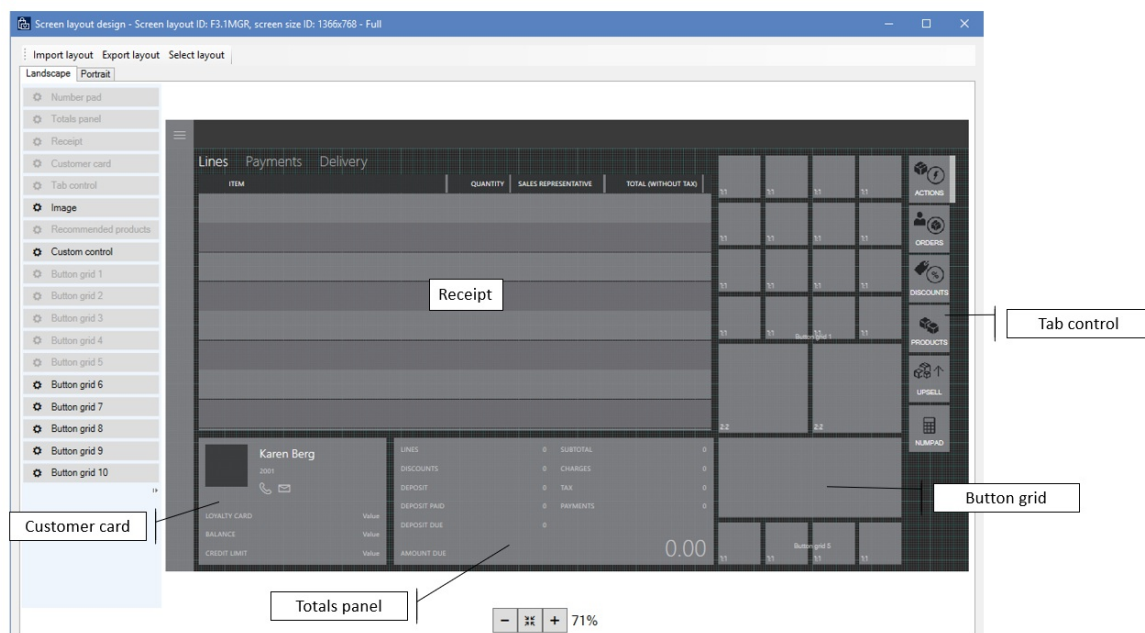


- **Import layout/Export layout** – You can export and import POS screen layout designs as XML files, so that you can easily reuse and share them across environments. It's important that you import layout designs for the correct layout sizes. Otherwise, UI elements might not fit correctly on the screen.
- **Landscape/Portrait** – If the POS device lets users switch between landscape and portrait modes, you must define a screen layout for each mode. The POS automatically detects screen rotation and shows the correct layout.
- **Layout grid** – The POS layout designer uses a 4-pixel grid. UI controls "snap" to the grid to help you correctly align the content.
- **Designer zoom** – You can zoom the designer view in and out to better view the content on the POS screen. This feature is useful when the screen resolution on the POS differs greatly from the resolution of the screen that is used in the designer.
- **Show/hide navigation bar** – For Full POS layouts, you can select whether the left navigation bar is visible on the **Transaction** screen. This feature is helpful for displays that have a lower resolution. To set

the visibility, right-click the navigation bar in the designer, and select or clear the **Always visible** check box. If the navigation bar is hidden, POS users can still access it by using the menu in the upper left.



- **POS controls** – The POS layout designer supports the following controls. You can configure many controls by right-clicking and using the shortcut menu.



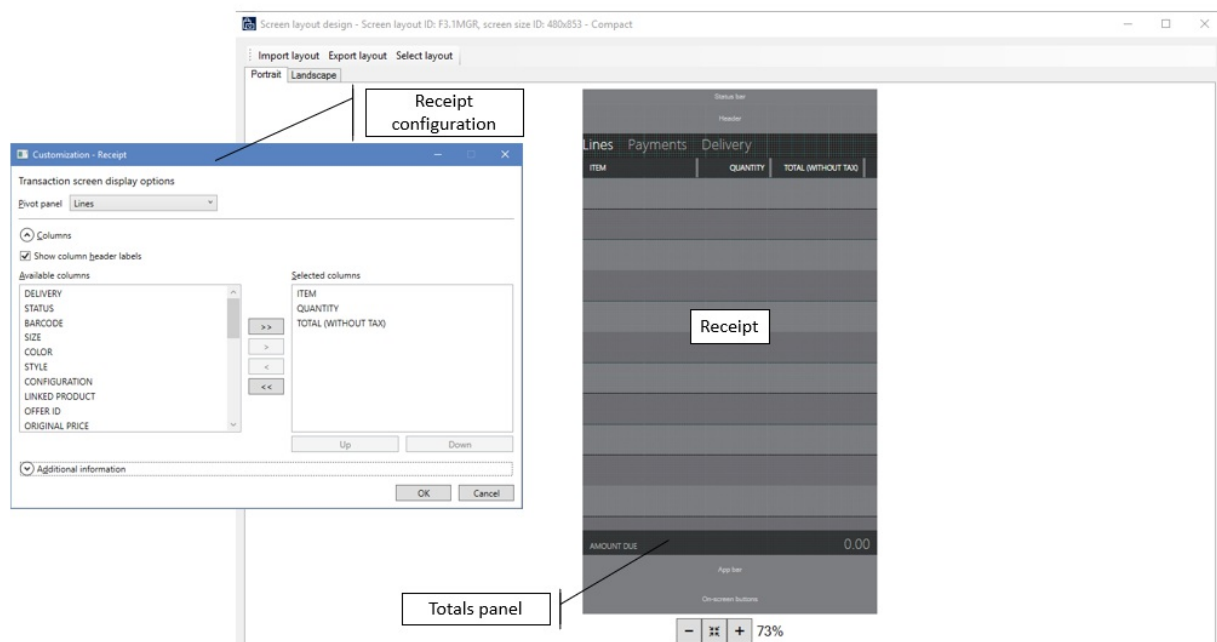
- **Number pad** – The number pad is the main mechanism for user input on the POS **Transaction** screen. You can configure the control so that the full number pad is shown. This option is ideal for touchscreen devices. Alternatively, you can configure it so that only the input field is shown. In this case, a physical keyboard is used for input. The number pad settings are available only for Full layouts.

For Compact layouts, the full number pad is always shown on the **Transaction** screen.

- **Totals panel** – You can configure the totals panel in either one column or two columns, to show values such as the line count, discount amount, charges, subtotal, and tax. Compact layouts support only a single column.
- **Receipt panel** – The receipt panel contains the sales lines, payment lines, and delivery information for the products and services that are processed in the POS. You can specify columns, widths, and placement. In Compact layouts, you can also configure additional information that appears in the row under the main line.
- **Customer card** – The customer card shows information about the customer who is associated with the current transaction. You can configure the customer card to hide or show additional information.
- **Tab control** – You can add the tab control to a screen layout, and then put other controls, such as the number pad, customer card, or button grids, in it. The tab control is a container that helps you fit more content on the screen. The tab control is available only for Full layouts.
- **Image** – You can use the image control to show the store's logo or another branding image on the **Transaction** screen. The image control is available only for Full layouts.
- **Recommended products** – If the recommended products control is configured for the environment, it shows product suggestions, based on machine learning.
- **Custom control** – The custom control acts as a placeholder in the screen layout and lets you reserve space for custom content. The custom control is available only for Full layouts.

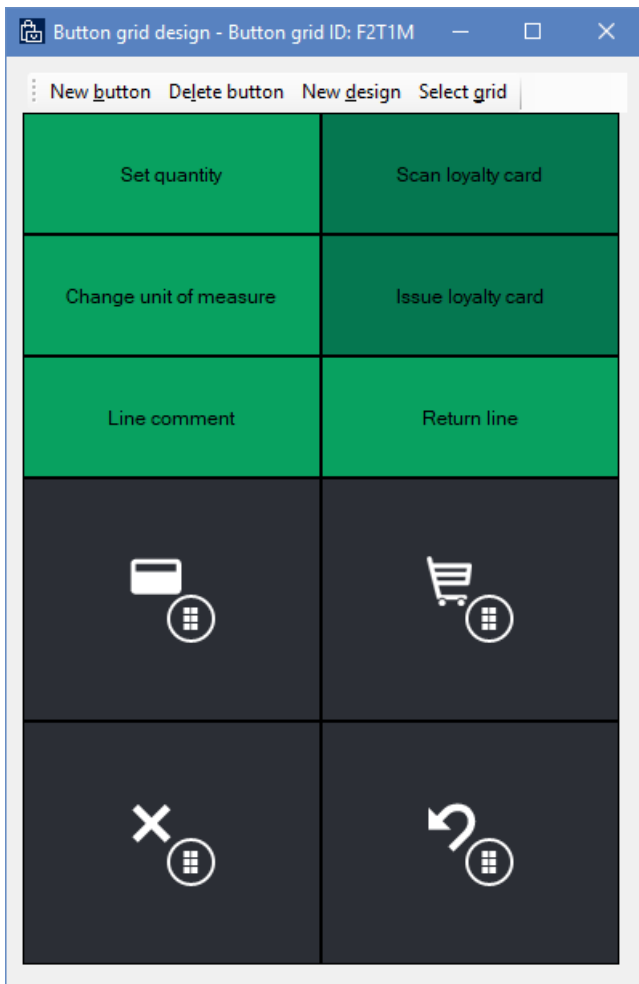
## Compact layout designer

Like the Full layout designer, the Compact layout designer lets you configure the POS screen layout for phones and small tablets. However, in this case, the layout itself is fixed. You can configure the controls in the layout by right-clicking and using the shortcut menu. However, you can't use drag-and-drop operations for additional content.



## Button grid designer

The button grid designer lets you configure button grids that can be used on the POS welcome screen and **Transaction** screen for both Full and Compact layouts. The same button grid can be used across layouts and layout types. Like the screen layout designer, the button grid designer uses the ClickOnce deployment technology to download, install, and start the latest version of the application every time that users access it. Be sure to check the browser requirements for ClickOnce. Some browsers, such as Google Chrome, require extensions.



- **New button** – Click to add a new button to the button grid. By default, new buttons appear in the upper-left corner of the grid. However, you can arrange buttons by dragging them in the layout.

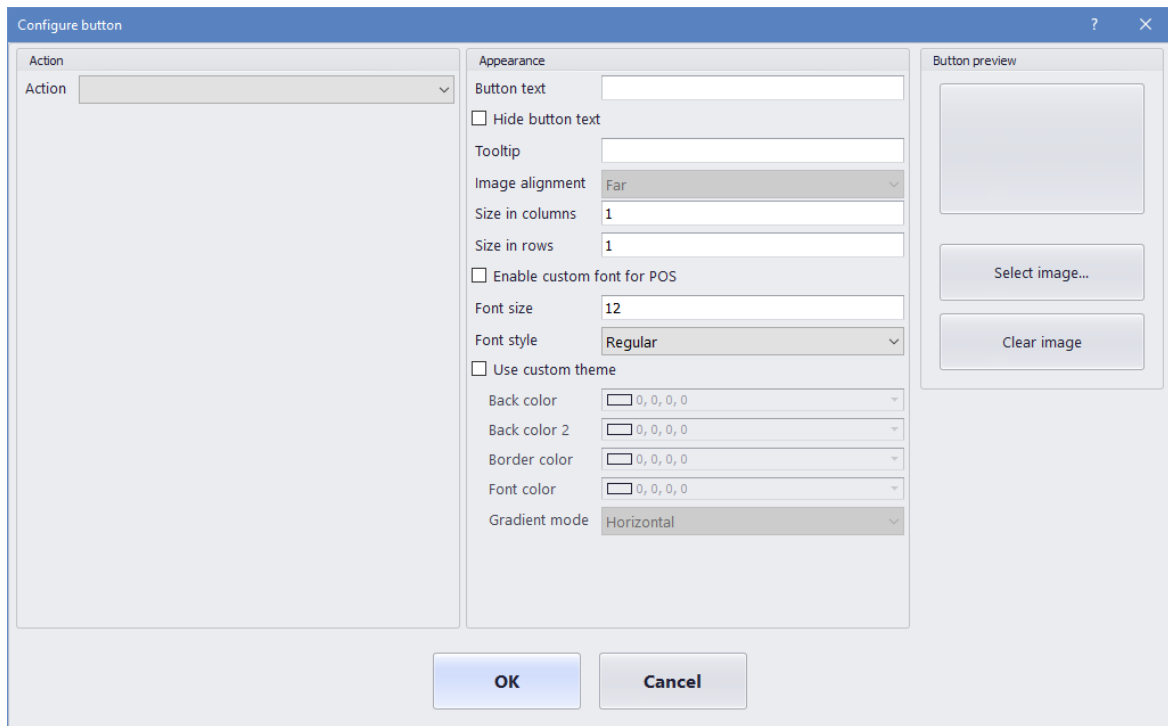
**IMPORTANT**

The contents of the button grid can overlap. When you arrange buttons, make sure that they don't hide other buttons.

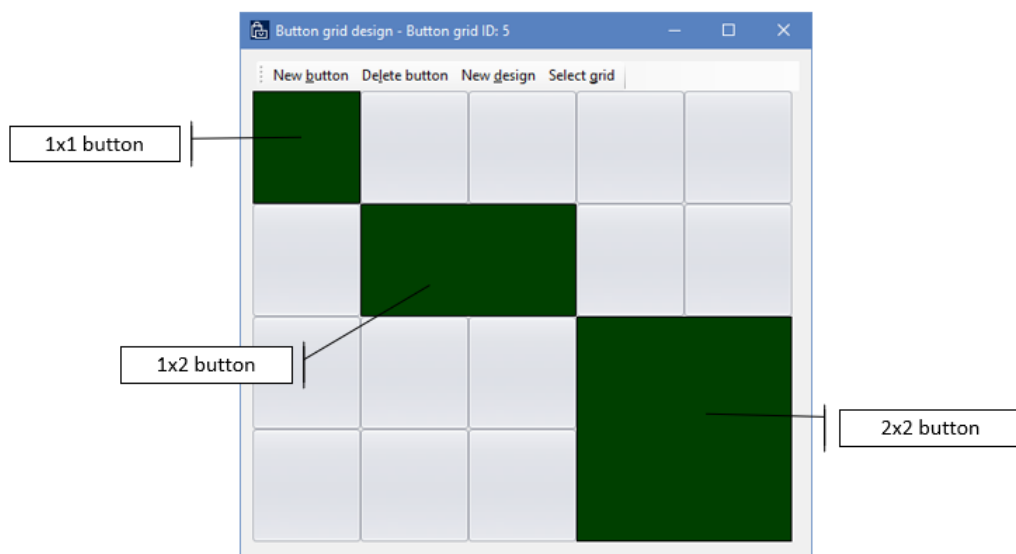
- **New design** – Click to automatically set up a button grid layout by specifying the number of buttons per row and column.
- **Button properties** – You can configure button properties by right-clicking the button and using the shortcut menu.

**IMPORTANT**

Some button grid settings apply only to Enterprise POS, not to Modern POS or Cloud POS.



- **Action** – In the list of applicable POS operations, select the operation that is invoked when the button is clicked in the POS.  
For the list of supported POS operations, see [Online and offline point of sale \(POS\) operations](#).
- **Action parameters** – Some POS operations use additional parameters when they are invoked. For example, for the Add product operation, users can specify the product to add.
- **Button text** – Specify the text that appears on the button in the POS.
- **Hide button text** – Use this check box to hide or show the button text. Button text is often hidden for small buttons that show only an icon.
- **Tooltip** – Specify additional Help text that appears when users mouse over the button.
- **Size in columns/Size in rows** – You can specify how tall and wide the button is.



- **Custom font** – When you select the **Enable custom font for POS** check box, you can specify a font other than the default system font for the POS.

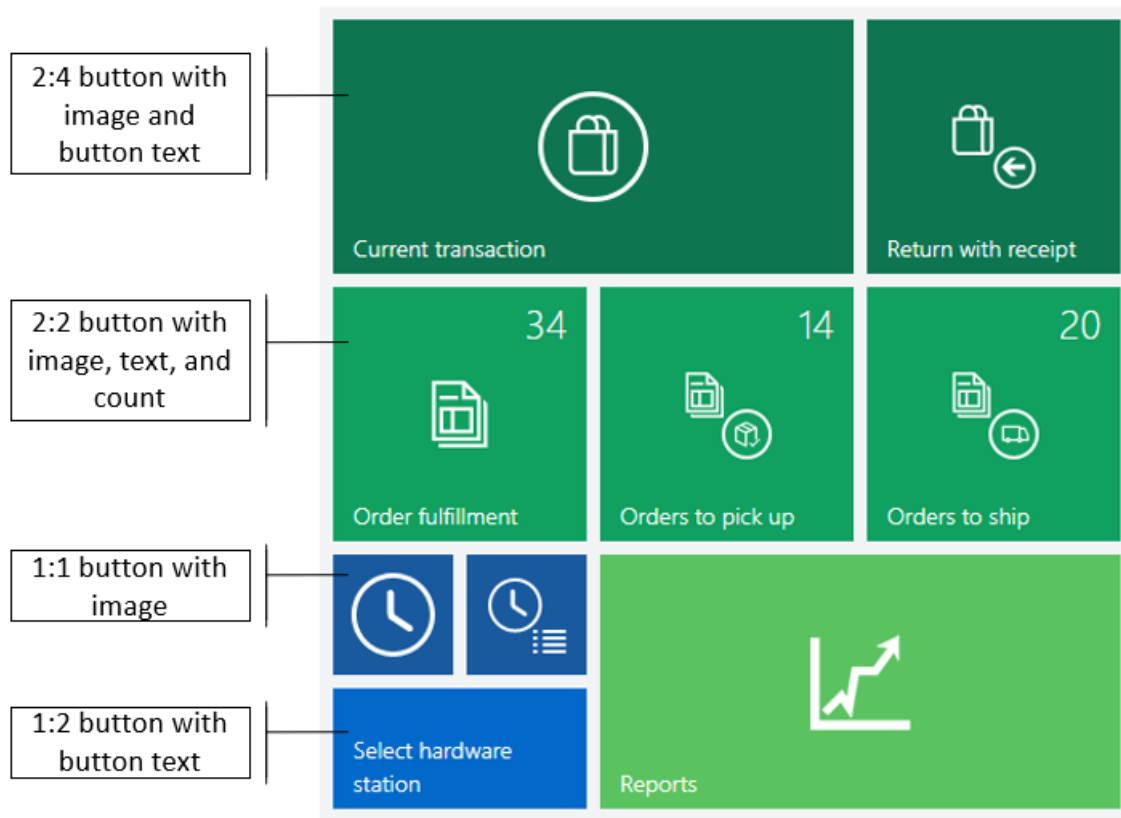


- **Custom theme** – By default, POS buttons use the accent color from the visual profile. When you select the **Use custom theme** check box, you can specify additional colors.

**NOTE**

Modern POS and Cloud POS use only the **Back color** and **Font color** values.

- **Button image** – Buttons can include images or icons. Select among the available images that are specified at **Retail and Commerce > Channel setup > POS setup > POS > Images**.



## Additional resources

[Install the Retail point of sale \(POS\) layout designer](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create an online functionality profile

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic presents an overview of setting up an online functionality profile for Microsoft Dynamics 365 Commerce.

## Overview

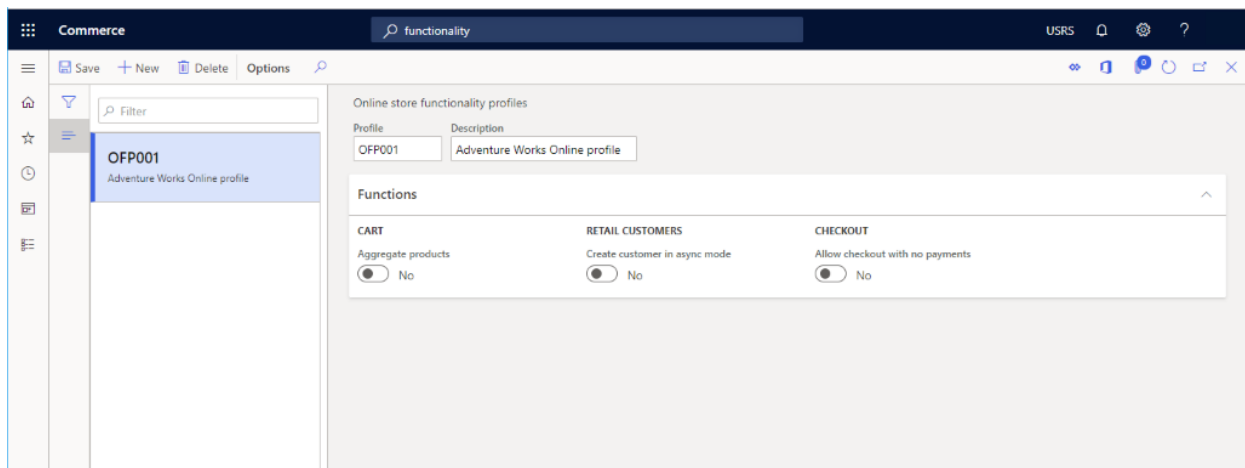
The online functionality profile provides various settings used for online channels. Each online channel must specify an online functionality profile.

## Create an online functionality profile

The following procedure explains how to create an online functionality profile from within Commerce Headquarters app.

1. In the navigation pane, go to **Modules > Channel setup > Online store setup > Functionality profiles**.
2. On the action pane, select **New**.
3. In the **Profile** field, enter an ID for the profile.
4. In the **Description** field, enter a value ("Adventure Works Profile" in the example image below).
5. In the **Functions** section, modify the **CART**, **RETAIL CUSTOMERS**, or **CHECKOUT** settings, as needed.
6. On the action pane, select **Save**.

The following image shows an example online functionality profile.



## Functions

- **Aggregate products:** When enabled, this function allows the cart to update quantity when the same item is added multiple times.
- **Allow checkout with no payments:** When enabled, this function handles the scenario when items added to cart have a price \$0.00.
- **Create customer in async mode:** This is a legacy setting applicable to third-party e-Commerce channels and is not applicable to the Dynamics 365 e-Commerce site.

## Additional resources

[Channels overview](#)

[Channel setup prerequisites](#)

[Set up an online channel](#)

[Set up a retail channel](#)

[Set up a call center channel](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up a retail channel

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to create a new retail channel in Microsoft Dynamics 365 Commerce.

## Overview

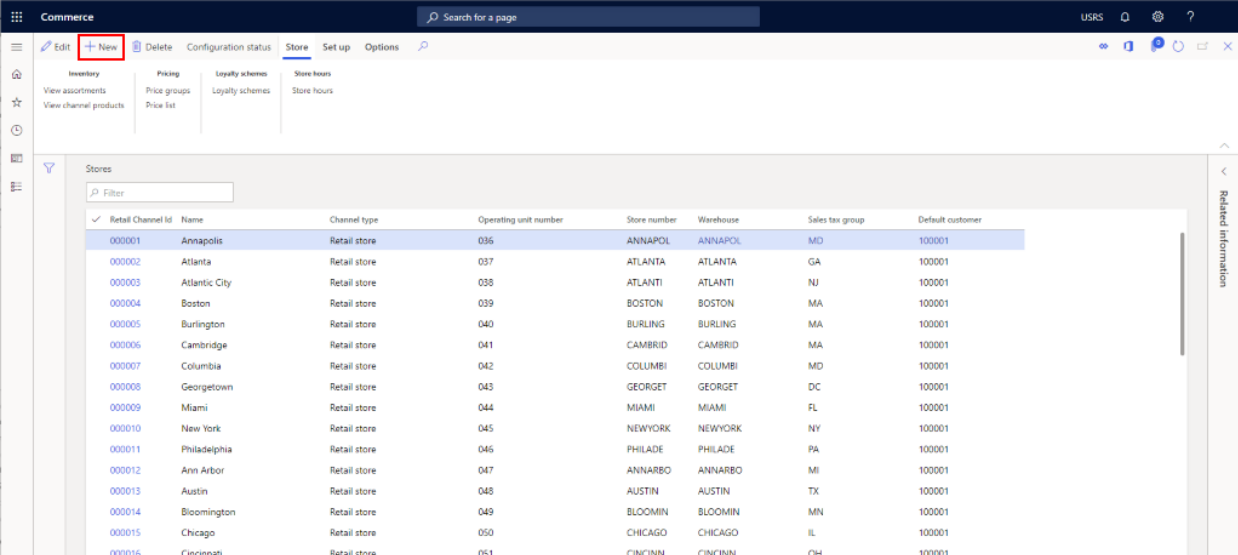
Dynamics 365 Commerce supports multiple retail channels. These retail channels include online stores, call centers, and retail stores (also known as brick-and-mortar stores). Each retail store channel can have its own payment methods, price groups, point of sale (POS) registers, income accounts and expense accounts, and staff. You must set up all of these elements before you can create a retail store channel.

Before a retail channel is created, ensure you follow the [channel prerequisites](#).

## Create and configure a new retail channel

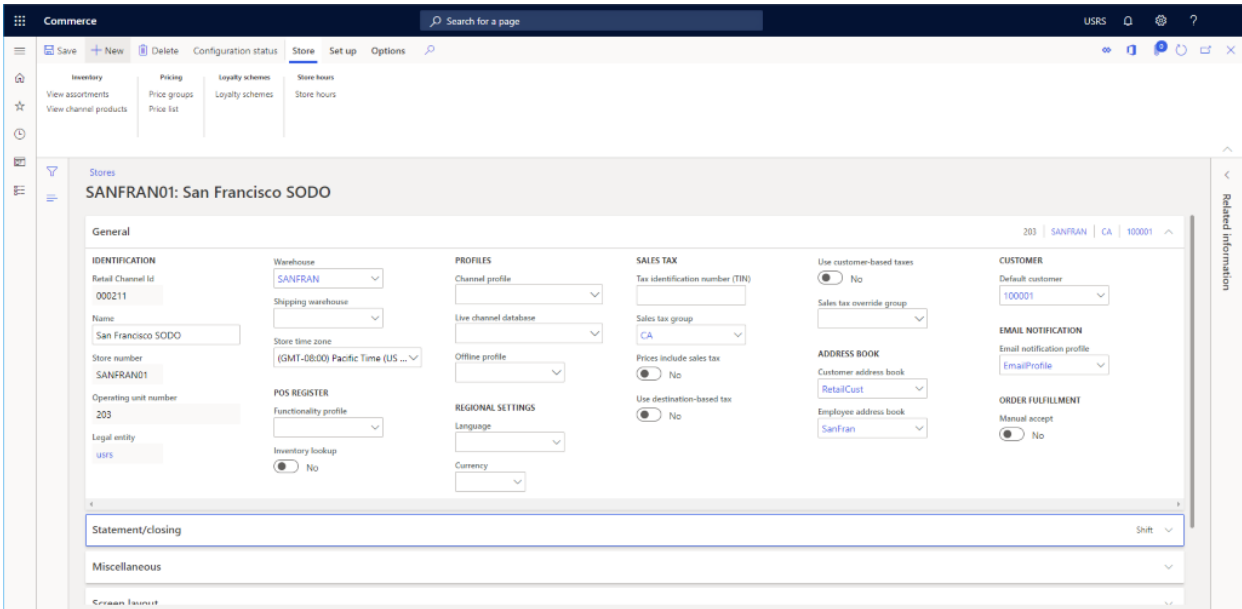
1. In the navigation pane, go to **Modules > Channels > Stores > All stores**.
2. On the action pane, select **New**.
3. In the **Name** field, provide a name for the new channel.
4. In the **Store number** field, provide a unique store number. The number can be alphanumeric with a maximum of 10 characters.
5. In the **Legal entity** drop-down list, enter the appropriate legal entity.
6. In the **Warehouse** drop-down list, enter the appropriate warehouse.
7. In the **Store time zone** field, select the appropriate time zone.
8. In the **Sales tax group** drop-down list, select an appropriate sales tax group for the store.
9. In the **Currency** field, select the appropriate currency.
10. In the **Customer address book** field, provide a valid address book.
11. In the **Default customer** field, provide a valid default customer.
12. In the **Functionality profile** field, select a functionality profile if applicable.
13. In the **Email notification profile** field, provide a valid email notification profile.
14. On the action pane, select **Save**.

The following image shows the creation of a new retail channel.



Retail Channel Id	Name	Channel type	Operating unit number	Store number	Warehouse	Sales tax group	Default customer
000001	Annapolis	Retail store	036	ANNAPOL	ANNAPOL	MD	100001
000002	Atlanta	Retail store	037	ATLANTA	ATLANTA	GA	100001
000003	Atlantic City	Retail store	038	ATLANTI	ATLANTI	NJ	100001
000004	Boston	Retail store	039	BOSTON	BOSTON	MA	100001
000005	Burlington	Retail store	040	BURLING	BURLING	MA	100001
000006	Cambridge	Retail store	041	CAMBRID	CAMBRID	MA	100001
000007	Columbia	Retail store	042	COLUMBI	COLUMBI	MD	100001
000008	Georgetown	Retail store	043	GEORGET	GEORGET	DC	100001
000009	Miami	Retail store	044	MIAMI	MIAMI	FL	100001
000010	New York	Retail store	045	NEWYORK	NEWYORK	NY	100001
000011	Philadelphia	Retail store	046	PHILADE	PHILADE	PA	100001
000012	Ann Arbor	Retail store	047	ANNARBO	ANNARBO	MI	100001
000013	Austin	Retail store	048	AUSTIN	AUSTIN	TX	100001
000014	Bloomington	Retail store	049	BLOOMIN	BLOOMIN	MN	100001
000015	Chicago	Retail store	050	CHICAGO	CHICAGO	IL	100001
000016	Cincinnati	Retail store	051	CINCINN	CINCINN	OH	100001

The following image shows an example retail channel.

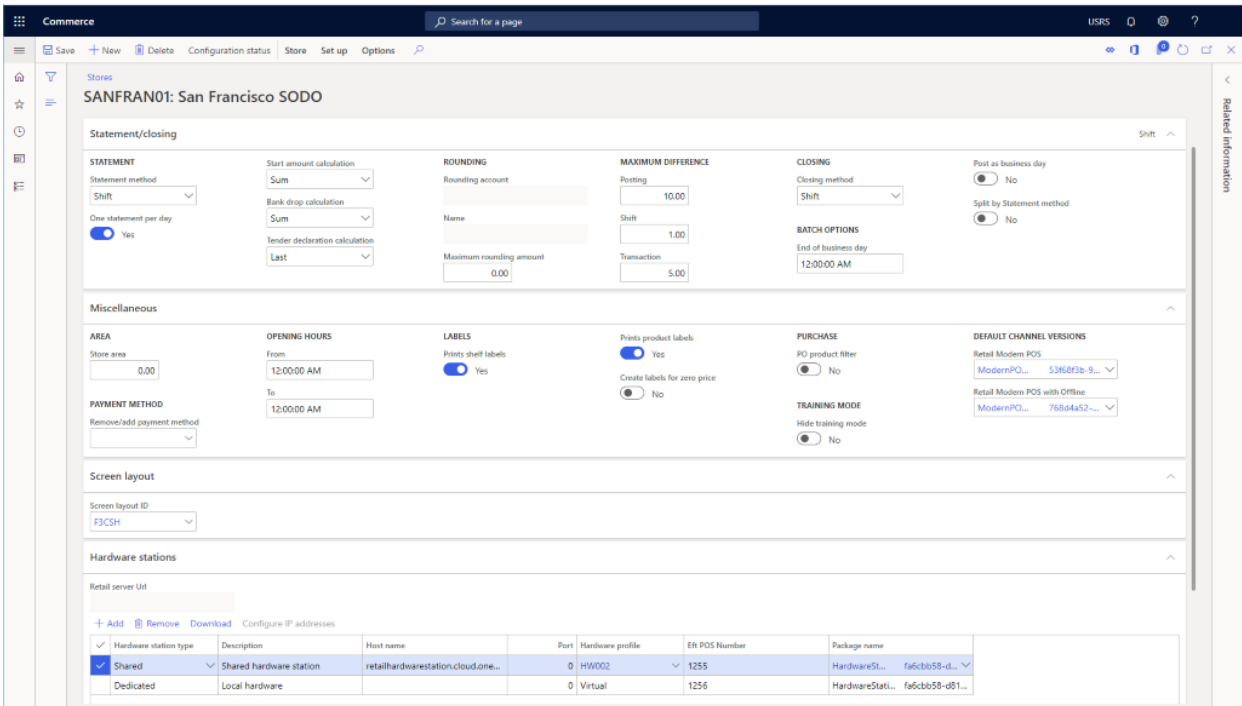


## Other settings

There are numerous other optional settings that can be set in the **Statement/closing** and **Miscellaneous** sections, based on the needs of the retail store.

In addition, see [Screen layouts for the point of sale \(POS\)](#) for information on setting up the default screen layout in the **Screen layout** section and [Configure and install Retail hardware station](#) for setup information about the **Hardware stations** section.

The following image shows an example retail channel setup configuration.



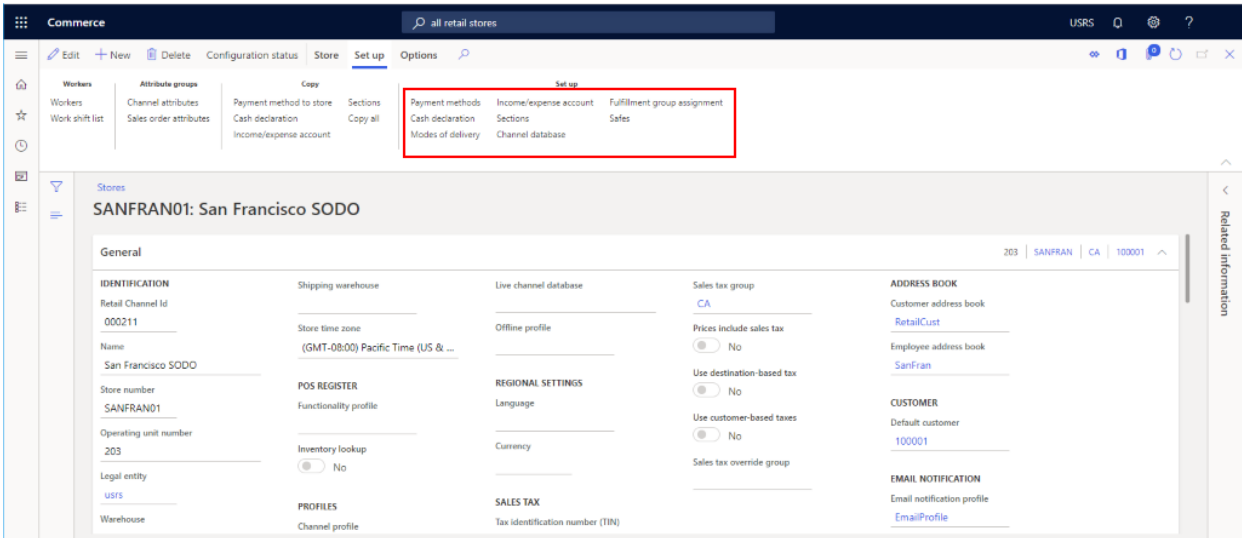
## Additional channel set up

There are additional items that need to be set up for a channel that can be found on the **Action** pane under the **Set up** section.

Additional tasks required for online channel setup include setting up payment methods, cash declaration, modes

of delivery, income/expense account, sections, the fulfillment group assignment, and safes.

The following image shows various additional retail channel setup options on the **Set up** tab.

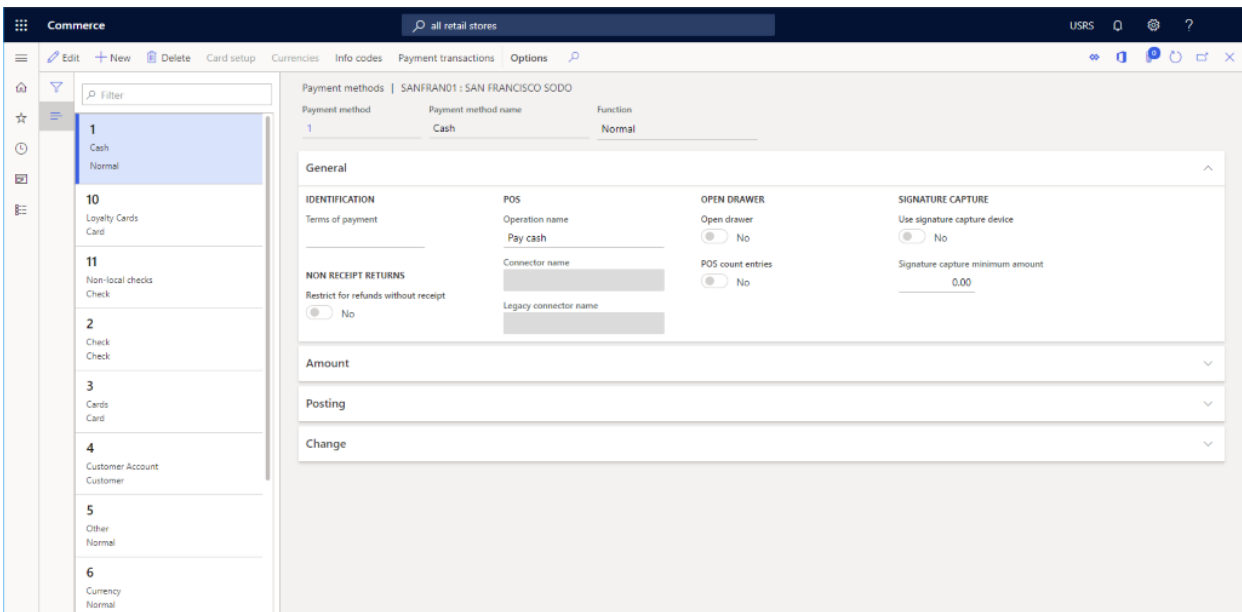


### Set up payment methods

To set up payment methods, for each payment type supported on this channel follow these steps.

1. On the action pane, select the **Set Up** tab, then select **Payment methods**.
2. On the action pane, select **New**.
3. In the navigation pane, select a desired payment method.
4. In the **General** section, provide an **Operation name** and configure any other desired settings.
5. Configure any additional settings as required for the payment type.
6. On the action pane, select **Save**.

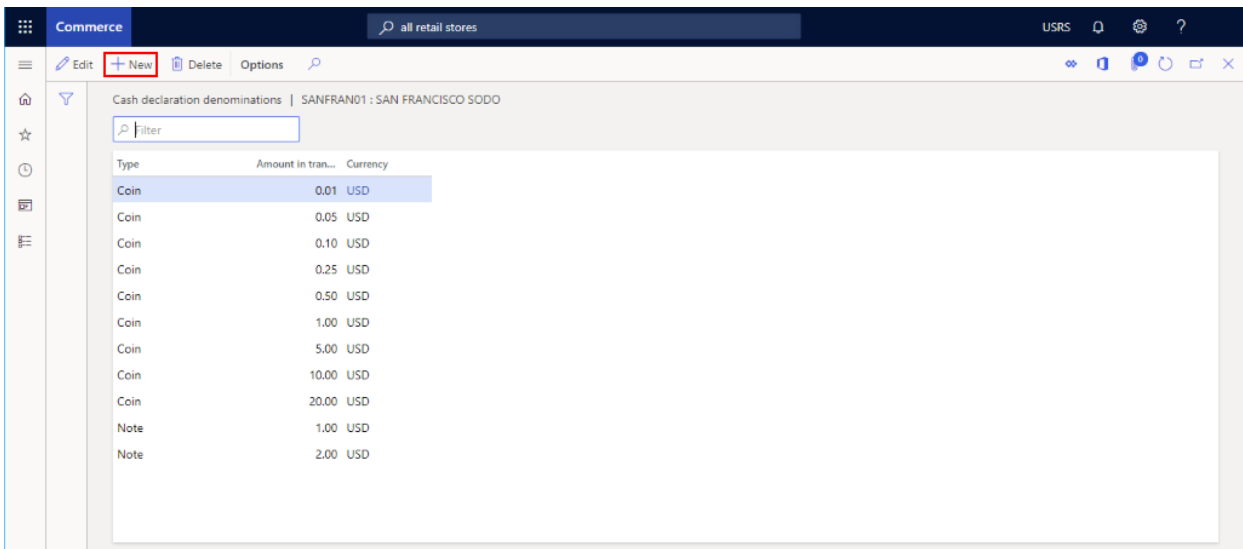
The following image shows an example of a cash payment method.



### Set up cash declaration

1. On the action pane, select the **Set Up** tab, and then select **Cash declaration**.
2. On the action pane, select **New**, and then create all **Coin** and **Note** denominations that are applicable.

The following image shows an example of a cash declaration.



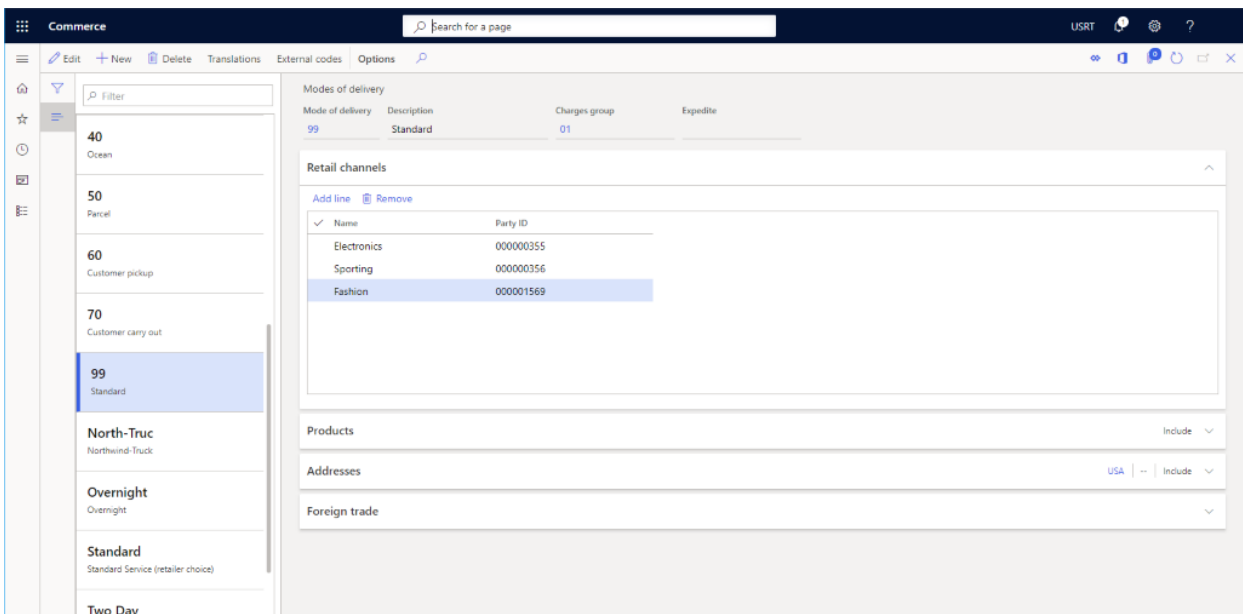
## Set up modes of delivery

You can see the configured modes of delivery by selecting **Modes of delivery** from the **Set up** tab on the **Action** pane.

To change or add a mode of delivery, follow these steps.

1. In the navigation pane, go to **Modules > Inventory management > Modes of delivery**.
2. On the action pane, select **New** to create a new mode of delivery, or select an existing mode.
3. In the **Retail channels** section, select **Add line** to add the channel. Adding channels using organization nodes instead of adding each channel individually can streamline adding channels.

The following image shows an example of a mode of delivery.



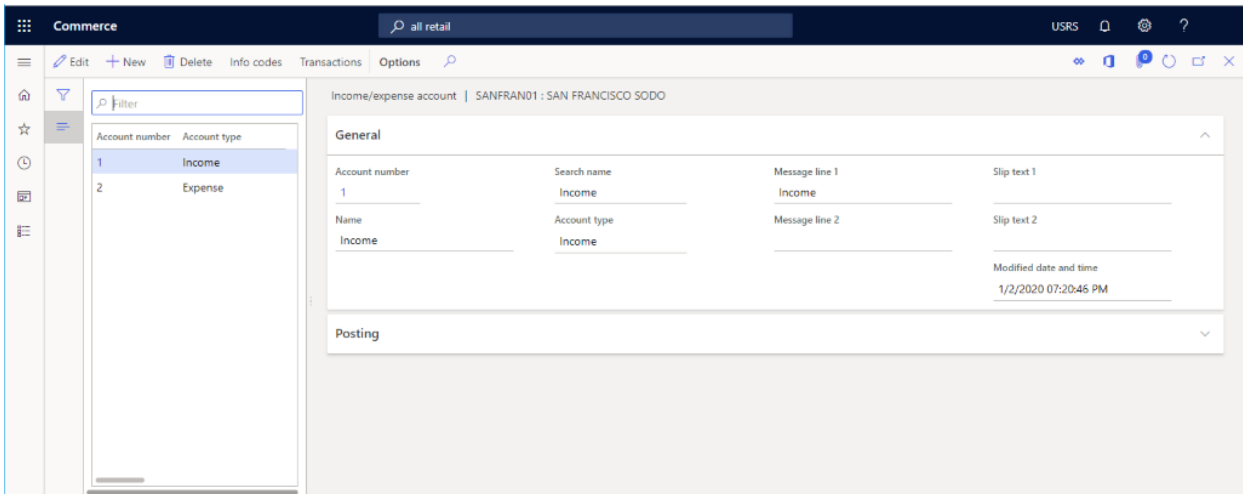
## Set up income/expense account

To set up income/expense account, follow these steps.

1. On the action pane, select the **Set Up** tab, and then select **Income/Expense account**.
2. On the action pane, select **New**.
3. Under **Name**, enter a name.
4. Under **Search name**, enter a search name.
5. Under **Account type**, enter the account type.
6. Enter text for **Message line 1**, **Message line 2**, **Slip text 1**, and **Slip text 2** as needed.

- Under **Posting**, enter posting information.
- On the action pane, select **Save**.

The following image shows an example of an income/expense account.



### Set up sections

To set up sections, follow these steps.

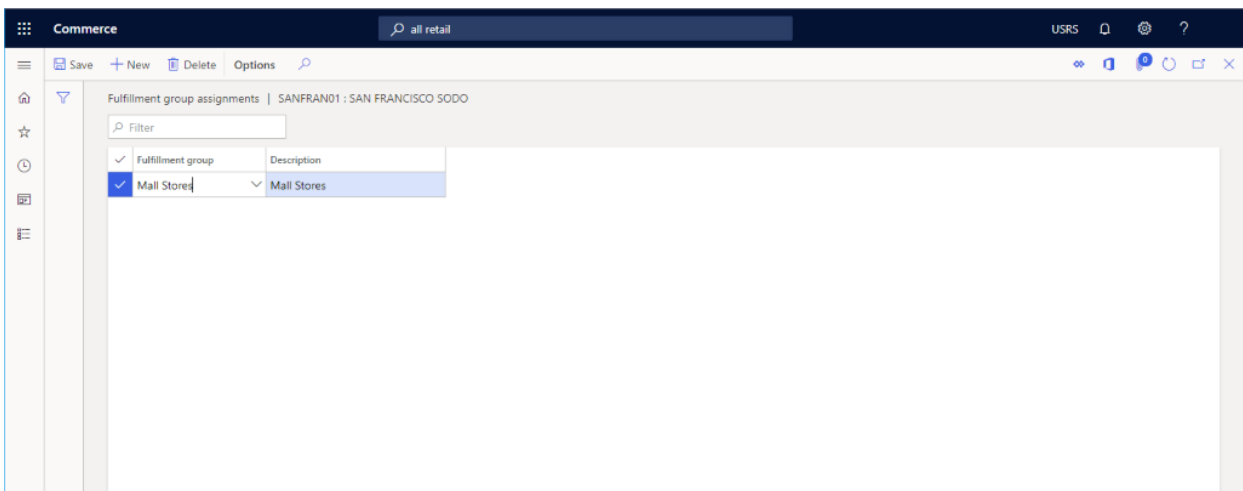
- On the action pane, select the **Set Up** tab and click **Sections**.
- On the action pane, select **New**.
- Under **Section number**, enter a section number.
- Under **Description**, enter a description.
- Under **Section size**, enter a section size.
- Configure additional settings for **General** and **Sales statistics** as needed.
- On the action pane, select **Save**.

### Set up a fulfillment group assignment

To set up a fulfillment group assignment, follow these steps.

- On the action pane, select the **Set up** tab, then select **Fulfillment group assignment**.
- On the action pane, select **New**.
- In the **Fulfillment group** drop-down list, select a fulfillment group.
- In the **Description** drop-down list, enter a description.
- On the action pane, select **Save**.

The following image shows an example of a fulfillment group assignment setup.





## Set up safes

To set up safes, follow these steps.

1. On the action pane, select the **Set Up** tab and click **Safes**.
2. On the action pane, select **New**.
3. Enter a name for the safe.
4. On the action pane, select **Save**.

## Additional resources

[Channels overview](#)

[Channel setup prerequisites](#)

[Set up an online channel](#)

[Set up a call center channel](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure a worker

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates how to configure a worker as a sales representative who is eligible for commission on sales in POS. This procedure uses the USRT demo data company.

## Create a commission sales group for the worker

1. Go to Sales and marketing > Commissions > Sales groups.
  - Workers can be assigned to one or more sales groups. In POS, you can choose any sales group that contains workers from the store's address book.
2. Click New.
3. In the Group field, type a value.
4. In the Name field, type a value.
5. Click Save.
6. On the Action Pane, click General.
7. Click Sales rep.
  - A sales group can contain more than one worker. Commissions can be split between workers based on how you define the commission share.
8. In the Name field, enter or select a value.
9. In the Commission share field, enter a number.
10. Click Save.
11. Close the page.
12. Close the page.

## Assign the workers default sales group

1. Go to Retail and Commerce > Employees > Workers.
2. In the list, find and select the desired record.
3. In the list, click the link in the selected row.
4. Click the Commerce tab.
  - A worker can be assigned to a default sales group. The default sales group will be automatically added to sales lines in POS if the option is enabled in the functionality profile for the store.
5. Click Edit.
6. In the Default group field, enter or select a value.
7. Click Save.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Choose between Modern POS (MPOS) and Cloud POS

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic gives implementers additional background, tips, and guidance for factors that they should consider when they deploy Dynamics 365 Commerce. By reviewing and following this guidance as part of the deployment process, implementers can avoid issues that might affect user satisfaction or performance.

## Insights

Commerce provides a wide range of deployment and topology options. Therefore, retailers can choose the components and configuration that best meet their business and technology requirements. One aspect of implementation that requires careful consideration is the choice of a platform and form factor for the point of sale (POS) component.

### POS platform and form factor considerations

Commerce supports the following POS options:

- Modern POS (MPOS) for Microsoft Windows
- MPOS for Microsoft Windows Phone
- MPOS for Apple iPad or Google Android tablet
- Cloud POS (CPOS), which supports the Microsoft Edge, Internet Explorer, and Google Chrome browsers

In all cases, the POS (MPOS and CPOS) shares the same core application code. This point is important for the following reasons:

- The user interface (UI) is consistent, regardless of the platform or form factor.
- Most of the functional capabilities are the same, regardless of the platform or form factor. However, there are some important differences. These differences are noted in this topic.
- In a given store, the POS variations can be combined and can run concurrently. For example, for its main registers, a retailer can use MPOS on computers that run Windows. However, the retailer can supplement those registers with browser-based terminals or mobile devices.
- Customizations and extensions can easily be used across platforms and form factors. Because the core application code is shared, most customizations can be implemented one time instead of multiple times.

### MPOS vs. CPOS

Although MPOS and CPOS are largely the same, there are some important differences that you must understand.

#### MPOS

MPOS on a Windows, iOS, or Android device is an application that is packaged, installed, and serviced on that device.

- **Windows** – The MPOS for Windows application contains all the application code and the embedded commerce runtime (CRT).
- **iOS/Android** – On these platforms, the application acts as a host for the CPOS application code. In other words, the application code comes from the CPOS server on Microsoft Azure or the Commerce Scale Unit. For more information, see [Commerce Scale Unit overview](#).

#### CPOS

Because CPOS runs in a browser, the application isn't installed on the device. Instead, the browser accesses the application code from the CPOS server. Therefore, CPOS can't directly access POS hardware or work in an offline state.

### Store deployment considerations

In addition to a platform and form factor, retailers must also choose a deployment option at the store. The following table shows the configurations that are available for each POS option.

POS APPLICATION	COMMERCE SCALE UNIT	AVAILABLE OFFLINE
MPOS for Windows	Cloud or RSSU	Yes
MPOS for iOS or Android	Cloud or RSSU	No
Cloud POS	Cloud or RSSU	No

#### Commerce Scale Unit

The Commerce Scale Unit is a component that hosts the CRT. The CRT contains all the business logic that the POS uses, and it provides access to the channel database. While they are online, all POS clients in the store use the Commerce Scale Unit. The Commerce Scale Unit can be deployed either in the cloud or in the store.

#### Offline mode

MPOS for Windows supports offline mode. In offline mode, the POS can continue to process sales even if it's disconnected from the Commerce Scale Unit. It can then be synchronized with the channel database when connectivity is restored. MPOS uses its own embedded instance of the CRT and temporarily uses its own local data source (offline SQL Server database). For more information about offline functionality, see [POS offline functionality](#).

#### POS peripheral/hardware considerations

Retailers must also consider how the POS will access devices and peripherals such as printers, cash drawers, and payment terminals. Only MPOS for Windows supports direct communication with these devices. MPOS for Windows Phone, iOS, or Android, and Cloud POS require a hardware station in order to access these devices. Hardware stations can be dedicated to a POS register or shared among the registers in a store. For more information about hardware stations, see [Configure and install Retail hardware station](#).

## Implementation considerations

Consider the following information as you plan your POS implementation in your stores:

- **Functional requirements** – The core business processes and capabilities are the same, regardless of the platform, form factor, or deployment topology. Therefore, most retailers don't have to consider functional requirements when they plan their implementation.
- **Connectivity** – Network availability (wide area network [WAN] and local area network [LAN]) is a major factor that requires careful consideration. Any benefits that a zero-footprint, cloud-hosted solution brings in terms of cost and simplicity are lost if the system isn't available for business-critical processes.

Unless the connectivity for a given device is very dependable and resilient, or unless a certain amount of downtime is acceptable to the retailer, we recommend one of the following options:

- Use MPOS in Windows, and enable offline mode.
- Deploy an on-premises Commerce Scale Unit.

These two options aren't mutually exclusive. For the most reliable topology, retailers can deploy a local RSSU to reduce the dependency on internet connectivity or Azure availability, and they can also deploy POS registers where offline mode is enabled if there is an issue with the local server or network.

- **Hardware devices/peripherals** – One important aspect of a Retail POS system is its ability to use POS peripherals such as printers, cash drawers, and payment terminals. Although all the available POS options can use peripheral devices, only MPOS for Windows supports them directly. For all other applications, one or more hardware stations are required. Although this approach adds flexibility, additional components must be deployed, configured, and serviced.
- **System requirements** – The system requirements for the POS application vary. Be sure to check the latest information before you make your choice. For example, because CPOS runs in a browser, it supports a wider range of operating systems. For more information about system requirements, see [System requirements for cloud deployments](#).
- **Deployment and servicing** – The complexity of the deployment and servicing requirements can vary, depending on the application and deployment choices. For example, for a cloud-hosted CPOS deployment, you don't have to install and update on every device. Therefore, this approach greatly reduces complexity and cost. However, if you deploy MPOS on every register and enable offline mode, and you also deploy shared hardware stations, you greatly increase the number of endpoints that must be managed.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Online and offline point of sale (POS) operations

2/18/2021 • 18 minutes to read • [Edit Online](#)

Most actions that user take in the point of sale (POS) are considered operations. Operations are configured and managed in the Dynamics 365 Commerce back office. Many operations can be added to buttons in the POS button grid. Users can then select the buttons to invoke the operations and perform their function. Other operations are part of the main POS application, and are invoked either from on-screen buttons or as part of other workflows or processes.

The following table provides details about the operations that are available in Modern POS and Cloud POS. The table also specifies where in the application the operations can be invoked, and whether they are available when the POS is in offline mode.

Some operations aren't currently available in Modern POS or Cloud POS. Some of these operations are either locale-specific operations that require additional extensions and configuration. Others are features from Microsoft Dynamics AX 2012 that aren't currently supported.

The following columns specify where the operations can be invoked:

- **Button grid** – The operation can be assigned to buttons in POS button grids, which are part of a POS screen layout.
- **Transaction screen** – The operation can be invoked from POS button grids that are configured on the POS transaction screen.
- **Welcome screen** – The operation can be invoked from POS button grids that are configured on the POS welcome screen.

## NOTE

The operations listed below apply to the latest version of Commerce. Some operations may have changed or may not be available in previous versions.

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
707	Activate device	Activate the current device by allowing an authenticated user to provide connection information and assign a device and register ID.	No	No	No	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
134	Add affiliation	Add a preselected affiliation to a transaction. Select the affiliation on the <b>Button properties</b> page.	Yes	Yes	No	Yes	No
135	Add affiliation from list	Add an affiliation to a transaction by selecting it in a list.	Yes	Yes	Yes	Yes	No
137	Add affiliation to customer	Add an affiliation to a customer on the <b>Customer details</b> page.	No	No	No	Yes	No
138	Remove affiliation from customer	Remove an affiliation on the <b>Customer details</b> page.	No	No	No	Yes	No
643	Add coupon code	Add a coupon by entering its code in the POS.	Yes	Yes	No	Yes	No
141	Add header charges	Add a misc charge to the order header.	Yes	Yes	No	No	No
141	Add line charges	Add a misc charge to a selected sales line.	Yes	Yes	No	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
117	Add loyalty card	Prompt the user to enter a loyalty card number that will be added to the current transaction.	Yes	Yes	No	Yes	No
136	Add serial number	This operation lets the user specify a serial number for the currently selected product.	Yes	Yes	No	Yes	No
1214	Add shipping address	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
519	Add to gift card	Add money to the specified gift card.	Yes	Yes	No	No	No
6000	Allow skip fiscal registration	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes
1212	Bank drop	Record the amount of money that is sent to the bank and other information , such as the number of the bank bag.	Yes	Yes	Yes	Yes	No
923	Bank totals verification	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes



ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
915	Blank operation	This operation represents a customizable button that a software developer can programmatically change for any specialized operation that the business requires.	Yes	Yes	Yes	Yes	No
1053	Blind close shift	Set the current shift to blind closed, and sign the user out. A blind-closed shift is closed to additional transactions but is still open to drawer operations, such as tender removal and tender declaration.	Yes	Yes	Yes	No	No
310	Calculate total	When discount calculation is delayed, this operation initiates the calculation for the current transaction.	Yes	Yes	No	Yes	No
642	Carry Out All Products	Set the mode of delivery for all lines to <b>Carryout</b> .	Yes	Yes	No	Yes*	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
641	Carry Out Selected Products	Set the mode of delivery for the selected lines to <b>Carryout</b> .	Yes	Yes	No	Yes*	No
647	Change mode of delivery	Change mode of delivery for preconfigured shipping sales lines.	Yes	Yes	No	No	No
1215	Change password	This operation lets the POS user change his or her password.	Yes	Yes	Yes	No	No
123	Change unit of measure	Change the unit of measure for the selected line item.	Yes	Yes	No	Yes	No
639	Clear default sales representative on transaction	Remove the commission sales group (sales rep) from the transaction.	Yes	Yes	No	Yes	No
106	Clear quantity	Reset the quantity on the currently selected line to 1.	Yes	Yes	No	Yes	No
640	Clear sales representative on line	Remove the commission sales group (sale rep) from the currently selected line.	Yes	Yes	No	Yes	No
121	Clear salesperson	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
1055	Close shift	Close the current shift, print a Z report, and sign the user out of the system.	Yes	Yes	Yes	No	No
139	Conclude transaction	Prompts user to select payment method	Yes	Yes	No	Yes	No
620	Create customer order	Convert the POS transaction to a customer order.	Yes	Yes	No	Yes*	No
925	Copy the bank check	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes
620	Create customer order	Convert the POS transaction to a customer order.	Yes	Yes	No	Yes*	No
621	Create quotation	Convert the POS transaction to a sales quotation.	Yes	Yes	No	Yes*	No
636	Create retail transaction	This operation lets the user create a standard sales transaction when the default POS behavior is to create customer orders.	Yes	Yes	No	Yes	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
600	Customer	Add the specified customer to the transaction.	No	No	No	Yes	No
1100	Customer account deposit	Make a payment to a customer's account.	Yes	Yes	Yes	Yes	Yes
612	Customer add	This operation lets the user create a new customer record.	Yes	Yes	Yes	Yes†	No
603	Customer clear	Remove the customer from the current transaction.	Yes	Yes	No	Yes	No
602	Customer search	This operation lets the user search for a customer record by navigating to the customer search page in the POS.	Yes	Yes	Yes	Yes	No
609	Customer transactions	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
917	Database connection status	This operation lets the user view the current connection settings, and switch between online and offline modes.	Yes	Yes	Yes	Yes	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
1200	Declare start amount	Declare the amount that is in the cash drawer when the day or shift starts.	Yes	Yes	Yes	Yes	No
132	Deposit override	Override the default deposit for customer orders.	Yes	Yes	No	Yes*	No
913	Design mode disable	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
912	Design mode enable	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
1217	Disassemble kits	Disassemble a kit into its component products.	Yes	Yes	Yes	Yes	No
624	Display refund amounts	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes
513	Display total	Show the balance of the transaction on the customer display.	Yes	Yes	Yes	Yes	No
623	Edit customer	Edit the current customer's details.	Yes	Yes	No	No	No
614	Edit customer order	Recall the selected order so that it can be modified in the POS.	No	No	No	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
615	Edit quotation	Recall the selected quotation so that it can be modified in the POS.	No	No	No	No	No
518	Expense accounts	Record money that is removed from the cash drawer for occasional expenses.	Yes	Yes	Yes	Yes	No
919	Extended log on	Assign or remove permission to sign in by scanning a bar code or swiping a card.	Yes	Yes	Yes	Yes	No
1201	Float entry	This operation lets the user add additional money to the current drawer or shift.	Yes	Yes	Yes	Yes	No
1218	Force unlock peripheral	The system uses this operation internally to unlock POS peripherals.	Not applicable	Not applicable	Not applicable	Not applicable	No
520	Gift card balance	Show the balance of a gift card.	Yes	Yes	No	No	No
708	Inactivate device	Inactivate the current device, so that it can't be used as a POS register.	No	No	No	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
804	Inbound operation	Access the features of inbound store inventory management.	Yes	No	Yes	No	No
517	Income accounts	Record money that is put into the cash drawer for a reason other than a sale.	Yes	Yes	Yes	Yes	No
801	Inventory lookup	Look up available, on order, and available-to-promise (ATP) quantities for the current store and other available locations.	Yes	Yes	Yes	No	No
122	Invoice comment	This operation lets the user enter a comment about the current transaction.	Yes	Yes	No	Yes	No
511	Issue credit memo	Issue a credit memo to provide a voucher instead of a refund.	Yes	Yes	No	No	No
512	Issue gift card	Issue a new gift card for the specified amount.	Yes	Yes	No	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
625	Issue loyalty card	Issue a loyalty card to a customer, so that the customer can participate in the store's loyalty program.	Yes	Yes	Yes	No	No
300	Line discount amount	Enter a discount amount for a line item in the transaction. This operation is used only for discountable items and only within specified discount limits.	Yes	Yes	No	Yes	No
301	Line discount percent	Enter a discount percentage for a line item in the transaction. This operation is used only for discountable items and only within specified discount limits.	Yes	Yes	No	Yes	No
703	Lock register	Lock the current register, so that it can't be used, but don't sign the current user out.	No	No	No	Yes	No



ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
701	Log off	Sign the current user out of the register.	Yes	Yes	Yes	Yes	No
521	Loyalty card points balance	Show the balance of points for the specified loyalty card.	Yes	Yes	No	No	No
142	Manage charges	View and manage misc charges applied to transaction.	Yes	Yes	No	No	No
918	Manage shifts	Show a list of active, suspended, and blind closed shifts.	Yes	Yes	Yes	No	No
914	Minimize POS window	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
1000	Open drawer	Perform a "no sale" operation, and open the currently selected cash drawer.	Yes	Yes	Yes	Yes	No
928	Order fulfillment	This operation allows users to pick, pack, ship, or recall orders for store picked up.	Yes	Yes	Yes	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
805	Outbound operation	Access features for managing shipments of outbound transfer orders.	Yes	No	Yes	No	No
129	Override line product tax	Override the tax on the selected line item, and use a different specified tax.	Yes	Yes	No	Yes	No
130	Override line product tax from list	Override the tax on the selected line item, and use the tax that the user selects in a list.	Yes	Yes	No	Yes	No
127	Override transaction tax	Override the tax on the transaction, and use a different specified tax.	Yes	Yes	No	Yes	No
128	Override transaction tax from list	Override the tax on the transaction, and use the tax that the user selects in a list.	Yes	Yes	No	Yes	No
131	Packing slip	Create a packing slip for the selected order.	No	No	No	No	No
710	Pair hardware station	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
201	Pay card	Accept a card such as a credit card or a debit card as payment.	Yes	Yes	No	Yes	No
200	Pay cash	Accept cash as payment.	Yes	Yes	No	Yes	No
206	Pay cash quick	Complete the transaction in one touch, and accept the amount that is due in cash (exact change).	Yes	Yes	No	Yes	No
204	Pay check	Accept a check as payment.	Yes	Yes	No	Yes	No
213	Pay credit memo	Accept a credit memo (voucher) that the store issued.	Yes	Yes	No	No	No
203	Pay currency	Accept payment in various currencies.	Yes	Yes	No	Yes	No
202	Pay customer account	Charge the transaction to the customer's account. This payment method isn't valid for customer order deposits.	Yes	Yes	No	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
214	Pay gift card	Accept a gift card that the store issued.	Yes	Yes	No	No	No
207	Pay loyalty	Accept a loyalty card for payment, and redeem points toward qualified products.	Yes	Yes	No	No	No
634	Payments history	Show the customer's payment history for the current customer order.	Yes	Yes	No	No	No
803	Picking and receiving	Open the <b>Picking and receiving</b> page, where you can select orders to pick or receive in the store.	Yes	Yes	Yes	No	No
632	Pickup all products	Set the fulfillment method to <b>Store pickup</b> for all lines.	Yes	Yes	No	Yes*	No
631	Pickup selected products	Set the fulfillment method to <b>Store pickup</b> for selected lines.	Yes	Yes	No	Yes*	No
400	Popup menu	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
101	Price check	This operation lets the user look up the price for a specified product.	Yes	Yes	Yes	Yes	No
104	Price override	Override the price of a product, if the product has been set up to allow for price overrides.	Yes	Yes	No	Yes	No
1058	Print fiscal X	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes
1059	Print fiscal Z	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes
927	Print item label	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
926	Print shelf label	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
1056	Print X	Print and X report for the current shift.	Yes	Yes	Yes	No	No
103	Product comment	Add a comment to the selected line item in the transaction.	Yes	Yes	No	Yes	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
100	Product sale	Add a specified product to the transaction.	Yes	Yes	Yes	Yes	No
108	Product search	This operation lets the user search for a product by navigating to the product search page in the POS.	Yes	Yes	Yes	Yes	No
633	Quote expiration date	This operation lets the user view or modify the expiration date on a sales quotation.	Yes	Yes	No	Yes*	No
627	Recalculate	Recalculate all customer order lines and taxes, based on the current configuration.	Yes	Yes	No	Yes*	No
143	Recalculate charges	Recalculate the auto-charges applied to the order.	Yes	Yes	No	No	No
515	Recall order	This operation lets the user search for and recall customer orders and sales quotations.	Yes	Yes	Yes	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
504	Recall transaction	This operation lets the user recall a previously suspended transaction from the current store.	Yes	Yes	No	Yes‡	No
305	Redeem loyalty points	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes
635	Refund shipping charges	This operation lets the user refund shipping charges on a canceled order.	No	No	No	No	No
644	Remove coupon code	Prompt the user to remove coupons by selecting them in a list of coupons that are currently associated with the transaction.	Yes	Yes	No	Yes	No
1057	Reprint Z	Reprint the Z report for the previous shift or a selected shift.	Yes	Yes	Yes	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
1216	Reset password	This operation lets a user who has the password-reset permission reset another employee's password by using a temporary password.	Yes	Yes	Yes	No	No
1219	Open URL in POS	This operation lets a user to open an admin configured URL in POS.	Yes	Yes	Yes	Yes	No
109	Return product	Perform a return of individual products. The next scanned product is shown as a returned product that has a negative quantity and price.	Yes	Yes	No	Yes	No
114	Return transaction	Recall a previous transaction by its receipt number to return some or all of the products.	Yes	Yes	Yes	Yes§	No
1211	Safe drop	Perform a safe drop to move money from the register to a safe.	Yes	Yes	Yes	Yes	No



ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
516	Sales invoice	This operation lets the customer make payments toward the selected sales invoice.	Yes	Yes	No	No	No
502	Salesperson	This operation lets the user set the <b>Sales taker</b> value on a sales order for customer orders in the POS.	Yes	Yes	No	Yes*	No
2000	Schedule management	This operation is not yet supported.	Yes	Yes	Yes	No	No
2001	Schedule requests	This operation is not yet supported.	Yes	Yes	Yes	No	No
622	Search	This operation lets users preconfigure POS buttons to perform searches by item, customer, or category.	Yes	Yes	Yes	Yes	No
1213	Search shipping address	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
709	Select hardware station	This operation lets the user select a hardware station in a list of available hardware stations.	Yes	Yes	Yes	Yes	No
637	Set default sales representative on transaction	This operation lets the user select one of the eligible commission sales groups (sale reps) as the default sales rep for lines that are added later.	Yes	Yes	No	Yes	No
105	Set quantity	Change the quantity of a line item in the transaction.	Yes	Yes	No	Yes	No
638	Set sales representative on line	This operation lets the user select one of the eligible commission sales groups (sale reps) for the currently selected line.	Yes	Yes	No	Yes	No
630	Ship all products	Set the fulfillment mode to <b>Shipping</b> for all line items.	Yes	Yes	No	Yes*	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
629	Ship selected products	Set the fulfillment mode to <b>Shipping</b> for the selected lines.	Yes	Yes	No	Yes*	No
115	Show journal	Show the store's journal. You can view transactions, reprint receipts and gift receipts, and recall for return.	Yes	Yes	Yes	Yes**	No
802	Stock count	This operation lets the user create or modify stock counting journals for physical inventory or cycle counts.	Yes	Yes	Yes	No	No
401	Sub menu	This operation takes the user to another linked button grid.	Yes	Yes	Yes	Yes	No
1054	Suspend shift	Suspend the current shift, so that a new or different shift can be activated on the current register.	Yes	Yes	Yes	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
503	Suspend transaction	Suspend the current sales transaction, so that it can be recalled later in the store.	Yes	Yes	No	Yes‡	No
1004	Task recorder	Open Task recorder to record procedural steps in the POS.	No	No	No	Yes	No
1052	Tender declaration	This operation lets the user specify the amount of money in the drawer for each counted payment method.	Yes	Yes	Yes	Yes	No
1210	Tender removal	This operation lets the user remove money from the current drawer or shift.	Yes	Yes	Yes	Yes	No
920	Time clock	This operation lets users punch in and punch out of work shifts and breaks.	Yes	Yes	Yes	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
302	Total discount amount	Enter a discount amount for the transaction. This operation applies only to discountable items and only within specified discount limits.	Yes	Yes	No	Yes	No
303	Total discount percent	Enter a discount percentage for the transaction. This operation applies only to discountable items and only within specified discount limits.	Yes	Yes	No	Yes	No
501	Transaction comment	Add a comment to the current transaction.	Yes	Yes	No	Yes	No
922	View product details	Open the product details page for the currently selected line item.	Yes	Yes	No	Yes	No
1003	View reports	Show the reports that have been configured for the current user.	Yes	Yes	Yes	No	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
921	View time clock entries	Show the time clock entries for all workers at the store.	Yes	Yes	Yes	No	No
211	Void payment	Void the currently selected payment line from the transaction.	Yes	Yes	No	Yes	No
102	Void product	Void the currently selected line item from the transaction.	Yes	Yes	No	Yes	No
500	Void transaction	Void the current transaction.	Yes	Yes	No	Yes	No
916	Windows workflow foundation	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	No
924	X report for bank cards	This operation isn't supported.	Not applicable	Not applicable	Not applicable	Not applicable	Yes
311	Remove system discounts from transactions	Remove all the system applied discounts, including coupon based discounts, from the transaction. This does not remove manual discounts.	Yes	Yes	Yes	Yes	No

ID	OPERATION	DESCRIPTION	BUTTON GRID	TRANSACTION SCREEN	WELCOME SCREEN	AVAILABLE OFFLINE	LOCALE-SPECIFIC
312	Reapply system discounts	Reapply system discounts on the transaction if they were removed using the <b>Remove system discounts from transaction</b> operation.	Yes	Yes	Yes	Yes	No

\* The operation is available in offline mode only when a customer order or sales quotation is being created, and only if offline creation of customer orders and sales quotations is configured in the POS functionality profile. The operation can't be performed when orders are created by using Real-time Service, or when orders are recalled or edited.

† The operation can be performed in offline mode only when the POS is configured to allow for offline creation of customers in the POS functionality profile.

‡ When the POS is offline, suspended transactions can be recalled only from the current register's offline database. Users can't suspend and recall transactions across registers.

§ When the POS is offline, only transactions in the current offline database can be recalled for return.

\*\* When the POS is offline, only transactions in the current offline channel database are shown in the journal.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Device activation of a customized Modern POS

2/18/2021 • 10 minutes to read • [Edit Online](#)

This topic explains how to configure Microsoft Dynamics 365 Commerce Headquarters so that device activation works correctly when a customized Modern POS application is used. It describes the steps that are required in order to obtain the customized reply address and enter that value in Headquarters.

Modern POS is a client-side component for Microsoft Dynamics 365 Commerce. To use the POS, you must perform device activation. Device activation uses Microsoft Azure Active Directory (Azure AD) to authenticate users. Enhanced functionality in this area has modified the device activation flow to take better advantage of the Web Account Manager service. As part of this enhancement, there is now enhanced security for the authentication approval process. This enhanced security requires additional configuration in Headquarters when the POS is customized, because a specific, unique value is now required for the callback URI. (The callback URI is also known as the redirect URI.)

By default, Modern POS is already registered for this callback URI. However, when customized, the callback URI is changed. Therefore, it must be correctly configured so that it works again. This topic describes the steps that you must follow to complete this configuration. If this configuration isn't completed, you receive an error message when you try to perform device activation in the customized POS application. This error message resembles the following example:

```
AADSTS50011: The reply address 'ms-appx-web://Microsoft.AAD.BrokerPlugin/[...]' does not match the reply addresses configured for the application.
```

The reply address is dependent on the Modern POS package SID that is shown at the end of the above error message. This is a function of the Package Family Name (PFN), so it is dependent on a name and public key. This means that when you customize Modern POS and sign it with your company certificate, the SID will change. The change is based on the certificate and not the package version, so as long as the entire package is not different and the certificate remains the same (which can be renewed to keep the same signature), then the SID will remain the same.

## NOTE

We recommend that you try to use the customized Modern POS application one time before you configure Dynamics 365 headquarters. In this way, you can see what the error message looks like and more easily obtain the customized reply address.

## Setup

The following steps are required so that device activation works correctly when the customized Modern POS application is used. You will create two Azure AD applications: one for Modern POS and one for Commerce Scale Unit. The Commerce Scale Unit Azure AD application is required because the POS uses resources through Commerce Scale Unit. Therefore, both Azure AD applications are used when the POS is used. In this scenario, Commerce Scale Unit serves as the endpoint for protected resources that the POS requests.

### Create the Commerce Scale Unit Azure AD application

1. In a web browser, go to <https://portal.azure.com/>.
2. Sign in by using Azure AD credentials that have enough permission to create Azure AD applications.



3. Select **Azure Active Directory** from the list of Azure services, then select **App registrations** from the leftmost menu that appears.
4. Create the Commerce Scale Unit Azure AD application by selecting **New registration** and entering the following values:
  - **Name:** Enter **Customized Commerce Scale Unit**. You can enter any other unique value, but be sure to make a note of the name entered.
  - **Supported account types:** Select the value **Accounts in this organization directory only** unless this application registration will be used across multiple tenants, which would require the next value **Accounts in any organizational directory**. Note that this selection is not typical.
  - **Redirect URI:** This value can be left blank but you need to verify that the initial drop-down selection is set to **Web**.
5. Select **Register** at the bottom of the page. The page will change to the newly created Azure AD application.
6. Select **Application ID URI** where it has a link stating **Add an Application ID URI**. This page can also be reached from the left menu by selecting **Expose an API**.
7. On the tile that opens, select the **Application ID URI** value that states **Set**. Copy the value shown before selecting the **Save** button. You will paste this value into the DLLHost.exe.config file for POS in the next section.
8. Select **Save** to set this URI.
9. Select the **Add a scope** button.
10. In the slider that appears, enter the following values:
  - **Scope name:** Enter **AccessRetailServer**. You can enter any other unique value, but be sure to make a note of the name entered.
  - **Who can consent:** Select the **Admins and users** option.
  - **Admin consent display name:** Enter **AccessRetailServer**. For simplicity, make this value match the **Scope name** above.
  - **Admin consent description:** Enter **AccessRetailServer**. You can enter any value here. This is used noting the reason for the scope.
  - The fields **User consent display name** and **User consent description** will not be used.
  - **State:** Verify that **Enabled** is selected.
11. Select the **Add scope** button.
12. Select the **Overview** tab on the leftmost menu. Verify that the **Application ID URI** value matches what was copied back in step seven.

#### **NOTE**

Do not close the web browser window because you will use it again later in this topic.

### **Update the Modern POS configuration**

1. In File Explorer, go to **C:\Program Files (x86)\Microsoft Dynamics 365\70\Retail Modern POS\ClientBroker**. (This path assumes that the Microsoft Windows operating system on the computer is based on the x64 architecture.)
2. In File Explorer, select **File > Open Windows PowerShell > Open Windows PowerShell as administrator**.
3. In the Microsoft Windows PowerShell window that appears, enter **notepad DLLHost.exe.config**, and then

press the Enter key. (The Windows PowerShell window will already be pointed to the current file directory.)

4. In the Notepad window that appears, find the value that corresponds to the **AADRetailServerResourceId** key. (By default, this value is `https://commerce.dynamics.com`.) Then paste the App ID URI value that you copied in step 7 in the previous section.
5. Select **File > Save**.

#### NOTE

Don't close the Notepad window, because you will use it again in the next section. Another way to perform the above steps would be to use a script or a post-step installation customization.

### Create the customized Modern POS Azure AD application

1. Return to the web browser window where <https://portal.azure.com/> is open, and create the Retail Modern POS Azure AD application by repeating steps 3-5 in the "Create the Commerce Scale Unit Azure AD application" section. However, enter the following values this time:

- **Name:** Enter **Customized Retail Modern POS**. (You can enter any other unique value, but be sure to make a note of the name entered.)
- **Supported account types:** Select the value **Accounts in this organization directory only** unless this application registration will be used across multiple tenants, which would require the next value **Accounts in any organizational directory**. Note that this selection is not typical.
- **Redirect URI:** The initial drop-down value should be changed to **Public client/native (mobile & desktop)**. The value to enter for this type of Redirect URI is the value shown at the beginning of this topic in the Modern POS error message received. Enter the reply address (redirect URI) that corresponds to that error message. The value will start with **ms-appx-web://Microsoft.AAD.BrokerPlugin/[...]**.

#### NOTE

- You can also see the reply address (redirect URI) in Event Viewer in Windows, under **Microsoft-Dynamics-Commerce-ModernPos/Operational**. The event ID is 40619. The error message will start with the following text: "This UWP application was assigned the following callback URI to be used while interacting with Azure AD: ms-appx-web://Microsoft.AAD.BrokerPlugin/S-1-15-2-[...]"
- After the Azure AD application is created, you can specify additional redirect URIs as you require. If multiple packages that have different callback URIs have been generated for any reason, keep this single Azure AD application, and maintain all redirect URIs in it.

2. It is important to validate this value in the **Redirect URI** field. Then select **Create**, and wait until the operation is successfully completed. (If an error occurs, address it, and then try again.)

A tile appears that shows the details of the new customized Retail Modern POS Azure AD application.

3. Select **Register** at the bottom of the page. The page will change to the newly created Azure AD application.
4. Find the **Application (client) ID** field, and copy the value. Switch to the notepad you opened in the previous section (or follow the above section **Update the Modern POS configuration** to open the **DLLHost.exe.config** file) to navigate to the value corresponding to **AADClientId**. Paste in the value for this field that was copied at the beginning of this step and then save the file (as shown in step 5 of the same section).
5. Returning to the web browser, in the leftmost menu, select the **API permissions** option.

6. On the **API permissions** tile, select **+ Add a permission**.
7. On the slider that appears, first select the **My APIs** heading. Then select the API titled **Customized Commerce Scale Unit** or whatever value was entered as the name in step 4 at the beginning of this document.
8. Under the **Select permissions** sub-heading, select **AccessRetailServer** or whatever value was entered as the scope name in step 10 at the beginning of this document.
9. Select **Add permissions** at the bottom of the slider.
10. Select **Grant admin consent for <your AAD name>**. Select **Yes**. This grants consent and can be verified with the **Granted** displaying in the **Status** column in the **AccessRetailServer** row.

> [!NOTE]  
 > Granting consent is not required, but simplifies the process by consenting in advance for all users in your tenant (and you as the Admin). If this step is not completed, then each user will be asked for consent the first time that they try to activate Modern POS.

### Configure Dynamics 365 Headquarters

The previous steps were required so that the Modern POS application can be authenticated. You must now follow these steps to add the new Azure AD applications to the list of safe programs in Headquarters, so that the requests are authorized. (A list of safe programs is sometimes also referred to as a safe list.)

1. In a web browser, go to the Headquarters URL, and sign in by using Azure AD credentials.
2. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
3. On the **Identity Providers** tab, in the **Identity providers** section, select the provider that begins with `HTTPS://sts.windows.net/`. The values in the **Relying parties** section are updated, based on the provider that you selected.
4. In the **Relying parties** section, select **+ Add**, and enter the following values:
  - **ClientId**: Enter the value that you copied in step 3 in the previous section and then pasted into the `DLLHost.exe.config` file in step 13.
  - **Type**: Select **Public**.
  - **UserType**: Select **Worker**.
  - **Name**: Enter a description to help users understand what this entry references.
5. On the Action Pane, select **Save**. The relying party that you just created should remain selected.
6. In the **Server resource IDs** section, select **+ Add**, and enter the following values:
  - **Server Resource Id**: Enter the URL that you copied in step 4 in the "Update the Modern POS configuration" section. (You originally created this value in step 4 in the "Create the Commerce Scale Unit Azure AD application" section.)
  - **Name**: Enter a description to help users understand what this entry references.
7. On the Action Pane, select **Save**.
8. Go to **Retail and Commerce > Retail and CommerceIT > Distribution schedule**.
9. Select job **1110 (Global configuration)**, and then, on the Action Pane, select **Run now**. This job synchronizes the new data. However, there is a cache in Commerce Scale Unit that won't be updated for several minutes. Therefore, if you require an immediate update, the Commerce Scale Unit application pool must be recycled.

**NOTE**

For best results, verify that Modern POS is closed, and that no instances of DLLHost.exe exist in Task Manager.

**Perform Modern POS device activation**

Try to activate the Modern POS device. If you still experience issues, open Event Viewer in Windows, and view the logs that correspond to Modern POS. Look for warnings and errors that might help you determine which steps you missed in the previous sections.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable Azure Active Directory authentication for POS sign-in

2/18/2021 • 2 minutes to read • [Edit Online](#)

Many customers who use Microsoft Dynamics 365 Commerce also use other Microsoft cloud services, and they might use Azure Active Directory (Azure AD) to manage user credentials for those services. In those cases, the customers might want to use the same Azure AD account across applications. This topic explains how to configure the Commerce point of sale (POS) sign-in experience to use Azure AD authentication.

## Configure Azure AD authentication

To make Azure AD available as the authentication method for POS sign-in for a store, you must configure the settings of the store's functionality profile and then apply those settings to POS clients.

To configure a functionality profile, follow these steps.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**.
2. Select the functionality profile to change.
3. On the **Functions** FastTab, in the **POS staff logon** section, change the value of the **Logon Authentication Method** field from **Personnel ID and Password** to **Azure Active Directory**.

By default, all functionality profiles use **Personnel ID and Password** as the POS authentication method. Therefore, you must change the value of the **Logon Authentication Method** field if you want to use Azure AD. Every retail store that is linked to the selected functionality profile will be affected by this change.

To apply the settings to POS clients, follow these steps.

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Run the **1070 (Channel configuration)** distribution schedule.

### NOTE

Azure AD authentication requires an internet connection. It won't work when POS is in offline mode.

Currently, the **Manager override** function doesn't support Azure AD as an authentication method. An operator ID and password are required even if Azure AD is configured as the authentication method for POS sign-in.

## Associate an Azure AD account with a worker

Before a store worker can use an Azure AD account to sign in to the POS application, the Azure AD account must be associated with that worker.

To associate an Azure AD account with a worker, follow these steps.

1. Go to **Retail and Commerce > Employees > Workers**.
2. Open the details page for a worker.
3. On the Action Pane, on the **Commerce** tab, in the **External identity** group, select **Associate existing identity**.
4. In the **Use existing external identity** dialog box, select **Search using email**, enter an Azure AD email address, and then select **Search**.

5. Select the Azure AD account that is returned, and then select **OK**.

The **Alias**, **UPN**, and **External sub identifier** fields on the **Commerce** tab of the worker's details page will be filled in.

**NOTE**

After a worker record is updated, for example if a new Azure AD account is associated, a password is changed, or an employee address book is updated, it's recommended that you run **1060 (Staff)** distribution schedule to synchronize the latest staff information to the channel. That way, the POS application can fetch the correct data for user authentication and authorization check.

## Additional resources

[Set up extended logon functionality for MPOS and Cloud POS](#)

[Create a retail functionality profile](#)

[Configure a worker](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product search and customer search in the point of sale (POS)

2/18/2021 • 8 minutes to read • [Edit Online](#)

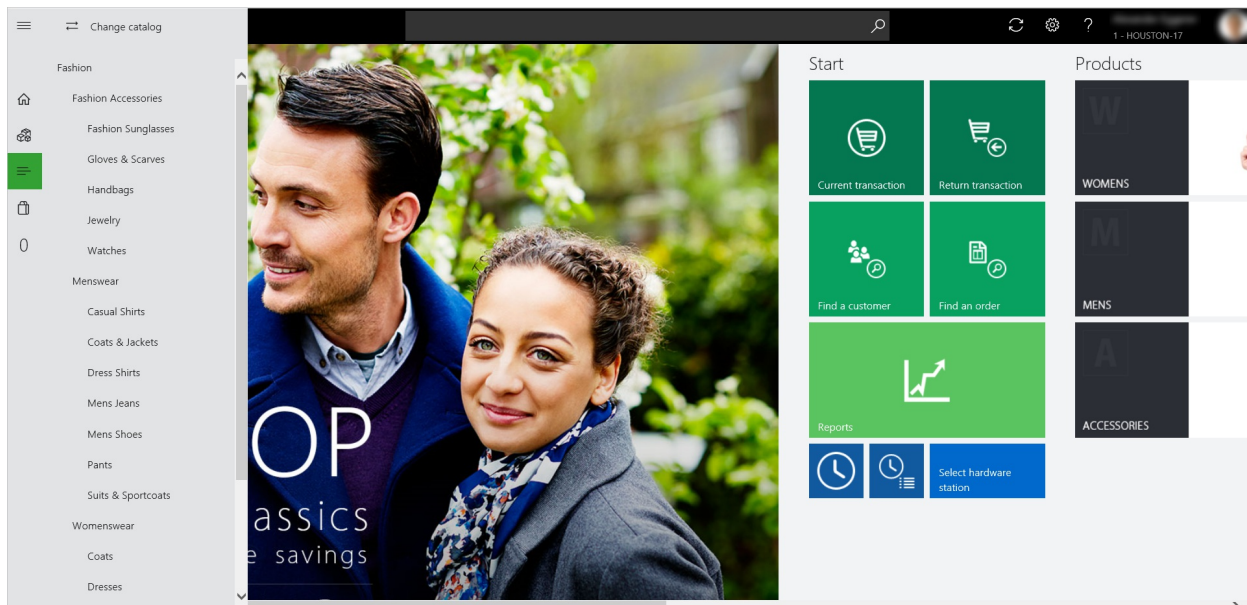
Modern Point of Sale (MPOS) and Cloud Point of Sale (CPOS) provide easy-to-use search functionality for products and customers. Because the search bar is always present at the top of the MPOS and CPOS windows, employees can quickly search for products and customers.

Employees can search for products in the assortments and catalogs that are associated with the current store. They can also search in the assortments and catalogs that are associated with any other store in the company. Therefore, cashiers can sell and return products outside the store assortment. Similarly, employees can search for customers who are associated with the current store or any other store in the company. Additionally, employees can search for customers who are associated with a different company in the parent organization.

## Product search

By default, product searches are done on the store assortment. This type of search is known as a *local product search*. However, employees can easily switch to any catalog that is associated with the current store, or they can search in a different store. This type of search is known as a *remote product search*. To change the catalog, select the **Categories** button on the left side of the page. At the top of the pane that appears, select the **Change catalog** button, and then select one of the available catalogs to browse it. The system will search the selected catalog for products.

On the **Change catalog** page, employees can easily select any store, or they can search for products across all stores.



A local product search searches in the following product properties:

- Product number
- Product name
- Description
- Dimensions
- Barcode

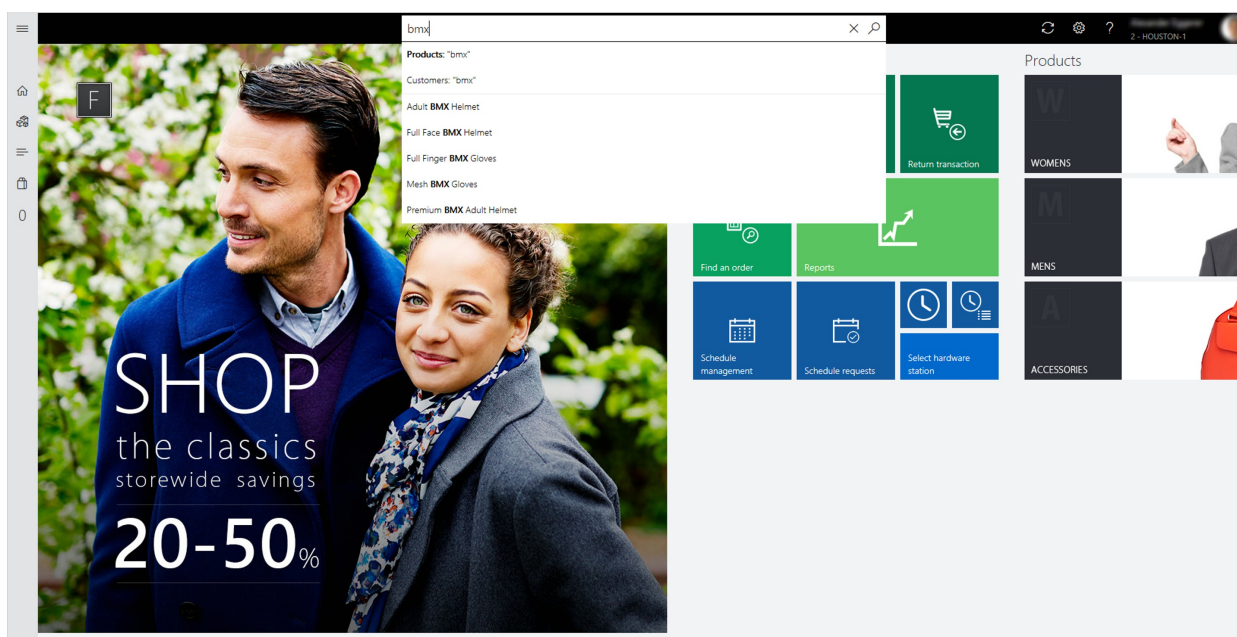
- Search name

### Additional local product search capabilities

- For multiple-keyword searches (that is, for searches that use search terms), retailers can configure whether the search results include results that match *any* search term or only results that match *all* search terms. The setting for this functionality is available in the POS functionality profile, in a new group that is named **Product search**. The default setting is **Match any search term**. This setting is also the recommended setting. When the **Match any search term** setting is used, all products that fully or partially match one or more search terms are returned as results. Those results are automatically sorted in ascending order of products that have the most keyword matches (full or partial).

The **Match all search terms** setting returns only products that match all the search terms (full or partial). This setting is helpful when the product names are lengthy, and employees want to see only limited products in the search results. However, this type of search has two limitations:

- The search is done on individual product properties. For example, only products that have all the searched keywords in at least one product property are returned.
  - Dimensions aren't searched.
- Retailers can configure product search to show search suggestions as users type product names. A new setting for this functionality is available in the POS functionality profile, in a group that is named **Product search**. The setting is named **Show search suggestions while typing**. This functionality can help employees quickly find the product that they are searching for, because they don't have to type the whole name manually.
  - The product search algorithm now also searches for the searched terms in the **Search name** property of the product.



## Customer search

Customer search is used to find customers for various purposes. For example, cashiers might want to view a customer's wish list or purchase history, or add the customer to a transaction. The search algorithm matches the search terms against the values that are present in the following customer properties:

- Name
- Email address
- Phone number
- Loyalty card number



- Address
- Account number

Among these properties, the name provides the most flexibility for multiple-keyword searches, because the algorithm returns all customers that match any of the searched keywords. The customers that match the most keywords appear at the top of the results. This behavior helps cashiers in situations where they search by typing the full name, but last name and first name were swapped during the initial data entry. However, for performance reasons, all the other properties preserve the order of the search keywords. Therefore, if the order of the search keywords doesn't match the order that the data is stored in, no results will be returned.

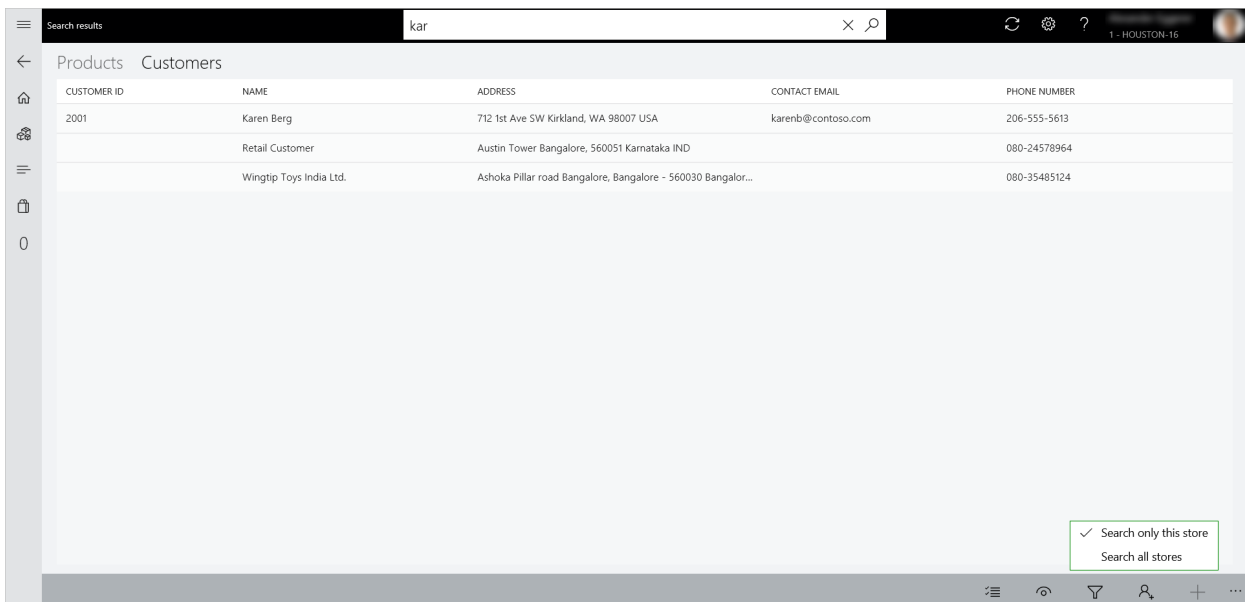
By default, a customer search is done on the customer address books that are associated with the store. This type of search is known as a *local customer search*. However, employees can also search for customers globally. In other words, they can search across the stores of the company and across all other legal entities. This type of search is known as a *remote customer search*.

To search globally, employees can select the **Filter results** button at the bottom of the page and then select the **Search all stores** option, as shown in the illustration that follows. In this case, not only customers are returned. All types of parties that are part of any address book in the headquarters are also returned. These parties include workers, vendors, contacts, and competitors.

#### NOTE

A minimum of four characters must be entered for a remote customer search to return results.

The customer ID isn't shown for customers queried from other legal entities, because no customer ID has been created for those parties in the current company. However, if an employee opens the customer details page, the system automatically generates a customer ID for the party and also associates the store's customer address books with the customer. Therefore, the customer will be visible in local store searches that are done later.



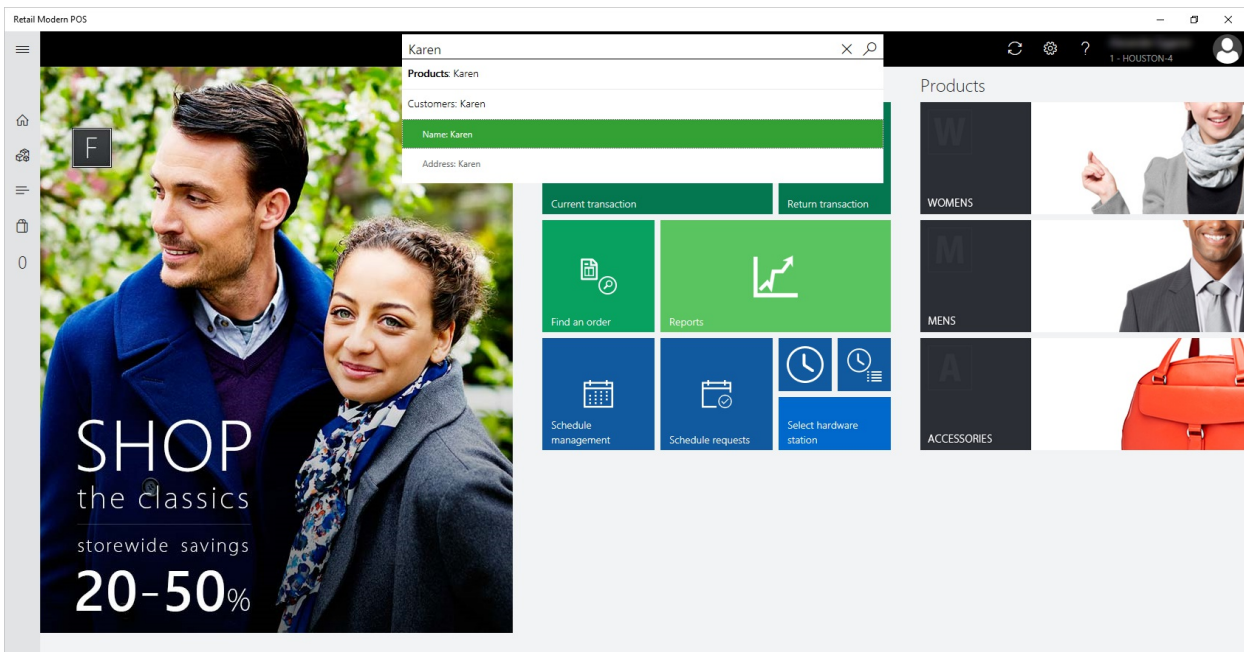
#### Additional local customer search capabilities

When the user searches for a phone number, the system ignores special characters (such as spaces, hyphens, and brackets) that might have been added when the customer was created. Therefore, cashiers don't have to worry about the phone number format when they search. For example, if a customer's phone number was entered as **123-456-7890**, a cashier can search for the customer by typing **1234567890**, or by entering the first few numbers of the phone number.

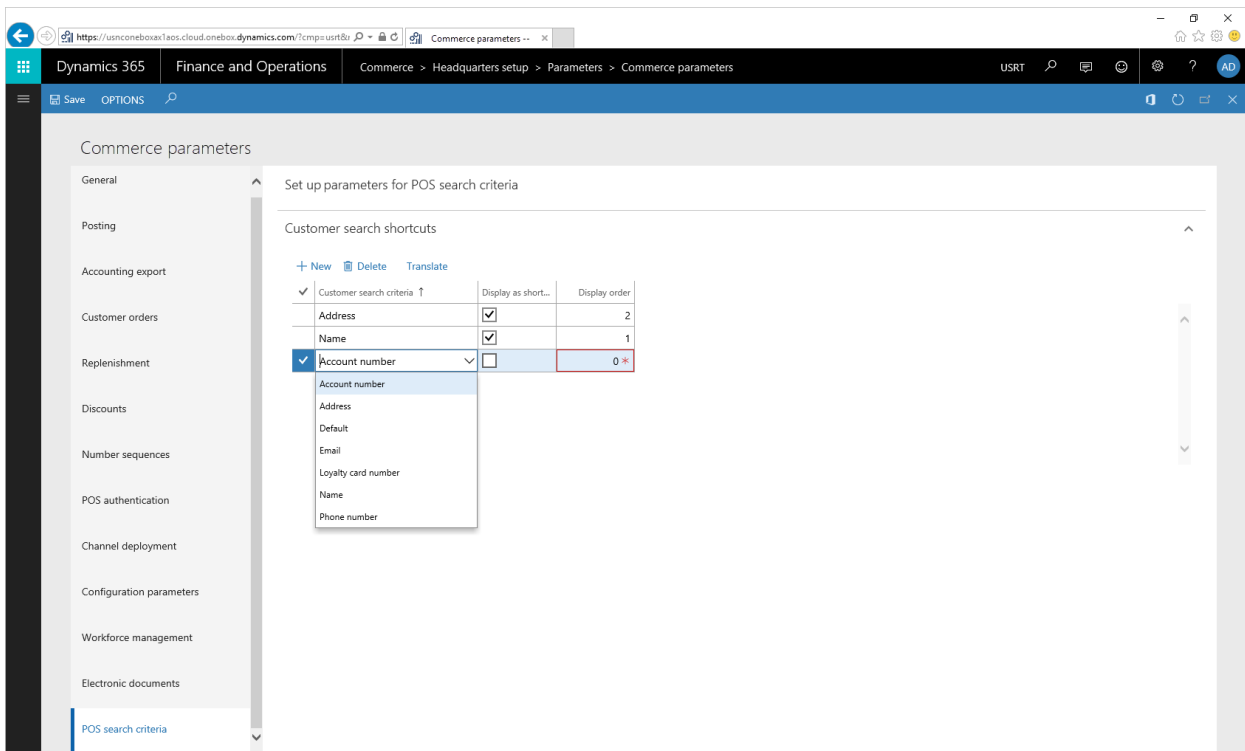
## NOTE

A customer can have multiple phone numbers and multiple emails. The customer search algorithm also searches through these secondary emails and phone numbers, but the customer search results page only displays the primary email and phone number. This may cause some confusion as the returned customer results would not show the searched email or phone number. In a future release we plan to improve the customer search results screen to show this information.

The traditional customer search can be time-consuming, because it searches across multiple fields. Instead, cashiers can search in a single customer property, such as name, email address, or phone number. The properties that the customer search algorithm uses are collectively known as the *customer search criteria*. The system admin can easily configure one or more criteria as shortcuts that will appear in POS. Because the search is limited to a single criterion, only the relevant search results are shown, and the performance is much better than the performance of a standard customer search. The following illustration shows the customer search shortcuts in POS.



To set search criteria as shortcuts, the admin must open the **Commerce parameters** page in Commerce, and then, on the **POS search criteria** tab, select all the criteria that should be shown as shortcuts.



#### NOTE

If you add too many shortcuts, the drop-down menu on the search bar in POS will become cluttered, and the employee's search experience can be affected. We recommend that you add only as many shortcuts as you require.

The **Display order** field determines the order in which shortcuts are shown in POS. The criteria that are shown are the out-of-box properties that the customer search algorithm uses to search customers. However, partners can add custom properties as search shortcuts. To add custom properties as search shortcuts, the system admin must extend the extensible enumeration (enum) that is used for the customer search criteria and then mark the partner's custom properties as shortcuts. Partners are responsible for writing the code to find results when their custom shortcuts are used for searches.

#### NOTE

A custom property that is added to the enum doesn't affect the standard customer search algorithm. In other words, the customer search algorithm won't search in the custom property. Users can use a custom property for searches only if that custom property is added as a shortcut, or if the default search algorithm is overridden.

Retailers can also set the default customer search mode in POS to **Search all stores**. This configuration can be helpful in scenarios where customers that were created outside POS must be searched immediately (for example, even before the distribution job is run). To do so, the retailer must turn on the **Default customer search mode** option in the POS functionality profile. Once it is set to **Yes**, every customer search attempt will then make a real-time call to headquarters.

To help prevent unexpected performance issues, this configuration is hidden behind a flighting flag that is named **CUSTOMERSEARCH\_ENABLE\_DEFAULTSEARCH\_FLIGHTING**. Therefore, to show the **Default customer search mode** setting the user interface (UI), the retailer should create a support ticket for its user acceptance testing (UAT) and production environments. After the ticket is received, the engineering team will work with the retailer to make sure that the retailer does testing in its non-production environments to assess the performance and implement any optimizations that are required.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Search for products and product variants during order entry

2/18/2021 • 5 minutes to read • [Edit Online](#)

Use the **Item number** field to search for products and product variants when you manually create a sales order line or a purchase order line. This lets you quickly find product variants when you only have the configuration string or one of the product dimensions available.

Sometimes, having too much of something is not the best situation to be in, and this is especially true if you sell a number of products that are similar, and you are trying to remember item numbers or product search names in order to find the right product to put on a sales order. You can use the **Item number** field on a sales order line or a purchase order line as a search field. You can enter any part of a product name, number, or dimension and get a lookup that displays all the items that match the search word.

## How search works

When you search for products or product variants, it is important to understand how the search feature finds the products that match the text that you enter. The key search rules in delivering search results are:

- Search results will return any matching record, disregarding the field that the search text is entered in.
- The search text needs to be present in the matching record in its full length.
- A match will occur even if the search text is found in the middle of a text string in the matching record. It does not have to appear in the beginning of a text string.
- The search text is treated as a single text string even if it contains white space.

### Examples

The following examples use products and product variants to illustrate how search is handled in various scenarios. **Prerequisite:** Under **Sales and marketing > Setup > Search > Search parameters > Search type**, select the **Full match** option.

PRODUCT TYPE	PRODUCT NAME	DISPLAY PRODUCT NUMBER	ITEM NUMBER	CONFIGURATION
Distinct product	SpeakerMidRange	D0001	D0001	NA
Product variant	Active speaker	D0010:::Black:	D0010	000005
Product variant	Active speaker	D0010:::White:	D0010	White

If you type 'speak' in the **Item number** field, you will get all the products above as a result in the lookup. If you type 'black' in the **Item number** field, you will get the second product as a result, because it has the text 'black' in the display product number. These two examples illustrate that the search is not only at the beginning of the field, a match will occur even if the search text is found in the middle of a text string in the matching record.

If you type '05' you will only get the second product variant as a result, because it has '05' in the configuration. This illustrates that the search is across all the enabled fields on the **Search criteria** page.

If you type 'speak 05' you will not get any results. This is because the search looks for the full text that is entered. The search will not try to find 'speak' and then narrow the results to those containing '05'.

You can limit the number of search results by using the **Number of results** field on the **Sales and marketing**

> **Setup** > **Search** > **Search parameters** page. If you set this field to 0, all search results will be returned. If you set it to 10, for example, it will return a maximum 10 search results.

## Configure the product search

Before you can use the product and product variant search feature, follow these steps to configure the product search.



### Step 1: Include all the relevant product and product variant identifiers and dimensions in the search criteria

Examples of product and product variant identifiers and dimensions that you can search by are **Product name**, **Item number**, **Display product number**, **Configuration**, **Color**, **Size**, **Style**, **Search name**, etc.

Go to **Sales and marketing** > **Setup** > **Search** > **Search criteria** page. The **Search criteria** page allows you to define criteria for customer, prospect, and product search. Make sure you filter the page by using product search criteria. You can do this by switching to **Product** in the page's menu.

To add the display product number to the search criteria, click **New** in the page's menu. This will add a new record in the **Search criteria** grid. Open the **Field name** column lookup and chose **DisplayProductNumber**. To add the product's configuration to the search criteria, create a new record in the **Search criteria** grid and chose **configId** in the **Field name** column. In the same manner, create a record with **Field name** **InventColorId** for the color dimension, **InventSizeId** for the size dimension, and **InventStyleId** for the style dimension.

### Step 2: Populate the database table that is used for product search

In the **Search criteria** page, click the **Update search data** button. In the **Update search data** dialog box, make sure that **Source** is set to **Product**, and then click **OK**. The system will aggregate in one table all the selected search criteria specified in step 1. If you have a lot of products and product variants, this operation can be quite lengthy and you may receive a warning. We recommend that you schedule the search table population on the batch server at a time when the server is not too busy.

Until the table is populated, product search will not provide the correct results. If you do not get any search results, make sure that this table is populated.

The table only has to be populated when the search criteria is modified. Newly released products and variants are automatically added to the table. Deleted products and variants are automatically removed from the table.

### Step 3: Enable the lookup for product search on sales and purchase order lines

You can enable this functionality by going to **Sales and marketing** > **Setup** > **Search** > **Search parameters** and setting **Enable lookup for search** to **Yes** on the **General** tab.

For sales order line entry, the default behavior is to open the **Product search** page when you start typing in the **Item number** field, and then press the **Tab** key. The **Product search** page changes the context during order line creation and may be considered unnecessarily intrusive. If you prefer to get the search results in a lookup and not lose context during order line entry, you can use the search lookup instead. If you search for a product or product variant, but you don't select anything in the lookup and press the **Tab** key, the **Product search** page will display.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cash management improvements

2/18/2021 • 3 minutes to read • [Edit Online](#)

Cash management is a key function for retailers in physical stores. Retailers want their stores to have systems that can help them provide complete traceability and accountability of cash and its movement across the different registers and cashiers in a store. They must be able to reconcile any differences and determine accountability.

Microsoft Dynamics 365 Commerce has cash management capabilities in its point of sale (POS) application. However, in versions of Retail that are earlier than version 10.0.3, cash management functionality isn't robust enough to provide complete traceability of cash movements in stores. Although retailers can reconcile the cash for a store, they can't precisely determine accountability in the event of a cash discrepancy.

In Retail version 10.0.3 and later, retailers will gain traceability for cash handling. As part of this traceability, retailers will be able to define safes, make two-sided cash transactions, and reconcile cash management transactions.

## Set up traceability and define safes

To set up the new cash management functionality, follow these steps to configure the functionality profile for stores.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**, and select a functionality profile that is linked to the stores where you want to make the improvements for cash management available.
2. On the **Functions** FastTab of the functionality profile, under **Advanced cash management**, set the **Enable cash traceability** option to **Yes**.
3. To set up safes, go to **Retail and Commerce > Channels > Stores > All stores**, and select a store.
4. On the **Stores** page, on the Action Pane, on the **Set up** tab, in the **Set up** group, select **Safes**. By using this option, you can define and maintain multiple safes for a store.
5. Before the functionality can be used, you must run the **1070 Channel configuration** distribution schedule job to sync these configurations to the channel database.

## Additional cash management changes

In Retail version 10.0.3 and later, the following capabilities that are related to cash transactions are also provided:

- A user who is prompted to "declare start amount" must enter the source of cash. The user can search for the available safes that are defined in the store and select the safe that the cash is being taken out of so that it can be put into the register.
- A user who does a "tender removal" operation is prompted to select, in a list of open "float entry" transactions, the transaction that the operation is being done against. If the corresponding float entry doesn't exist in the system, the user can create a non-linked tender removal transaction.
- A "float entry" operation prompts a user to select, in a list of open "tender removal" transactions, the transaction that the float entry operation is being done against. If the corresponding tender removal doesn't exist in the system, the user can create a non-linked float entry transaction.
- A user who makes a "safe drop" is prompted to select the safe where the cash is being dropped.
- For safes that are defined in a store, users can manage operations such as declaring the start amount, doing a float entry, doing a tender removal, and making a bank drop.



- For users who have the appropriate user privileges, "manage shifts" operations show the cash balances of active, suspended, and blind closed shifts.
- To reconcile the cash transactions within a shift or across shifts, select the shift to reconcile, and then select **Reconcile**. The view that is opened shows the list of reconciled and unreconciled transactions on separate tabs. From this view, users can either select unreconciled transactions and reconcile them, or select previously reconciled transactions and unreconcile them.
- During reconciliation, if the selected transaction doesn't balance, the user must enter a description of the reason for the unbalanced reconciliation. Users can select a single transaction and reconcile it with the relevant reason description as they require.
- Users can continue to reconcile and unreconcile transactions until the shift is closed. After a shift is closed, the transactions can't be unreconciled.
- When a user chooses to close a shift, Commerce validates that there are no unreconciled cash management transactions in the shift. Users can't close a shift if there are unreconciled transactions.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

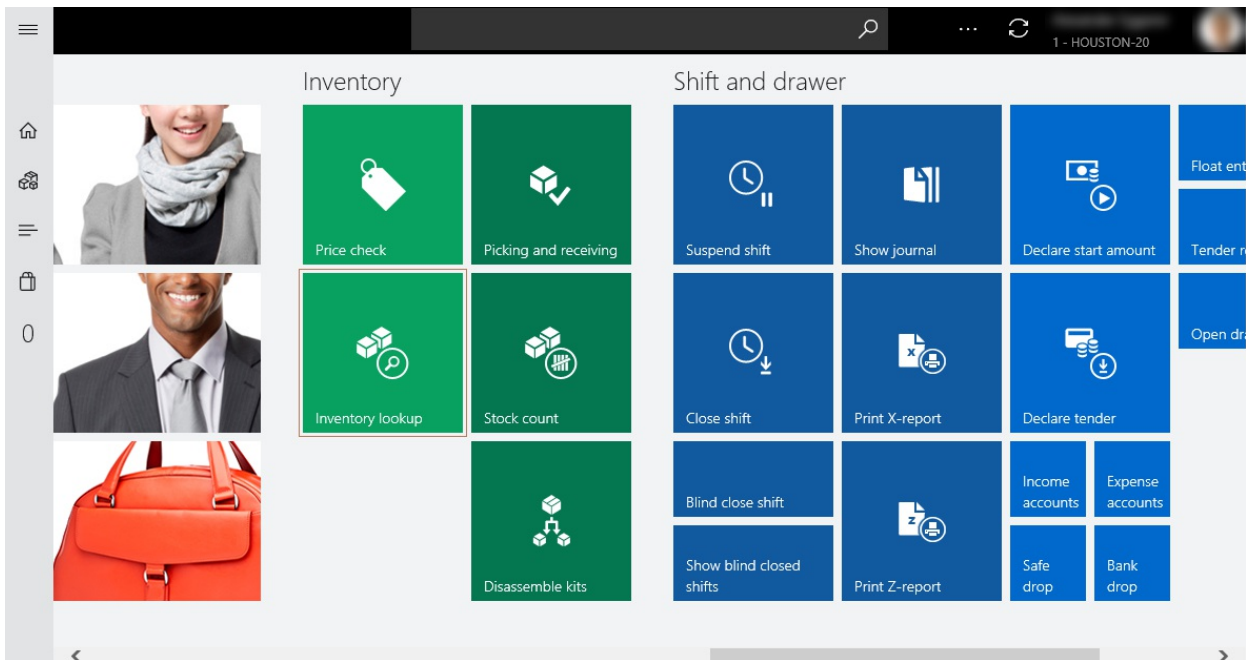
# Inventory lookup in the point of sale (POS)

2/18/2021 • 5 minutes to read • [Edit Online](#)

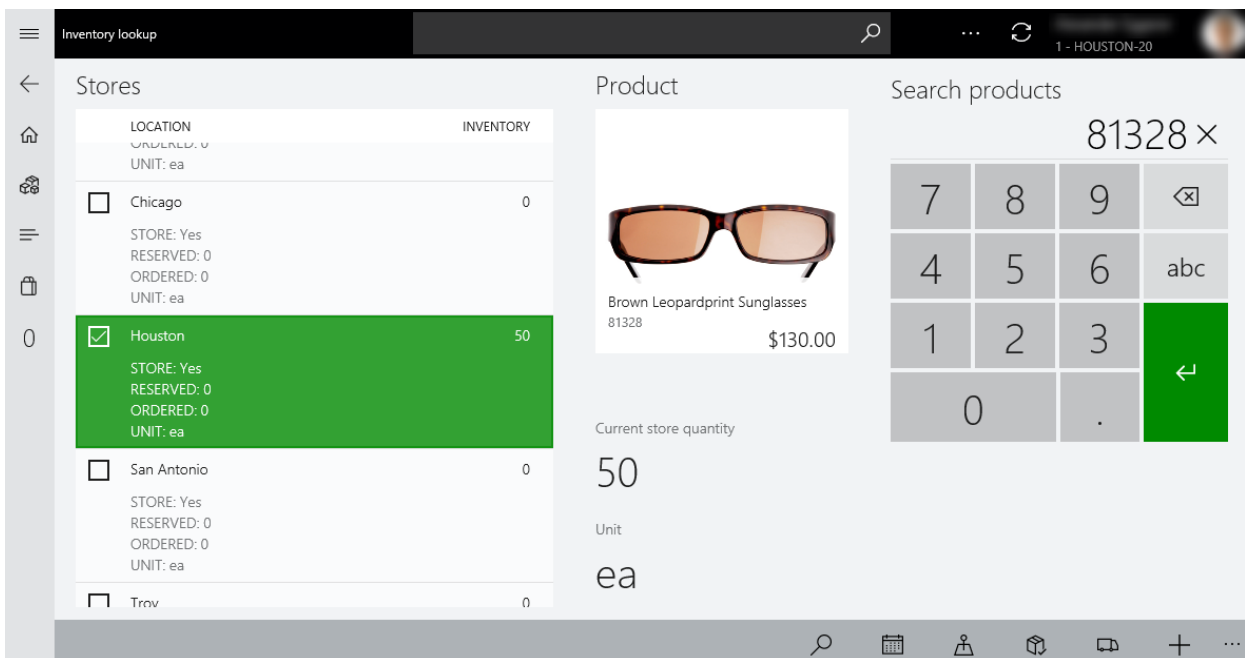
Inventory lookup in the point of sale (POS) helps retailers achieve real-time operational excellence and gain insights by connecting stores, the POS, and the back office. This functionality provides an accurate real-time view of product inventory across stores and distribution centers. It also helps retailers drive additional efficiencies and cost savings by improving inventory planning in real time.

An accurate real-time view of inventory across an organization helps store associates provide timely, superior customer service. The moment that matters most is the moment when the customer is ready to make a purchase decision. It's important that cashiers in the store have real-time inventory information at their fingertips, so that they can accurately promise product delivery and pickup.

You can open the **Inventory lookup** page from the **Retail Modern POS** workspace or the **Retail Cloud POS** workspace.



On the **Inventory lookup** page, you can use the numeric keyboard to enter a product number. You can then view the on-hand quantity for multiple stores and warehouses.



**Reserved** and **Ordered** quantities are also shown for each location.

- **Reserved** – This quantity refers to the **Physical reserved** value from the back office for the specified product number at the location.
- **Ordered** – This quantity refers to the **Ordered in total** value from the back office for the specified product number at the location.

## Locations that inventory availability information is shown for

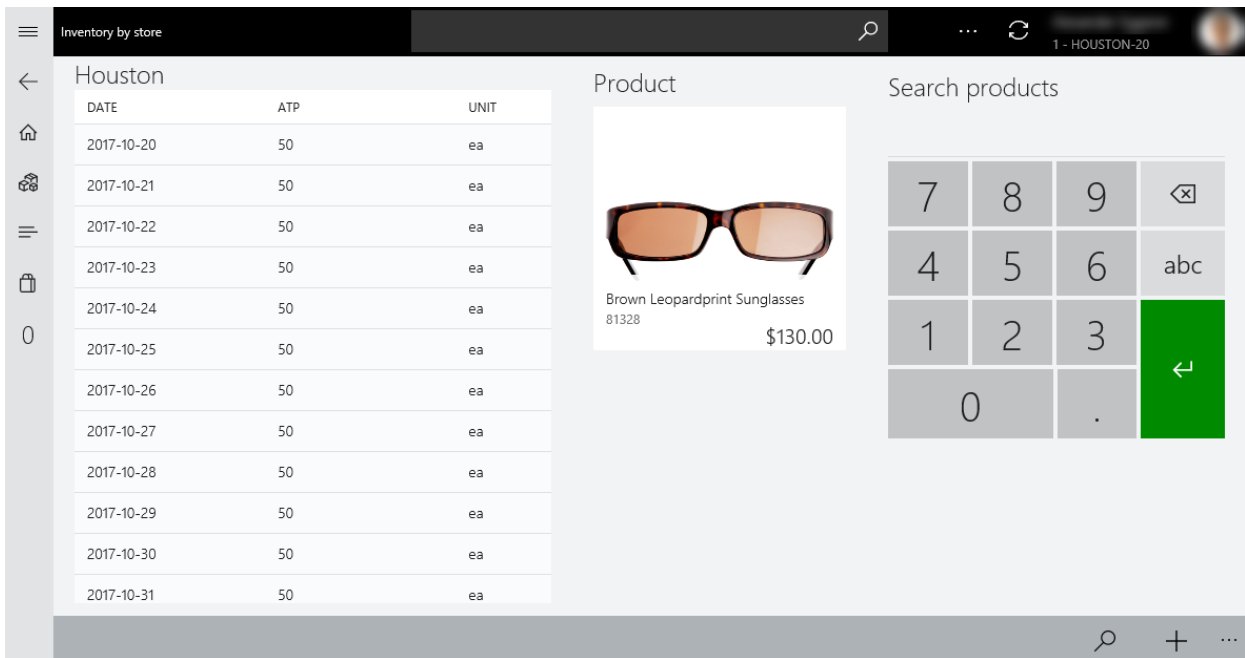
The list of locations includes two types of entities:

- **Stores** – The list shows stores that are configured by using the store locator group for the current store in the Headquarters.
- **Distribution centers** – Various types of distribution centers (such as warehouses) can be configured in Commerce. However, the list shows inventory availability information only for distribution centers of the **Standard** default type.

### NOTE

Inventory availability information isn't shown for warehouses of the **Transit**, **Quarantine**, and **Goods in Route** types for the POS.

On the **Inventory lookup** page, you can view available to promise (ATP) quantities for each store, in addition to the current on-hand quantities, reserved quantities, and ordered quantities. Select the store to view the ATP information for, and then select **Show store availability**.

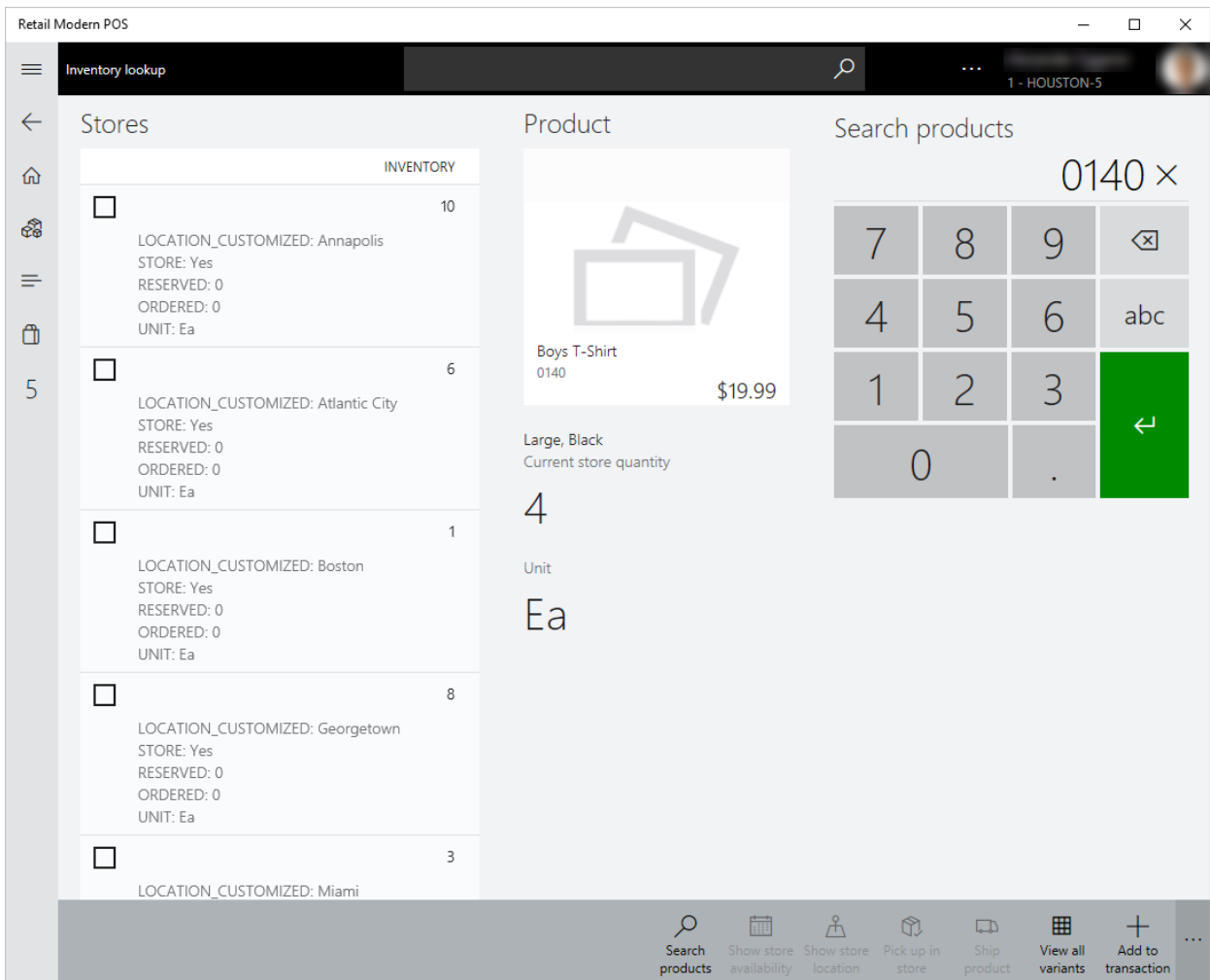


## Opening the Dimension based matrix view to show all variants

On the **Product details** page of a product master, or on the **Inventory lookup** page, select **View all variants** from the app-bar at bottom of the page. The **Dimension based matrix** view for the initial launch from these pages shows the inventory availability information for all variants of a product for the current store.

### NOTE

The **View all variants** button is available only for item product masters that have product variants. It isn't available for standalone products or kits.



Select **View all variants** on the **Product details** page of a product master, or on the **Inventory lookup** page, without selecting a location, to go to the **Dimension based matrix** view to view the inventory availability information for all variants of a product for the current store.

Inventory lookup: Houston

1 - HOUSTON-5

0140 : Boys T-Shirt

Color	Size				
	LARGE	MEDIUM	SMALL	XLARGE	XSMALL
BLACK	RESERVED: 0 ORDERED: 0 4 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA
BLUE	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA
GREEN	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA
MAGENTA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA	RESERVED: 0 ORDERED: 0 3 EA
PURPLE	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA
RED	n/a	n/a	n/a	n/a	n/a
WHITE	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA	RESERVED: 0 ORDERED: 0 0 EA
YELLOW	n/a	n/a	n/a	n/a	n/a

Store

**NOTE**

In the preceding illustration, the display order of the dimensions is alphabetic, because the display order of dimensions wasn't configured for the selected product.

In the **Dimension based matrix view**, the cells for the product variants include an on-hand value in the lower-right corner. The following table explains the meaning of the various values.

ON-HAND VALUE	DESCRIPTION
Numeric value that is more than 0 (zero)	A variant has been released to the selected location, and you can perform additional actions in the cell. (These actions are described in more detail later in this topic.)
0 (zero)	A variant has been released to the selected location, but the item isn't available in selected location. However, you can perform additional actions in the cell. (These actions are described in more detail later in this topic.)
n/a or an inactive cell	A variant hasn't been released to the selected location, and you can't perform additional actions in the cell.

You can also change the pivot for dimensions by selecting the new dimension to use.

Inventory lookup: Houston

0141 : Girls T-Shirt

Color Size

Color	XSMALL	SMALL	MEDIUM	LARGE	XLARGE
WHITE	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA
BLUE	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 7 EA	RESERVED: 0 ORDERED: 0 6 EA
MAGENTA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA
PURPLE	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA
DGREY	n/a	n/a	n/a	n/a	n/a

Inventory lookup: Houston

0141 : Girls T-Shirt

Size Color

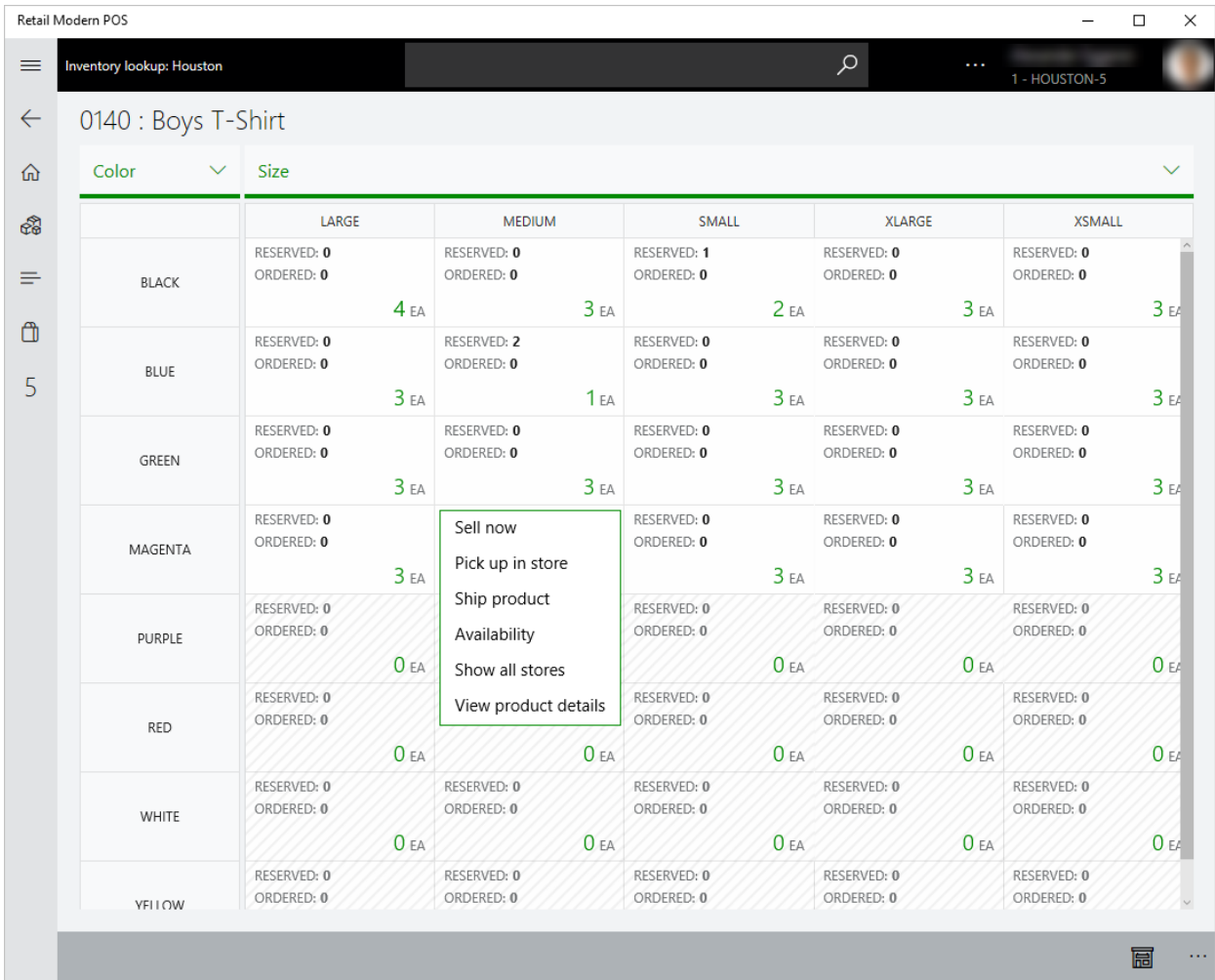
Size	WHITE	BLUE	MAGENTA	PURPLE	DGREY
XSMALL	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	n/a
SMALL	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	n/a
MEDIUM	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	n/a
LARGE	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 7 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	n/a
XLARGE	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	RESERVED: 0 ORDERED: 0 6 EA	n/a

**NOTE**

In the preceding illustrations, the display order of the dimensions for the selected product is custom (non-alphabetic). It's based on the dimension display order that is set in the back office.

Additionally, in the **Dimension based matrix** view, more actions can be performed to help boost a store associate's productivity. Here are some examples:

- Change the store location to look up the inventory availability of all product variants at other locations. These locations include other stores in the store locator group and distribution centers of the **Standard** default type.
- Sell an individual product variant to a customer by using cash and carry, in-store pickup, or shipment to an address.
- Provide the customer with ATP information for an individual product variant at a specific location.



**NOTE**

In the preceding illustration, the display order of the dimensions is alphabetic, because the display order of dimensions wasn't configured for the selected product.

The following table provides more information about the additional actions that are available.

ACTION	DESCRIPTION
Sell now	Add the selected item variant to the transaction, and redirect the user to the transaction screen. (This action isn't available when the selected location is a distribution center.)
Pick up in store	Create a customer order for the product variant that will be picked up from the selected location, and redirect the user to the transaction screen. (This action isn't available when the selected location is a distribution center.)
Ship product	Create a customer order for the product variant that will be shipped from the selected location, and redirect the user to the transaction screen.
Availability	Show the ATP information for the selected variant combination for the selected location.



ACTION	DESCRIPTION
Show all locations	Switch to the standard inventory lookup view, and highlight inventory availability information for the item variant across all stores in the store locator group, and also in distribution centers of the <b>Standard/Default</b> type.
View product details	Redirect the user to the <b>Product details</b> page of the associated product master.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Suspend and resume a transaction in the point of sale (POS)

2/18/2021 • 3 minutes to read • [Edit Online](#)

Point of sale (POS) users can suspend in-progress transactions, and then resume them later or on a different register. Transactions are often suspended to quickly free up a register for a different task without losing any progress on the current transaction. For example, a store associate starts to process a customer's transaction on a mobile device but must complete it on a register that has a cash drawer. In this case, the store associate can suspend the transaction on the mobile device, and then recall and resume it on a register.

## Configure suspend and resume functionality

### POS operations

Two [POS operations](#) let the POS support suspend and resume scenarios. You can assign these operations to [button grids](#) on the transaction page or the welcome page.

- 503: Suspend transaction
- 504: Recall transaction

### Receipt template

The POS can be configured to generate a printed slip when a transaction is suspended. That slip can then be used to quickly identify and recall the transaction later.

To enable the POS to print a slip, you must add the **Suspended transaction** receipt format to the store's receipt profile. You can design the receipt format so that it includes or excludes specific details about the transaction. For example, the format can include a barcode to support scanning.

### Receipt numbering

As for other POS transaction types that generate a printed receipt, you can define a number sequence for suspended transactions in the **Receipt numbering** section of the store's functionality profile.

### Void when closing shift

You can use the **Void when closing shift** option to require that users either complete or void any suspended transactions before they close their shift. During the **Close shift** operation, the POS will prompt users to either view or void any outstanding suspended transactions.

## Suspend and resume a transaction

### Suspend a transaction

Users who have sufficient privileges, and who have a screen layout that includes the **Suspend transaction** operation, can suspend a transaction so that it can be recalled later or on a different register.

Transactions can be suspended only if they do **not** contain the following types of lines:

- Active payment lines
- Gift card lines (either to issue a gift card or to add to the gift card balance)

A suspended transaction doesn't affect sales information or inventory availability information for the store.

### Resume a suspended transaction

Suspended transactions can be recalled and resumed in the same store by any user who has sufficient privileges, and who also has a layout that includes the **Recall transaction** operation.

To quickly and easily recall a suspended transaction, scan the barcode on the printed slip while you're viewing the list of transactions from the **Recall transaction** operation.

### Considerations for offline mode

- Any transaction that is suspended while the POS is in online mode can't be resumed in offline mode, because the data isn't synced to the offline database.
- If you suspend a transaction while the POS is in offline mode, you can recall it in offline mode, provided that the POS wasn't switched back to online mode at any time since you suspended the transaction. When the POS is switched back to online mode, data about suspended transactions is moved to the online database and removed from the offline database. Therefore, the transactions can be resumed only in online mode. If you switch the POS back to offline mode, you won't be able to resume those suspended transaction, because they have already been removed from the offline database.

### Void a suspended transaction

You can void suspended transactions either by recalling the transaction and then performing the **Void transaction** operation, or by selecting the transaction in the **Recall transaction** list and selecting **Void** on the app bar. Alternatively, the store can be configured to prompt users to void suspended transactions when they close their shift.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Offline point of sale (POS) functionality

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides information about offline mode for Modern POS, in which POS devices automatically switch from the channel database to the offline database if the Commerce Scale Unit is unavailable. This article also includes general setup information for offline mode and explains the data synchronization that occurs between the offline database and the channel database.

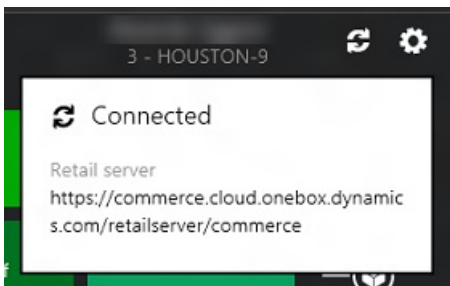
## Overview

In Modern POS, a point of sale (POS) device goes into offline mode whenever the Commerce Scale Unit is unavailable. Therefore, if the connection is lost, the POS automatically switches to the offline database.

During a sales transaction, if a data request doesn't succeed within the time-out interval that is configured in the offline profile, the POS automatically switches to the offline database and continues the sales transaction. While the POS device is in offline mode, Rail Modern POS tries to reconnect to the Commerce Scale Unit after the reconnection attempt interval that is configured in the offline profile. This reconnection attempt occurs only at the beginning of a transaction.

### Determining the connection mode of Modern POS

The status header in Modern POS indicates the current connection status, and the **Connection status** window shows the status of the last attempt to sync with the offline database.



### Creating a button to manually switch between online and offline modes

You can add a button to Modern POS to manually switch between online and offline modes. Create a button for POS operation 917 – **Database connection status**. The name of this button is **Disconnect** when the POS is connected to the Commerce Scale Unit and **Connect** when it is disconnected. You can use this button to view the connection, and to disconnect from the Commerce Scale Unit or connect to it.

3 - HOUSTON-9

## Connection status

Overview

Connected  
Yes

Retail server  
https://commerce.cloud.onebox.dynamics.com/retailserver/commerce

Pending transactions in offline database  
2

Offline sync status  
Last updated 21 hours ago

[Disconnect](#) [Sync](#)

Download Upload

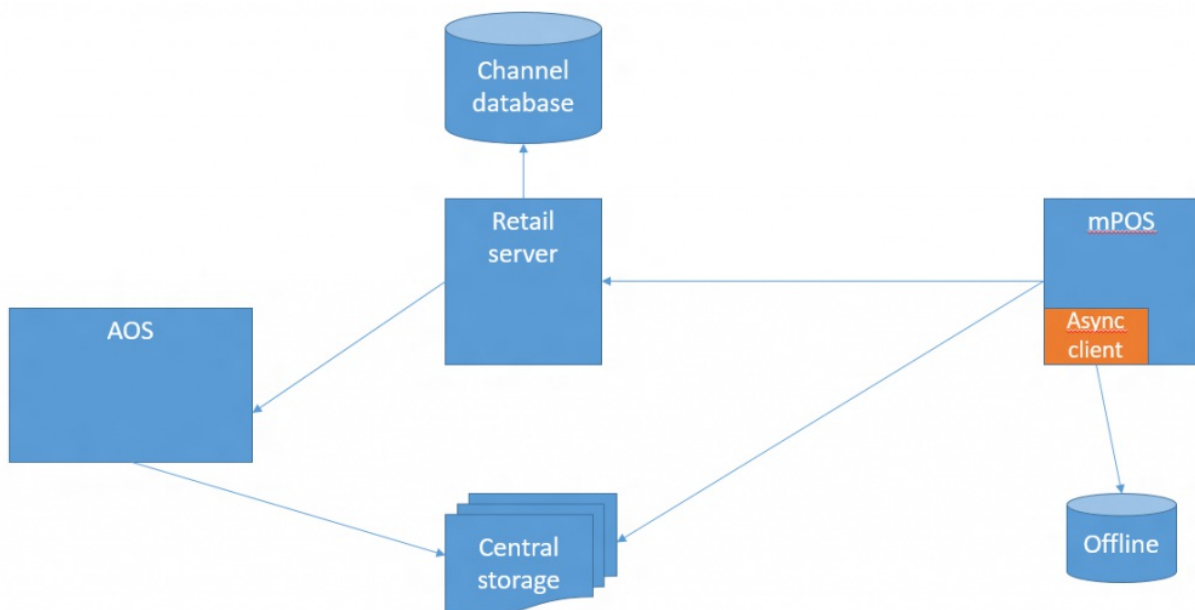
JOB NAME	STATUS	DATE AND TIME	SIZE (MB)
1160 - POS redeployment	OK	08/12/2015 3:26 PM	< 1MB
1170 - POS task recorder	OK	08/12/2015 3:26 PM	< 1MB
1140 - Sales orders	OK	08/12/2015 3:26 PM	< 1MB
1150 - Catalog	OK	08/12/2015 3:26 PM	36MB
1120 - Modes of delivery	OK	08/12/2015 3:26 PM	< 1MB
1130 - Product availability	OK	08/12/2015 3:26 PM	< 1MB
1115 - Global address book reference data	OK	08/12/2015 3:26 PM	< 1MB
1110 - Global configuration	OK	08/12/2015 3:25 PM	< 1MB
1095 - Offline database configuration	OK	08/12/2015 3:25 PM	< 1MB
1100 - Product and price parameters	OK	08/12/2015 3:25 PM	< 1MB
1070 - Channel configuration	OK	08/12/2015 3:25 PM	36MB

## Setup

To enable offline support for a POS device (register), set the **Support offline** option to **Yes** on the **Register** page. A new channel database entity is created and added to the store's channel data group. Then run all the required distribution schedules to generate the data packages for the offline database. Next, install the offline version of Modern POS. The installation process creates the offline database. Additionally, install Microsoft SQL Server 2014 Express if it is required. Offline data synchronization starts after the first sign-in to Modern POS.

## Data synchronization

The Commerce scheduler is used to send master data to the offline database. By default, when a distribution schedule is run, data changes are sent to both the channel database and the offline database. Modern POS includes the async sync library, which downloads any available data packages and inserts them into the offline database. If any transactions are created offline, the POS uploads them to the Commerce Scale Unit, so that they can be inserted into the channel database. Offline data synchronization can occur only if Modern POS is running.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Shift and cash drawer management

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic explains how to set up and use shifts in Commerce point of sale (POS).

In Dynamics 365 Commerce, the term *shift* describes the collection of POS transactional data and activities between two points in time. For each shift, the amount of money that is expected is compared against the amount that was counted and declared.

Typically, shifts are opened at the start of the business day. At that point, a user declares the starting amount that the cash drawer contains. Sales transactions are then performed throughout the day. Finally, at the end of the day, the drawer is counted, and the closing amounts are declared. The shift is closed, and a Z report is generated. The Z report indicates whether there is an overage or shortage.

## Typical shift scenarios

Commerce provides several configuration options and POS operations to support a wide range of end-of-day business processes for the POS. This section describes some typical shift scenarios.

### Fixed till

Traditionally, this scenario has been used most often. It's still extensively used. In a "fixed till" shift, the shift and till are associated with a specific register. They aren't moved from one register to another. A "fixed till" shift can be used by a single user or shared among multiple users. "Fixed till" shifts don't require any special configuration.

### Floating till

In a "floating till" shift, the shift and cash drawer can be moved from one register to another. Although a register can have only one active shift per cash drawer, shifts can be suspended and then resumed later or on a different register.

For example, a store has two registers. Each register is opened at the start of the day when the cashier opens a new shift and provides the starting amount. When one cashier is ready to take a break, that cashier suspends his or her shift and removes the till from the cash drawer. That register then becomes available to other cashiers. Another cashier can sign in to and open his or her own shift on the register. After the first cashier's break has ended, that cashier can resume his or her shift when one of the other registers becomes available. "Floating till" shifts don't require any special configuration or permission.

### Single user

Many retailers prefer to allow only one user per shift, to help guarantee the highest level of accountability for the cash in the cash drawer. If only one user is allowed to use the till that is associated with a shift, that user can be held solely responsible for any discrepancies. If more than one user uses a shift, it's difficult to determine who made an error, or who might be trying to steal from the till.

### Multiple users

Some retailers are willing to sacrifice the level of accountability that single-user shifts provide and to allow more than one user per shift. This approach is typical when there are more users than available registers, and the need for flexibility and speed outweighs the potential for loss. It's also typical when store managers don't have their own shifts but can, as required, use the shifts of any of their cashiers. To sign in to and use a shift that was opened by another user, a user must have the **Allow multiple shift logon** POS permission.

### Shared shift

A "shared shift" configuration lets retailers have a single shift across multiple registers, cash drawers, and users. A shared shift has a single starting amount and a single closing amount that are summarized across all cash drawers. Shared shifts are most typical when mobile devices are used. In this scenario, a separate cash drawer isn't reserved for each register. Instead, all registers can share one cash drawer.

For shared shifts to be used in a store, the cash drawer must be configured as a "shared shift drawer" at **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles > Drawer**. Additionally, users must have one or both of the shared shift permissions (Allow manage shared shift and Allow use shared shift).

#### **NOTE**

Only one shared shift can be open at a time in each store. Shared shifts and stand-alone shifts can be used in the same store.

## Shift and drawer operations

Various operations can be performed to change the state of a shift, or to increase or decrease the amount of money in the cash drawer. This section describes these shift operations for Modern POS and Cloud POS.

### **Open shift**

The POS requires that users have an active, open shift in order to perform any operations that will produce a financial transaction, such as a sale, return, or customer order.

When a user signs in to the POS, the system first verifies whether an active shift is available for that user on the current register. If an active shift isn't available, the user can open a new shift, resume an existing shift, or sign in in "non-drawer" mode, depending on the system configuration and the user's permissions.

### **Declare start amount**

This operation is often the first operation that is performed for a newly opened shift. For this operation, users specify the starting cash amount in the cash drawer for the shift. This operation is important because the overage/shortage calculation that occurs when a shift is closed considers the start amount.

### **Float entry**

*Float entries* are non-sales transactions that are performed in an active shift to increase the amount of cash in the cash drawer. A typical example of a float entry is a transaction to add additional change to the drawer when it's running low.

### **Tender removal**

*Tender removals* are non-sales transactions that are performed in an active shift to reduce the amount of cash in the cash drawer. This operation is most often used in conjunction with a Float entry operation on a different shift. For example, because register 1 is running low on change, the user on register 2 does a tender removal to reduce the amount in his or her cash drawer. The user on register 1 then does a float entry to increase the amount in his or her cash drawer.

### **Suspend shift**

Users can suspend their active shift to free up the current register for another user, or to move their shift to a different register (in this case, the shift is often referred to as a "floating till" shift).

Suspension of a shift prevents any new transactions or changes to the shift until it's resumed.

### **Resume shift**

This operation lets users resume a previously suspended shift on any register that doesn't already have an active shift.



## **Tender declaration**

This operation is performed to specify the total amount of money that is currently in the cash drawer. Users most often perform this operation before they close a shift. The specified amount is compared against the expected shift amount to calculate the overage/shortage amount.

## **Safe drop**

Safe drops can be done on an active shift at any time. This operation removes money from the cash drawer so that it can be transferred to a more secure location, such as a safe in the back room. The total amount that is recorded for safe drops is included in shift totals, but it doesn't have to be counted as part of the tender declaration.

## **Bank drop**

Like safe drops, bank drops are done on active shifts. This operation removes money from the shift to prepare for the bank deposit.

## **Blind close shift**

*Blind-closed shifts* are no longer active but haven't been fully closed. Unlike suspended shifts, blind-closed shifts can't be resumed. However, operations such as Declare start amount and Tender declaration can be performed on them later or from a different register.

Blind-closed shifts are often used to free up a register for a new user or shift without first having to fully count, reconcile, and close the shift.

## **Close shift**

This operation calculates shift totals and overage/shortage amounts, and then finalizes an active or blind-closed shift. Depending on the user's permissions, a Z report is also printed for the shift. Closed shifts can't be resumed or modified.

## **Print X**

This operation generates and prints an X report for the current active shift.

## **Reprint Z**

This operation reprints the last Z report that the system generated when a shift was closed.

## **Manage shifts**

This operation lets users view all active, suspended, and blind-closed shifts for the store. Depending on their permissions, users can perform their final closing procedures, such as Tender declaration and Close shift operations for blind-closed shifts. This operation also lets users view and delete invalid shifts, in the rare event that shifts are left in a bad state after a switch between offline and online modes. These invalid shifts don't contain any financial information or transactional data that is required for reconciliation.

# Shift and drawer permissions

The following POS permissions affect what a user can and can't do in various scenarios:

- **Allow blind close**
- **Allow X-report printing**
- **Allow Z-report printing**
- **Allow tender declaration**
- **Allow floating declaration**
- **Open drawer without sale**
- **Allow multiple shift logon** – This permission allows the user to sign in to and use a shift that a different user opened. Users who don't have this permission can sign in to and use only shifts that they have opened.
- **Allow manage shared shift** – Users must have this permission to open or close a shared shift.

- **Allow use shared shift** – Users must have this permission to sign in to and use a shared shift.

## Back-office end-of-day considerations

The way that shifts and cash drawer reconciliation are used in the POS differs from the way that transaction data is summarized during statement calculation. It's important that you understand this difference. Depending on your configuration and your business processes, the shift data in the POS (the Z report) and a calculated statement in the back office can give you different results. This difference doesn't necessarily mean that either the shift data or the calculated statement is incorrect, or that there is a problem with the data. It just means that the parameters that are provided might be including additional transaction or fewer transactions, or that the transactions have been summarized differently.

Although every retailer has different business requirements, we recommend that you set up your system in the following way to avoid situations where differences of this type occur:

Go to **Retail and Commerce > Channels > Stores > All stores > Statement/closing**, and for each store, set both the **Statement method** field and the **Closing method** field to **Shift**.

This setup helps guarantee that back-office statements include the same transactions as shifts in the POS, and that the data is summarized by that shift.

For more information about statement and closing methods, see [Store configurations for Retail statement](#).

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cash out gift card balance for a retail customer

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides an overview of the cash out gift card feature for the Dynamics 365 Retail Modern POS (MPOS).

The purpose of the cash out feature is to allow cashiers to cash out the remaining amount on a gift card. Retailers often need to exchange a low balance gift card for cash at the customer's request.

## Prerequisites

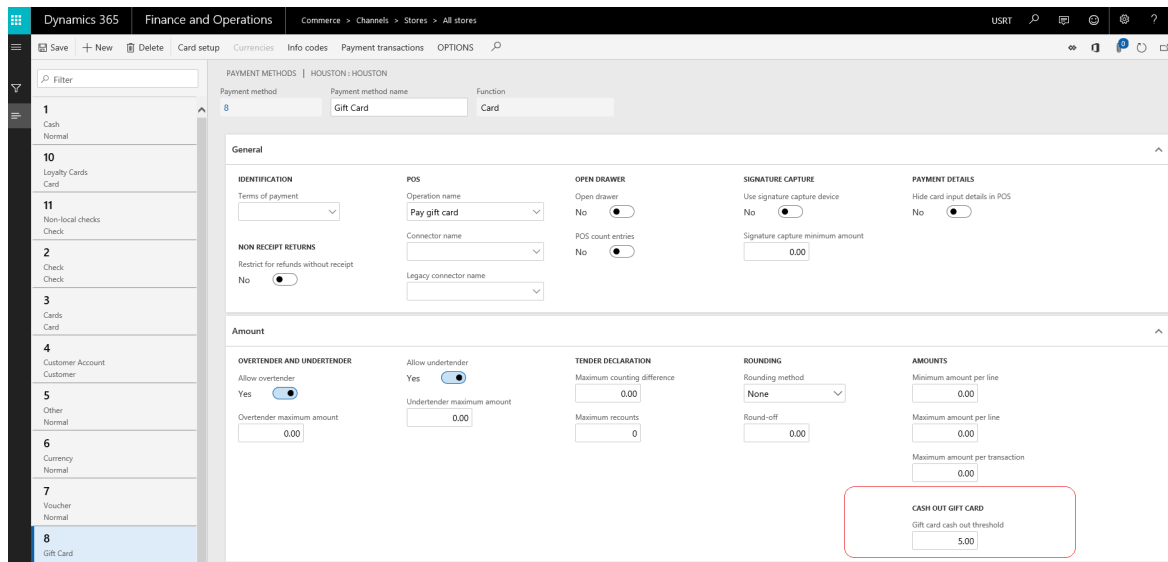
- The payment connector and corresponding payment gateway or processor must support the feature. The *payment connector* is an extension which facilitates communication between Dynamics 365 Commerce (and associated components) and a payment service. The connector described in this topic was implemented using the standard payments SDK.
- If the gift cards are external gift cards, the external gift card must be configured for both the Headquarters and the POS. Before the gift card can be configured, the retailer must have an account with an external gift card service provider.

## Scenarios

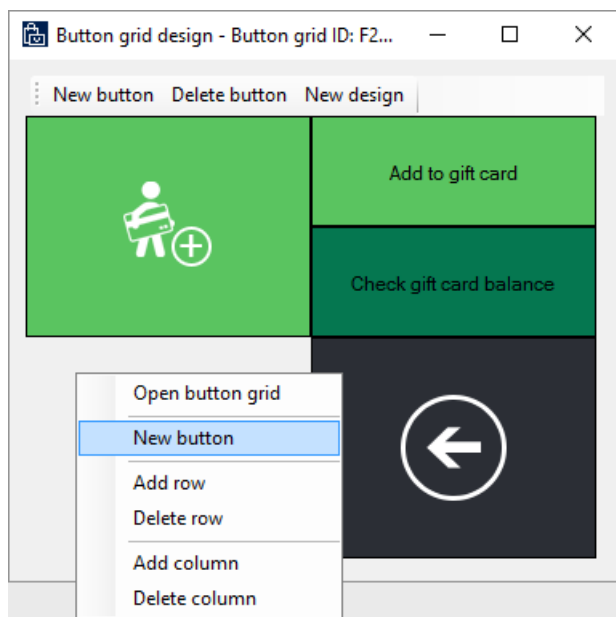
The cash out gift card feature is applicable to a scenario where, for example, in Washington state, the cash out threshold is \$5. Retailers in this case will have the option to set up an operation to cash out a gift card and set the gift card balance limits under which the cash out operation can be enabled.

## Configure Headquarters

1. Open the **All stores** page.
2. In the list, select the **Houston** store.
3. On the **Action Pane**, select **Set up > Payment methods**.
4. Search for **payment methods** to open the **Payment methods** page.
5. Select the **Gift Card** payment method, and then follow these steps:
  - a. In the **Amount** FastTab section, select the **Cash Out Gift Card** field.
  - b. In the **Cash Out Gift Card** field, enter the **Gift card Cash out threshold** amount.
  - c. Select **Save**.



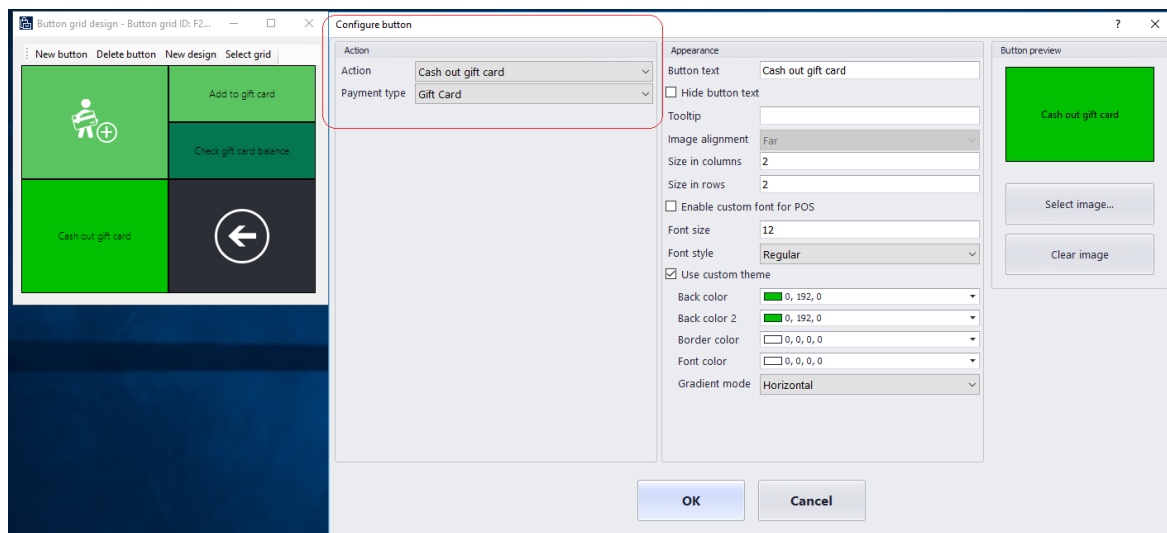
6. Open the **Button grid** page.
7. In the navigation bar on the left side of the page, search for **F2S1M**, and select the filtered option.
8. On the **Action Pane**, select **Designer** to download the button designer application.
9. When the grid designer appears, right-click on an empty (gray) area, and then select **New button**.



10. Right-click the new button, and then select **Button properties**.
11. Set the **Action**, **Cash out gift card**, and **Text on button** properties according to the following matrix.

ACTION	PAYMENT TYPE	TEXT ON BUTTON
Cash out gift card	Gift Card	Cash out gift card

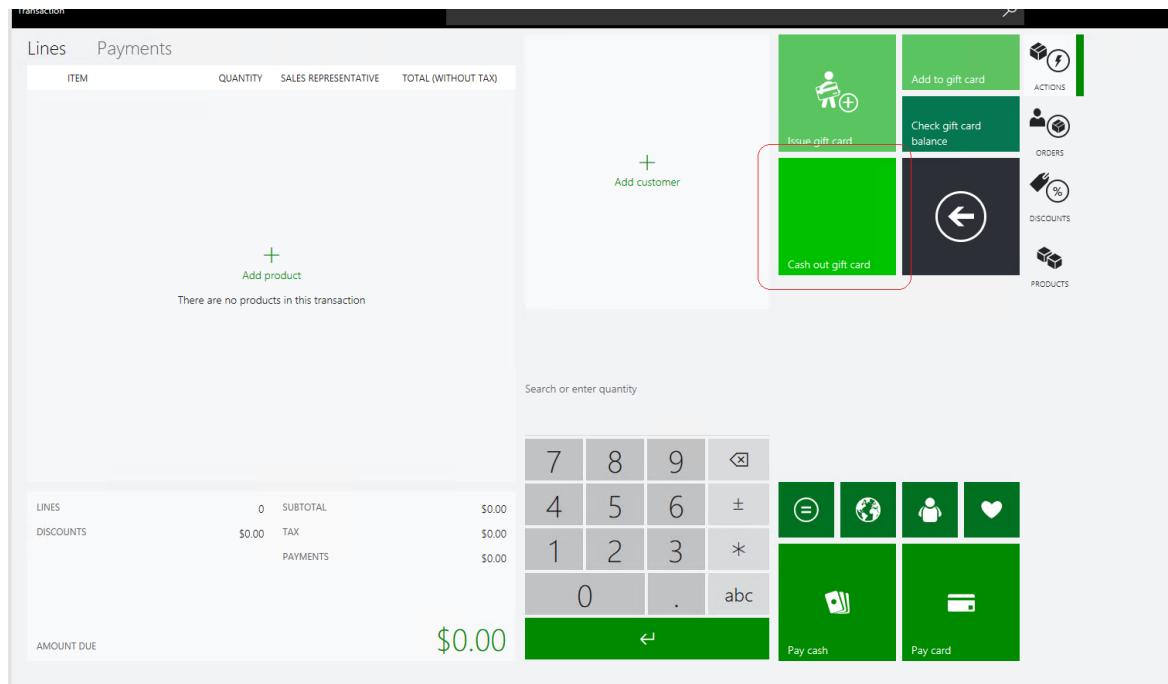
When you've finished, your button layout should resemble the following illustration.



12. Click **Ok** and close the designer.
13. Search for **Distribution Schedule**.
14. In the navigation bar on the left side of the page, search for **1090, 1115, and 1070**.
15. On the **Action Pane**, select **Run now**.
16. Check the status of the job by searching for **Download sessions**.
17. Wait until **Applied** appears next to all the jobs, and then close the browser.

## Configure and test Retail Modern POS

1. Start the Retail Modern POS (MPOS) application.
2. Sign in by using the standard credentials.
3. When you're prompted, select **Perform a non-drawer operation**.
4. On the main screen, select **Select hardware station**.
5. On the bar on the right side of the page, select **Manage**.
6. Turn on **Virtual Peripherals**, and then select **OK**.
7. In the **Available paired stations** field, select **Virtual Peripherals**.
8. You're prompted to either open a new shift or perform non-drawer operations. You can now open a new shift.
9. On the main screen, select **Current transaction**.
10. Select **Gift cards**.
11. Select **Cash out gift card**.
12. Enter or scan the gift number.
13. The line for **gift card cash out** will be added to the **Current transaction** for cash out.
14. Select the **Cash** payment method and the drawer will open when the transaction is completed.



## Troubleshooting

For all general issues, you should always consult the Modern POS or IIS Hardware Station event logs. The logs can be found under these nodes in the Windows event log:

- Application and Services Logs > Microsoft > Dynamics > Commerce-ModernPOS
- Application and Services Logs > Microsoft > Dynamics > Commerce-Hardware Station

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Ship orders from another store by using the Charge send feature

2/18/2021 • 2 minutes to read • [Edit Online](#)

With the Charge send feature in Commerce, customer orders can be placed in one store and shipped from another store.

Customer orders in the point of sale (POS) client support multiple fulfillment options. Some examples of fulfillment options include:

- Pick up from the same store on a different date.
- Pick up from a different store on the same date or a different date.
- Ship from the default shipping warehouse that is assigned to the store, and deliver on a specific date.

The Charge send feature uses the following POS operations: Ship all products and Ship selected products. This allows the store clerk to select the "ship from" location that the order or order line can be fulfilled from. By default, the "ship from" location is the shipping warehouse that is associated with the store. However, the store clerk can change this location and select any store that is defined in the store locator group that is assigned to the store.

The ability to select "ship to" addresses remains unchanged.

The shipping methods that can be used to fulfill the order line are based on the configuration of valid modes of delivery for products and addresses. Because the rules about valid of modes of delivery are maintained only in the Headquarters (HQ), the POS client makes a real-time call to fetch the valid modes of delivery for a ship line.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Hide non-carrier delivery modes from the shipping options in POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes a configuration option that is available for the point of sale (POS) application. This configuration option changes the behavior for the selection of a mode of delivery when shipment orders are created in POS.

When users create customer shipment orders in POS, they can select a mode of delivery for the shipment. This functionality is available regardless of whether the whole order is being shipped or only selected lines.

By default, the dialog box where a mode of delivery is selected shows all the valid modes of delivery for the combination of a channel, an item, and a delivery address. These modes of delivery are defined on the **Modes of delivery** page in Headquarters (**Sales and marketing > Setup > Distribution > Modes of delivery**). "Non-carrier" modes of delivery, such as **Carryout** or **Pickup**, might also appear for selection in the dialog box.

However, a feature has been added that lets you hide non-carrier modes of delivery in the dialog box. To turn on this feature, on the **Commerce parameters** page, on the **Customer orders** tab, set the **Show only carrier mode options for ship orders** option to **Yes**. After you turn on this feature and run the appropriate distribution jobs to sync the information to the channel database, non-carrier modes of delivery won't appear for selection during the process of creating shipment orders in POS.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Task recorder and Help for Retail Modern POS (MPOS) and Cloud POS

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to use Task recorder in Retail Modern POS and Cloud POS.

## Overview

Task recorder in Retail Modern POS or Cloud POS is a new solution that was built with a focus on high responsiveness. It provides a flexible application programming interface (API) for extensibility and seamless integration with consumers of business process recordings. Additionally, Task recorder integration with the Business process modeler (BPM) tool on Microsoft Dynamics Lifecycle Services (<https://bpm.lcs.dynamics.com>) has been brought forward. Therefore, users can continue to produce rich business process diagrams from recordings to analyze and design their applications.

## Architecture

Task recorder can record user actions in the client with exact fidelity. Each control is instrumented to notify Task recorder about the execution of a user action. The control notifies Task recorder that an event occurred and passes along all pertinent information about the corresponding user action in real time. From this information, Task recorder can capture the type of user action (such as a button click, value entry, or navigation) and any data that is related to the user action (such as the input data value and type, form context, or record context). Task recorder persists the information with enough detail to help guarantee that a playback of the recording can perform the recorded actions exactly as the user performed them. (The playback feature isn't yet implemented at Retail modern POS or Cloud POS.)

## Basic configuration

To enable task recording in POS, follow these steps.

1. Click **Retail and Commerce > Channel Setup > POS Setup > Registers**.
2. Click the register to enable task recording on.
3. On the **Register** tab, on the **General** FastTab, set the **Enable task recording** option to **Yes**.
4. Click **Save**.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
6. Select the **Registers (1090)** job, and then click **Run now**.

## Create a recording

Follow these steps to create a new recording using Task recorder.

1. Start Retail Modern POS or Cloud POS, and sign in.
2. On the **Settings** page, in the **Task Recorder** section, click **Open task recorder**. The **Task recorder** pane appears. You can click the **Close** button (X) in the upper-right corner to close the **Task recorder** pane before you begin a new recording. To reopen the pane, repeat step 2.



3. Enter a name and description for the recording, and then click **Start**. The recording session begins as soon as you click **Start**.

**NOTE**

If you click the **Close** button (X) in the upper-right corner while recording is in progress, the **Task recorder** pane is closed, but the recording session isn't ended. To reopen the Task recorder pane, click the **Help** button (question mark) at the top of the screen.

4. After you click **Start**, Task recorder enters recording mode. The **Task recorder** pane shows information and controls that are related to the recording process.
5. Perform the actions that you want to perform in the Retail Modern POS or Cloud POS user interface (UI).
6. To end the recording session, click **Stop**.

## Download options

After you end the recording session, several options are shown, so that you can download your recording.

Save to this PC

Export as Word document

Save as developer recording

Privacy statement: Any information that you enter into the application while you are recording will be captured and included in the recording file, Training document and BPM package. If you decide to share the recording file, training document or BPM package others may be able to see the information that was captured.

### Save to this PC

You can use the recording package to play a Task guide, maintain the recording, or edit the annotations in the

recording. (This feature isn't yet implemented in Retail Modern POS and Cloud POS.)

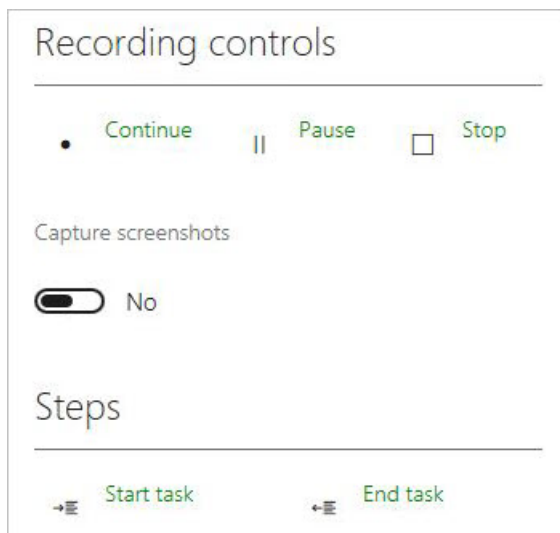
### Export as Word document

You can save the recording as a Microsoft Word document. The document will contain the recorded steps and the screenshots that were captured.

### Save as developer recording

The raw recording file will be useful for developer scenarios such as test code generation. (This feature isn't yet implemented.)

## Recording controls



### Stop

Click **Stop** to end the recording session. Note that you can't restart a session after you end it. Therefore, make sure that the recording is completed before you end it.

### Pause

Click **Pause** to temporarily stop (pause) the recording session and continue with the operation. Steps that you perform after you click **Pause** aren't recorded.

### Continue

To resume the recording session after you've paused it, click **Continue**.

### Capture screenshots

Task recorder can capture screenshots of the Retail Modern POS UI as you record a business process. To turn on the screenshot capture feature, set the **Capture screenshot** option to **Yes** and then make the recording. Once the recording is completed, click **Stop** and download the Word document. The document will contain the steps with relevant screenshots.

#### NOTE

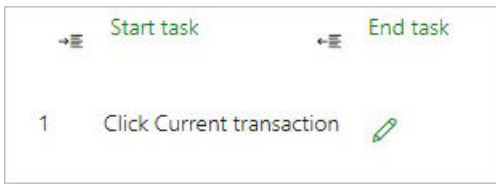
Capture screenshot functionality is not supported in Cloud POS.

### Start task and End task

You can specify the beginning and end of a set of grouped steps by using the **Start task** and **End task** buttons. Click **Start task** to add a "Start Task" step, and then perform the steps that should be included in the group. After you've finished performing the steps for the group, click **End task**. Tasks help you organize your procedures. Tasks can be nested within other tasks. In this way, you can better organize very long and complex business processes.

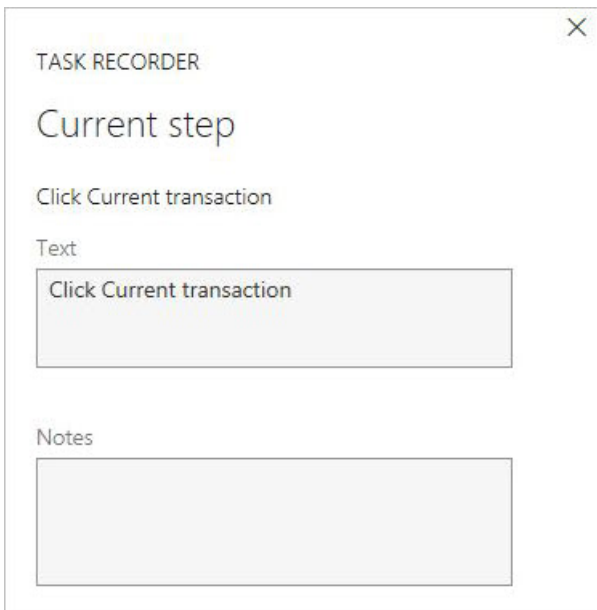
# Adding annotations

An annotation is additional text that you add to a step in a recording. For example, you can use annotations to give the user more context or instructions. You can add annotations before or after a step. You can add an annotation to any step by clicking the **Edit** button (pencil symbol) to the right of the step.



## Texts and notes

You can use the **Texts** and **Notes** fields to add text that should be associated with a step in a Task guide.



### Text

Text that you enter in the **Text** field appears *above* the step text in the Task guide. This location is appropriate for text that you want the user to read before he or she completes the step.

### Notes

Text that you enter in the **Notes** field appears *below* the step text in the Task guide. To read the note text, the user must expand the step text in the pop-up window. This location is appropriate for optional reading material or other information that might be useful to the user, but that the user doesn't require in order to complete the action.

# Help in Retail Modern POS and Cloud POS

To show your own custom task recordings in the Help pane of Retail Modern POS and Cloud POS so that they can be viewed as text, you must save your task recordings to your own BPM library, and then update your Help system parameters to point to your BPM library. For more information, see [Connecting the Help system](#). Retail Modern POS and Cloud POS Help searches LCS in real time. It searches across all the BPM libraries that are selected in the Commerce Help system parameters and shows the relevant results. To access the **Help** menu, click the **Help** button (question mark) at the top of the screen and then in the search box type your process name and hit the search button.



When you click a Task guide in the search results, you can either view the steps as a Help topic or export the

steps to a Word document.

**NOTE**

Help in Retail Modern POS and Cloud POS will not bring up task guides according to what form you're on or operation you're doing. You have to type the process name in the search box and then click **Search**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Peripherals

2/18/2021 • 33 minutes to read • [Edit Online](#)

This topic explains the concepts that are related to store peripherals. It describes the various ways that peripherals can be connected to the point of sale (POS) and the components that are responsible for managing the connection with the POS.

## Concepts

### POS registers

Navigation: Click **Retail and Commerce** > **Channel setup** > **POS setup** > **Registers**. The point of sale (POS) register is an entity that is used to define the characteristics of a specific instance of the POS. These characteristics include the hardware profile or setup for peripherals that will be used at the register, the store that the register is mapped to, and the visual experience for the user who signs in to that register.

### Devices

Navigation: Click **Retail and Commerce** > **Channel setup** > **POS setup** > **Devices**. A device is an entity that represents a physical instance of a device that is mapped to a POS register. When a device is created, it's mapped to a POS register. The device entity tracks information about when a POS register is activated, the type of client that is being used, and the application package that has been deployed to a specific device.

Devices can be mapped to the following application types: Retail Modern POS, Retail Cloud POS, Retail Modern POS – Windows Phone, Retail Modern POS – Android, and Retail Modern POS – iOS.

### Modern POS

Modern POS is the POS program for Microsoft Windows. It can be deployed on Windows 10 operating systems (OSs).

### Cloud POS

Cloud POS is a browser-based version of the Modern POS program that can be accessed in a web browser.

### Modern POS for iOS

Modern POS for iOS is an iOS-based version of the Modern POS program that can be deployed on iOS devices.

### Modern POS for Android

Modern POS for Android is an Android-based version of the Modern POS program that can be deployed on Android devices.

### POS peripherals

POS peripherals are devices that are explicitly supported for POS functions. These peripherals are typically divided into specific classes. For more information about these classes, see the "Device classes" section of this topic.

### Hardware station

Navigation: Click **Retail and Commerce** > **Channels** > **Stores** > **All stores**. Select a store, and then click the **Hardware stations** FastTab. The **Hardware station** setting is a channel-level setting that is used to define instances where the peripheral logic will be deployed. This setting at the channel level is used to determine characteristics of the hardware station. It's also used to list hardware stations that are available for a Modern POS instance in a given store. The hardware station is built into the Modern POS programs for Windows and Android. The hardware station can also be deployed independently as a stand-alone Microsoft Internet

Information Services (IIS) program. In this case, it is accessed via network.

### **Hardware profile**

Navigation: Click **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**. The hardware profile is a list of devices that are configured for a POS register or a hardware station. The hardware profile can be mapped directly to a POS register or a hardware station.

## **Devices classes**

POS peripherals are typically divided into classes. This section describes and gives an overview of the devices that Modern POS supports.

### **Printer**

Printers include traditional POS receipt printers and full-page printers. Printers are supported through Object Linking and Embedding for Retail POS (OPOS) and Microsoft Windows driver interfaces. Up to two printers can be used at the same time. This capability supports scenarios where cash-and-carry customer receipts are printed on receipt printers, whereas customer orders, which carry more information, are printed on a full-page printer. Receipt printers can be connected directly to a computer via USB, connected to a network via Ethernet, or connected via Bluetooth.

### **Scanner**

Up to two bar code scanners can be used at the same time. This capability supports scenarios where a scanner that is more mobile is required in order to scan large or heavy items, whereas a fixed embedded scanner is used for most standard-sized items, to speed up checkout times. Scanners can be supported through OPOS, Universal Windows Platform (UWP), or keyboard wedge interfaces. USB or Bluetooth can be used to connect a scanner to a computer.

### **MSR**

One USB magnetic stripe reader (MSR) can be set up by using OPOS drivers. If you want to use a stand-alone MSR for electronic funds transfer (EFT) payment transactions, the MSR must be managed by a payment connector. Stand-alone MSRs can be used for customer loyalty entry, employee sign-in, and gift card entry, independently of the payment connector.

### **Cash drawer**

Two cash drawers can be supported per hardware profile. This capability enables two active shifts per register to be available at the same time. In the case of a shared shift, or a cash drawer that is used by multiple mobile POS devices at the same time, only one cash drawer is allowed per hardware profile. Cash drawers can be connected directly to a computer via USB, connected to a network, or connected to a receipt printer via an RJ12 interface. In some cases, cash drawers can also be connected via Bluetooth.

### **Line display**

Line displays are used to show products, transaction balances, and other useful information to the customer during a transaction. One line display can be connected to the computer via USB by using OPOS drivers.

### **Signature capture**

Signature capture devices can be connected directly to a computer via USB by using OPOS drivers. When signature capture is configured, the customer is prompted to sign on the device. After the signature is provided, it's shown to the cashier for acceptance.

### **Scale**

Scales can be connected to the computer via USB by using OPOS drivers. When a product that is marked as a "Weighed" product is added to a transaction, the POS reads the weight from the scale, adds the product to the transaction, and uses the quantity that the scale provided.

### **PIN pad**

Personal identification number (PIN) pads are supported through OPOS, but they must be managed via a payment connector.

## Secondary display

When a secondary display is configured, the number 2 Windows display is used to show basic information. The purpose of the secondary display is to support independent software vendor (ISV) extension, because out of the box, the secondary display isn't configurable and shows limited content.

## Payment device

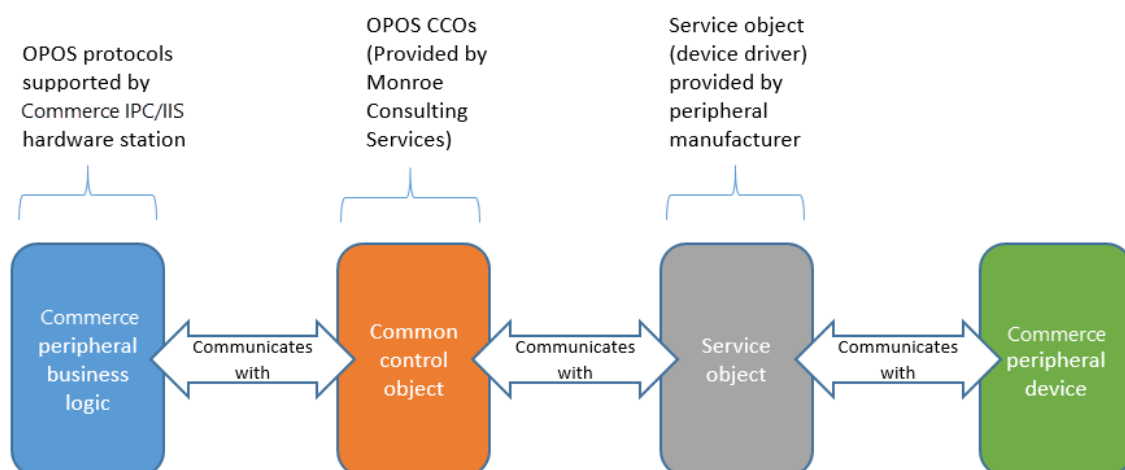
Payment device support is implemented through the payment connector. Payment devices can perform one or many of the functions that other device classes provide. For example, a payment device can function as an MSR/card reader, line display, signature capture device, or PIN pad. Support for payment devices is implemented independently of the stand-alone device support that is provided for other devices that are included in the hardware profile.

# Supported interfaces

## OPOS

To help guarantee that the largest range of devices can be used with Commerce, the OLE for POS industry standard is the primary peripheral device platform that is supported. The OLE for POS standard was produced by the National Retail Federation (NRF), which establishes industry-standard communication protocols for peripheral devices. OPOS is a widely adopted implementation of the OLE for POS standard. It was developed in the mid-1990s and has been updated several times since then. OPOS provides a device driver architecture that enables easy integration of POS hardware with Windows-based POS systems. OPOS controls handle communication between compatible hardware and the POS software. An OPOS control consists of two parts:

- **Control object** – The control object for a device class (such as line displays) provides the interface for the software program. Monroe Consulting Services ([www.monroeecs.com](http://www.monroeecs.com)) provides a standardized set of OPOS control objects that are known as the common control objects (CCOs). The CCOs are used to test the POS component of Commerce. Therefore, the testing helps guarantee that, if Commerce supports a device class through OPOS, many device types can be supported, provided that the manufacturer provides a service object that is built for OPOS. You don't have to explicitly test each device type.
- **Service object** – The service object provides communication between the control object (CCO) and the device. Typically, the service object for a device is provided by the device manufacturer. However, in some cases, you might have to download the service object from the manufacturer's website. For example, a more recent service object might be available. To find the address of the manufacturer's website, see your hardware documentation.



Support for the OPOS implementation of OLE for POS helps guarantee that, if the device manufacturers and



POS publishers implement the standard correctly, POS systems and supported devices can work together, even if they weren't previously tested together.

#### NOTE

OPOS support doesn't guarantee support for all devices that have OPOS drivers. Commerce must first support that device type, or class, through OPOS. In addition, service objects might not always be up to date with the latest version of the CCOs. You should also be aware that, in general, the quality of service objects varies.

## Windows

Receipt printing at the POS is optimized for OPOS. OPOS tends to be much faster than printing through Windows. Therefore, it's a good idea to use OPOS, especially in environments where 40-column receipts are printed and transaction times must be fast. For most devices, you will use OPOS controls. However, some OPOS receipt printers also support Windows drivers. By using a Windows driver, you can access the latest fonts and network one printer for multiple registers. However, there are drawbacks to using Windows drivers. Here are some examples of these drawbacks:

- When Windows drivers are used, images are rendered before printing occurs. Therefore, printing tends to be slower than it is on printers that use OPOS controls.
- Devices that are connected through the printer ("daisy-chained") might not work correctly when Windows drivers are used. For example, the cash drawer might not open, or the slip printer might not work as you expect.
- OPOS also supports a more extensive set of variables that are specific to receipt printers, such as paper cutting or slip printing.
- Windows printers are not supported through the IIS hardware station.

If OPOS controls are available for the Windows printer that you're using, the printer should still work correctly with Commerce.

## Universal Windows Platform

UWP, in the case of peripherals, is related to Windows support for Plug and Play devices. When a Plug and Play device is connected to a Windows OS version that supports that type of device, no driver is required for the device to be used as intended. For example, if Windows detects a Bluetooth speaker device, the OS knows that the device has the **Speaker** class type. Therefore, and it treats that device as a speaker. No additional setup is required. In the case of POS devices, many USB devices can be plugged in, and Windows will recognize them as Human Interface Devices (HIDs). However, it might not be able to determine the capabilities that the device provides, because the device doesn't specify the class, or type, of device. In Windows 10, device classes for bar code scanners and MSRs have been added. Therefore, if a device declares itself to Windows 10 as a device of one of these classes, Windows will listen for events from the device at the appropriate times. Modern POS supports UWP MSRs and scanners. Therefore, when it's ready for input from one of these devices, and a device that belongs to one of these classes is connected, the device can be used. For example, if a UWP bar code scanner is plugged into a Windows 10 computer, and bar code sign-in is configured for Modern POS, the bar code scanner will become active on the sign-in screen. No additional setup is required. Additional classes of point of service UWP devices are being added to Windows. These classes include classes for cash drawers and receipt printers. Support for these new device classes in Modern POS is pending.

## Keyboard wedge

Keyboard wedge devices send data to the computer as if that data were typed on a keyboard. Therefore, by default, the field that is active at the POS will receive the data that is scanned or swiped. In some cases, this behavior can cause the wrong type of data to be scanned into the wrong field. For example, a bar code might be scanned into a field that is intended for input of credit card data. In many cases, there is logic at the POS that determines whether the data that is scanned or swiped is a bar code or card swipe. Therefore, the data is handled correctly. However, when devices are set up as OPOS instead of keyboard wedge devices, there is more

control over how the data from those devices can be consumed, because more is "known" about the device that the data originates from. For example, data from a bar code scanner is automatically recognized as a bar code, and the associated record in the database is found more easily and faster than if a generic string search were used, as in the case of keyboard wedge devices.

### **Native printer**

Native (or "Device" as the type is named in the hardware profile) printers can be configured to prompt the user to select a printer that is configured for the computer. When a printer of the **Device** type is configured, if Modern POS encounters a print command, the user is prompted to select a printer in a list. This behavior differs from the behavior for Windows drivers, because the **Windows** printer type in the hardware profile doesn't show a list of printers. Instead, it requires that a named printer be provided in the **Device name** field.

### **Network**

Network-addressable cash drawers, receipt printers, and payment terminals can be used over a network, either directly through the Interprocess Communications (IPC) hardware station that is built into the Modern POS for Windows application or through the IIS hardware station for other Modern POS clients.

## Hardware station deployment options

### **Dedicated**

Modern POS clients for Windows and Android include **Dedicated** or built-in hardware stations. Those clients can communicate directly with peripherals using business logic that is built into the applications. The Android application only supports network devices. For more information on peripheral support for the Android, visit the [Set up POS hybrid app on Android and iOS](#) article.

To use the dedicated hardware station, assign a hardware profile to a register that will use the Modern POS for Windows or Android applications. Then create a hardware station of the **Dedicated** type for the store where the register will be used. Start the Modern POS in non-drawer mode and use the **Manage hardware stations** operation to turn on the hardware station capabilities, the dedicated hardware station will be active by default. Next, log out of the Modern POS, then log back in and open a shift and the peripherals configured in the hardware profile will be usable.

### **Shared**

Also sometimes referred to as the "IIS" hardware station, "IIS" implying that the POS application connects to the hardware station via Microsoft Internet Information Services. The POS application connects to the IIS hardware station via web services that run on a computer where the devices are connected. When the shared hardware station is used, the peripherals that are connected to a hardware station can be used by any POS register that is on the same network as the IIS hardware station. Because only Modern POS for Windows and Android include built-in support for peripherals, all other Modern POS applications must use the IIS hardware station to communicate with POS peripherals that are configured in the hardware profile. Therefore, each instance of the IIS hardware station requires a computer that runs the web service and application that communicates with the devices.

The shared hardware station can be used to allow multiple point of sale clients to share peripherals or can be used to manage a committed set of peripherals for a single point of sale.

When a hardware station is used to support sharing of peripherals between multiple POS clients, only cash drawers, receipt printers, and payment terminals should be used. You can't directly connect stand-alone bar code scanners, MSRs, line displays, scales, or other devices. Otherwise, conflicts will occur when multiple POS devices try to claim those peripherals at the same time. Here is how conflicts are managed for supported devices:

- **Cash drawer** – The cash drawer is opened via an event that is sent to the device. The only issue that can occur when a cash drawer is called occurs if the cash drawer is already open. In the case of shared hardware stations, the cash drawer should be set to **Shared** in the hardware profile. This setting prevents the POS

from checking whether the cash drawer is already open when it sends open commands.

- **Receipt printer** – If two receipt printing commands are sent to the hardware station at the same time, one of the commands can be lost, depending on the device. Some devices have internal memory or pooling that can prevent this issue. If a print command isn't successful, the cashier receives an error message and can retry the print command from the POS.
- **Payment terminal** – If a cashier tries to tender a transaction on a payment terminal that is already being used, a message notifies the cashier that the terminal is being used and asks the cashier to try again later. Usually, cashiers can see that a terminal is already being used and will wait until the other transaction is completed before they try to tender again.

Validation is planned for a future release, to detect whether unsupported devices are set up for a hardware profile that is mapped to a shared hardware station. If any unsupported devices are detected, the user will receive a message that states that the devices aren't supported for shared hardware stations. In the case of shared hardware stations, the **Select upon tendering** option is set to **Yes** at the register level. The POS user is then prompted to select a hardware station when a tender is selected for a transaction at the POS. When the hardware station is selected only at the time of tender, the hardware station selection is added directly to the POS workflow for mobile scenarios. As an additional benefit, the line display on the payment terminal isn't used for shared scenarios. If the payment terminal is used as a line display, other users might be blocked from using that terminal until the transaction is completed. In mobile scenarios, lines might be added to a transaction over a longer period. Therefore, the **Select upon tendering** option is required in order to ensure optimum device availability.

### **Network peripherals**

The network designation for devices in the hardware profile enables cash drawers, receipt printers, and payment terminals to be connected via a network connection.

### **Modern POS for Windows**

You can specify IP addresses for network peripherals in two places. If the Modern POS Windows client is using a single set of network peripherals, you should set the IP addresses for those devices by using the **IP configuration** option on the Action Pane for the register itself. In the case of network devices that will be shared among POS registers, a hardware profile that has network devices assigned to it can be mapped directly to a shared hardware station. To assign IP addresses, select that hardware station on the **Stores** page, and then use the **IP configuration** option in the **Hardware stations** section to specify the network devices that are assigned to that hardware station. For hardware stations that have only network devices, you don't have to deploy the hardware station itself. In this case, the hardware station is required only in order to conceptually group network-addressable devices according to their location in the store.

### **Cloud POS and Modern POS for iOS**

The logic that drives physically connected and network-addressable peripherals is contained in the hardware station. Therefore, for all POS clients except Modern POS for Windows and Android, an IIS hardware station must be deployed and active to enable the POS to communicate with peripherals, regardless of whether those peripherals are physically connected to a hardware station or addressed over the network.

## Setup and configuration

### **Hardware station installation**

For information, see [Configure and install hardware station](#).

### **Modern POS for Windows setup and configuration**

For information, see [Configure, install and activate Modern POS \(MPOS\)](#).

### **Modern POS for Android and iOS setup and configuration**

For information, see [Set up POS hybrid app on Android and iOS](#).

## OPOS device setup and configuration

For more information about OPOS components, see the "Supported interfaces" section of this document. Typically, OPOS drivers are provided by the device manufacturer. When an OPOS device driver is installed, it adds a key to the Windows registry in one of the following locations:

- **32-bit system:** HKEY\_LOCAL\_MACHINESOFTWAREOLEforRetailServiceOPOS
- **64-bit system:** HKEY\_LOCAL\_MACHINESOFTWAREWOW6432NodeOLEforRetailServiceOPOS

Within the ServiceOPOS registry location, configured devices are organized according to the OPOS device class. Multiple device drivers are saved.

## Supported scenarios by hardware station type

### Client support – IPC hardware station vs. IIS hardware station

The following table shows the topologies and deployment scenarios that are supported.

CLIENT	IPC HARDWARE STATION	IIS HARDWARE STATION
Windows app	Yes	Yes
Cloud POS	No	Yes
Android	Yes	Yes
iOS	No	Yes

### Network peripherals

Network peripherals can be supported directly through the hardware station that is built into the Modern POS for Windows and Android applications. For all other clients, you must deploy an IIS hardware station.

CLIENT	IPC HARDWARE STATION	IIS HARDWARE STATION
Windows app	Yes	Yes
Cloud POS	No	Yes
Android	Yes	Yes
iOS	No	Yes

## Supported device types by hardware station type

### Modern POS for Windows with an IPC (built-in) hardware station

SUPPORTED DEVICE CLASS	SUPPORTED INTERFACES
Printer	<ul style="list-style-type: none"><li>• OPOS</li><li>• Windows driver</li><li>• Device</li><li>• Network</li></ul>

SUPPORTED DEVICE CLASS	SUPPORTED INTERFACES
Printer 2	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Windows driver</li> <li>• Device</li> <li>• Network</li> </ul>
Line display	OPOS
Dual display	Windows driver
MSR	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• UWP (No setup is required.)</li> <li>• Keyboard wedge (No setup is required.)</li> </ul>
Drawer	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul> <p><b>Note:</b> Only one drawer can be set up if <b>Use shared shift</b> is configured on the drawer.</p>
Drawer 2	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul> <p><b>Note:</b> Only one drawer can be set up if <b>Use shared shift</b> is configured on the drawer.</p>
Scanner	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• UWP (No setup is required.)</li> <li>• Keyboard wedge (No setup is required.)</li> </ul>
Scanner 2	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• UWP (No setup is required.)</li> <li>• Keyboard wedge (No setup is required.)</li> </ul>
Scale	OPOS
PIN pad	OPOS (Support is provided through customization of the payment connector.)
Signature capture	OPOS
Payment terminal	<ul style="list-style-type: none"> <li>• Custom device support</li> <li>• Network (For more information, see the payment connector documentation.)</li> </ul>

### All Modern POS clients that have a committed "Shared" IIS hardware station

#### NOTE

When the IIS hardware station is "committed" there is a one-to-one relationship between the POS client and the hardware station.

SUPPORTED DEVICE CLASS	SUPPORTED INTERFACES
Printer	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul>
Printer 2	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul>
Line display	OPOS
MSR	OPOS
Drawer	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul> <p><b>Note:</b> Only one drawer per hardware profile can be set up if <b>Use shared shift</b> is configured on the drawer.</p>
Drawer 2	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul>
Scanner	OPOS
Scanner 2	OPOS
Scale	OPOS
PIN pad	OPOS (Support is provided through customization of the payment connector.)
Sig. capture	OPOS
Payment terminal	<ul style="list-style-type: none"> <li>• Custom device support</li> <li>• Network (For more information, see the payment connector documentation.)</li> </ul>

### All Modern POS clients shared an IIS hardware station

#### NOTE

When the IIS hardware station is "shared," multiple devices can use the hardware station at the same time. For this scenario, you should use only the devices that are listed in the following table. If you try to share devices that aren't listed here, such as bar code scanners and MSRs, errors will occur when multiple devices try to claim the same peripheral. In the future, such a configuration will be explicitly prevented.

SUPPORTED DEVICE CLASS	SUPPORTED INTERFACES
Printer	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul>

SUPPORTED DEVICE CLASS	SUPPORTED INTERFACES
Printer 2	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul>
Drawer	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul> <p><b>Note:</b> Only one drawer per hardware profile can be set up if <b>Use shared shift</b> is configured on the drawer.</p>
Drawer 2	<ul style="list-style-type: none"> <li>• OPOS</li> <li>• Network</li> </ul>
Payment terminal	<ul style="list-style-type: none"> <li>• Custom device support</li> <li>• Network (For more information, see the payment connector documentation.)</li> </ul>

## Configuration for supported scenarios

For more information about how to create hardware profiles, see [Define and maintain channel clients, including registers and hardware stations](#).

### Modern POS for Windows with an IPC (built-in) hardware station

This configuration is the most typical configuration for traditional, fixed POS registers. For this scenario, the hardware profile information is mapped directly to the register itself. The EFT terminal number should also be set on the register itself. To set up this configuration, follow these steps.

1. Create a hardware profile where all the required peripherals are configured.
2. Map the hardware profile to the POS register.
3. Create a hardware station of the **Dedicated** type for the store where the POS register will be used. A description is optional.

#### NOTE

You don't have to set any other properties on the hardware station. All other required information, such as the hardware profile, will come from the register itself.

4. Click **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
5. Select the **1090** distribution schedule to sync the new hardware profile to the store. Click **Run now** to sync changes to the POS.
6. Select the **1040** distribution schedule to sync the new hardware station to the store. Click **Run now** to sync changes to the POS.
7. Install and activate Modern POS for Windows.
8. Start Modern POS for Windows, and begin to use the connected peripheral devices.

### Modern POS for Android with an IPC (built-in) hardware station

**New for 10.0.8** - Epson network printers and cash drawers connected to those printers via DK port are now supported for the Modern POS for Android app. For details, visit the [Set up POS hybrid app on Android and iOS](#) article.

### All Modern POS clients that have a committed, shared IIS hardware station

This configuration can be used for all Modern POS clients that have a hardware station that is used exclusively by one POS register. To set up this configuration, follow these steps.

1. Create a hardware profile where all the required peripherals are configured.
2. Create a hardware station of the **Dedicated** type for the store where the POS register will be used.
3. On the dedicated hardware station, set the following properties:

- **Host name** – The name of the host computer where the hardware station will run.

#### NOTE

Cloud POS can resolve **localhost** to determine the local computer where Cloud POS is running. However, the certificate that is required in order to pair Cloud POS with the hardware station must also have "localhost" as the computer name. To avoid issues, we recommend that you list an instance of each dedicated hardware station for the store, as required. For each hardware station, the host name should be the specific computer name where the hardware station will be deployed.

- **Port** – The port to use for the hardware station to communicate with the Modern POS client.
  - **Hardware profile** – If the hardware profile isn't provided on the hardware station itself, the hardware profile that is assigned to the register will be used.
  - **EFT POS number** – The EFT terminal ID to use when EFT authorizations are sent. This ID is provided by the credit card processor.
  - **Package name** – The hardware station package to use when the hardware station is deployed.
4. Click **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
  5. Select the **1090** distribution schedule to sync the new hardware profile to the store. Click **Run now** to sync changes to the POS.
  6. Select the **1040** distribution schedule to sync the new hardware station to the store. Click **Run now** to sync changes to the POS.
  7. Install the hardware station. For more information about how to install the hardware station, see [Configure and install Retail hardware station](#).
  8. Install and activate Modern POS. For more information about how to install Modern POS, see [Configure, install and activate Modern POS \(MPOS\)](#).
  9. Sign in to Modern POS, and select **Perform non-drawer operations**.
  10. Start the **Manage hardware stations** operation.
  11. Click **Manage**.
  12. On the hardware station management page, set the option to turn on the hardware station.
  13. Select the hardware station to use, and then click **Pair**.
  14. After the hardware station is paired, click **Close**.
  15. On the hardware station selection page, click the recently selected hardware station to make it active.



## All Modern POS clients that have a shared IIS hardware station

This configuration can be used for all Modern POS clients that share hardware stations with other devices. To set up this configuration, follow these steps.

1. Create a hardware profile where the required peripherals are configured.
2. Create a hardware station of the **Shared** type for the store where the POS register will be used.
3. On the shared hardware station, set the following properties:
  - **Host name** – The name of the host computer where the hardware station will run.
  - **Description** – Text that will help identify the hardware station, such as **Returns** or **Front of store**.
  - **Port** – The port to use for the hardware station to communicate with the Modern POS client.
  - **Hardware profile** – For shared hardware stations, each hardware station should have a hardware profile. Hardware profiles can be shared among hardware stations, but they must be mapped to each hardware station. In addition, we recommend that you use shared shifts when multiple devices use the same shared hardware station. To set up a shared shift, click **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**. For each shared hardware profile, select the cash drawer, and set the **Shared shift drawer** option to **Yes**.
  - **EFT POS number** – The EFT terminal ID to use when EFT authorizations are sent. This ID is provided by the credit card processor.
  - **Package name** – The hardware station package to use when the hardware station is deployed.
4. Repeat steps 2 and 3 for each additional hardware station that is required in the store.
5. Click **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
6. Select the **1090** distribution schedule to sync the new hardware profile to the store. Click **Run now** to sync changes to the POS.
7. Select the **1040** distribution schedule to sync the new hardware station to the store. Click **Run now** to sync changes to the POS.
8. Install the hardware station on each host computer that you set up in steps 2 and 3. For more information about how to install the hardware station, see [Configure and install Retail hardware station](#).
9. Install and activate Modern POS. For more information about how to install Modern POS, see [Configure, install, and activate Modern POS \(MPOS\)](#).
10. Sign in to Modern POS, and select **Perform non-drawer operations**.
11. Start the **Manage hardware stations** operation.
12. Click **Manage**.
13. On the hardware station management page, set the option to turn on the hardware station.
14. Select the hardware station to use, and then click **Pair**.
15. Repeat step 14 for each hardware station that Modern POS will use.
16. After all the required hardware stations are paired, click **Close**.
17. On the hardware station selection page, click the recently selected hardware station to make it active.

#### NOTE

If devices often use different hardware stations, we recommend that you configure Modern POS to prompt cashiers to select a hardware station when they begin the tender process. Click **Retail and Commerce** > **Channel setup** > **POS setup** > **Registers**. Select the register, and then set the **Select upon tender** option to **Yes**. Use the **1090** distribution schedule to sync changes to the channel database.

## Extensibility

For information about extensibility scenarios for the hardware station, see [Hardware Station extensibility](#).

## Security

According to current security standards, the following settings should be used in a production environment:

### Hardware station installer

The hardware station installer will automatically make these registry edits as part of the installation through self-service.

- Secure Sockets Layer (SSL) should be disabled.
- Only Transport Layer Security (TLS) version 1.2 (or the current highest version) should be enabled and used.

### SSL and TLS

By default, SSL and all version of TLS except TLS 1.2 are disabled. To edit or enable these values, follow these steps: 1. Press the Windows logo key+R to open a **Run** window. 2. In the **Open** field, type **Regedit**, and then click **OK**. 3. If a **User Account Control** message box appears, click **Yes**. 4. In the **Registry Editor** window, navigate to **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\SecurityProviders\SCHANNEL\Protocols**. The following keys have been automatically entered to allow for TLS 1.2 only: - TLS 1.2Server:Enabled=1 - TLS 1.2Server:DisabledByDefault=0 - TLS 1.2Client:Enabled=1 - TLS 1.2Client:DisabledByDefault=0 - TLS 1.1Server:Enabled=0 - TLS 1.1Client:Enabled=0 - TLS 1.0Server:Enabled=0 - TLS 1.0Client:Enabled=0 - SSL 3.0Server:Enabled=0 - SSL 3.0Client:Enabled=0 - SSL 2.0Server:Enabled=0 - SSL 2.0Client:Enabled=0

- No additional network ports should be open, unless they are required for known, specified reasons.
- Cross-origin resource sharing must be disabled and must specify the allowed origins that are accepted.
- Only trusted certificate authorities should be used to obtain certificates that will be used on computers that run the hardware station.

#### NOTE

It's very important that you review security guidelines for IIS and the Payment Card Industry (PCI) requirements.

## Peripheral simulator

For information, see [Peripheral simulator for Commerce](#).

## Microsoft-tested peripheral devices

### IPC (built-in) hardware station

The following peripherals were tested by using the IPC hardware station that is built into Modern POS for Windows.

#### Printer

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Epson	Tm-T88IV	OPOS	
Epson	TM-T88V	OPOS	
Epson	TM-T88	Custom	Connected via network
Star	TSP650II	Custom	Connected via network
Star	mPOP	OPOS	Connected via Bluetooth
HP	F7M67AA	OPOS	Powered USB

#### Bar code scanner

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Motorola	DS9208	OPOS	
Honeywell	1900	UWP	
Symbol	LS2208	OPOS	
HP Integrated	E1L07AA	OPOS	
Datalogic	Magellan 8400	OPOS	

#### PIN pad

MANUFACTURER	MODEL	INTERFACE	COMMENTS
VeriFone	1000SE	OPOS	Requires customization of the payment connector

#### Payment terminal

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Equinox	L5300	Custom	Requires customization of the payment connector
VeriFone	MX925	Custom	Requires customization of the payment connector; connected via network and USB
VeriFone	MX915	Custom	Requires customization of the payment connector; connected via network and USB

#### Cash drawer

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Star	mPOP	OPOS	Connected via Bluetooth
APG	Atwood	Custom	Connected via network
Star	SMD2-1317	OPOS	
HP	QT457AA	OPOS	
Epson		Custom	Connected to network Epson printer via DK port

#### Line display

MANUFACTURER	MODEL	INTERFACE	COMMENTS
HP integrated	G6U79AA	OPOS	
Epson	M58DC	OPOS	

#### Signature capture

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Scriptel	ST1550	OPOS	

#### Scale

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Datalogic	Magellan 8400	OPOS	

#### MSR

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Magtek	21073075	UWP	
Magtek	21073062	OPOS	
HP	IDRA-334133	OPOS	

#### Dedicated IIS hardware station

The following peripherals were tested by using a dedicated (not shared) IIS hardware station together with Modern POS for Windows and Cloud POS.

#### Printer

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Epson	Tm-T88IV	OPOS	
Epson	TM-T88V	OPOS	

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Epson	TM-T88V	Custom	Connected via network
Star	TSP650II	Custom	Connected via network
HP	F7M67AA	OPOS	Powered USB

#### Bar code scanner

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Motorola	DS9208	OPOS	
Symbol	LS2208	OPOS	
HP Integrated	E1L07AA	OPOS	

#### PIN pad

MANUFACTURER	MODEL	INTERFACE	COMMENTS
VeriFone	1000SE	OPOS	Requires customization of the payment connector

#### Payment terminal

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Equinox	L5300	Custom	Requires customization of the payment connector
VeriFone	MX925	Custom	Requires customization of the payment connector; connected via network and USB
VeriFone	MX915	Custom	Requires customization of the payment connector; connected via network and USB

#### Cash drawer

MANUFACTURER	MODEL	INTERFACE	COMMENTS
APG	Atwood	Custom	Connected via network
Star	SMD2-1317	OPOS	
HP	QT457AA	OPOS	
Epson		Custom	Connected to network Epson printer via DK port

#### Line display

MANUFACTURER	MODEL	INTERFACE	COMMENTS
HP integrated	G6U79AA	OPOS	
Epson	M58DC	OPOS	

#### Signature capture

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Scriptel	ST1550	OPOS	

#### Scale

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Datalogic	Magellan 8400	OPOS	

#### MSR

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Magtek	21073075	UWP	
Magtek	21073062	OPOS	
HP	IDRA-334133	OPOS	

#### Shared IIS hardware station

The following peripherals were tested by using a shared IIS hardware station together with Modern POS for Windows and Cloud POS.

#### NOTE

Only a printer, payment terminal, and cash drawer are supported.

#### Printer

MANUFACTURER	MODEL	INTERFACE	COMMENTS
Epson	TM-T88IV	OPOS	
Epson	TM-T88V	OPOS	
Epson	TM-T88	Custom	Connected via network
Star	TSP650II	Custom	Connected via network
HP	F7M67AA	OPOS	Powered USB

#### Payment terminal

MANUFACTURER	MODEL	INTERFACE	COMMENTS
VeriFone	MX925	Custom	Requires customization of the payment connector; connected via network and USB
VeriFone	MX915	Custom	Requires customization of the payment connector; connected via network and USB

#### Cash drawer

MANUFACTURER	MODEL	INTERFACE	COMMENTS
APG	Atwood	Custom	Connected via network
Star	SMD2-1317	OPOS	
HP	QT457AA	OPOS	
Epson		Custom	Connected to network Epson printer via DK port

## Troubleshooting

### Modern POS can detect the hardware station in its list for selection, but it can't complete the pairing

**Solution:** Verify the following list of potential failure points:

- The computer that is running Modern POS trusts the certificate that is used on the computer that runs the hardware station.
  - To verify this setup, in a web browser, go to the following URL: <https://<Computer Name>:<Port Number>/HardwareStation/ping>.
  - This URL uses a ping to verify that the computer can be accessed, and the browser indicates whether the certificate is trusted. (For example, in Internet Explorer, a lock icon appears in the address bar. When you click this icon, Internet Explorer verifies whether the certificate is currently trusted. You can install the certificate on the local computer by viewing the details of the certificate that is shown.)
- On the computer that runs the hardware station, the port that will be used by the hardware station is opened in the firewall.
- The hardware station has correctly installed merchant account information through the Install merchant information tool that runs at the end of the hardware station installer.

### Modern POS can't detect the hardware station in its list for selection

**Solution:** Either of the following factors can cause this issue:

- The hardware station hasn't been set up correctly in headquarters. Use the steps earlier in this topic to verify that the hardware station profile and the hardware station are correctly entered.
- The jobs haven't been run to update the channel configuration. In this case, run the 1070 job for channel configuration.

### Modern POS doesn't reflect new cash drawer settings

**Solution:** Close the current batch. Changes to the cash drawer aren't updated to Modern POS until the current batch is closed.

## Modern POS is reporting an issue with a peripheral

**Solution:** Here are some typical causes of this issue:

- Make sure that other device driver configuration utilities are closed. If these utilities are open, they might prevent Modern POS or the hardware station from claiming the device.
- If the peripheral is shared with multiple POS devices, make sure that it belongs to one of the following categories:
  - Cash drawer
  - Receipt printer
  - Payment terminal

If the peripheral doesn't belong to one of these categories, the hardware station isn't designed to enable the peripheral to be shared among multiple POS devices.

- Sometimes, device drivers can cause the common control objects (CCOs) to stop working correctly. If a device has recently been installed, but it isn't working properly or you notice other issues, you can often resolve the issue by reinstalling the CCOs. To download the CCOs, visit [http://monroecs.com/oposccos\\_current.htm](http://monroecs.com/oposccos_current.htm).
- If you make frequent peripheral changes during testing or troubleshooting, you might have to reset IIS instead of waiting for the cache to refresh itself. To reset IIS, follow these steps:
  1. From the **Start** menu, type **CMD**.
  2. In the search results, right-click **Command prompt**, and then click **Run as administrator**.
  3. In the **Command prompt** window, type **iisreset /Restart** and then press Enter.
  4. After IIS has restarted, restart Modern POS.
- While you're making frequent changes to peripheral devices, if you also frequently start and exit the POS client, the dllhost process from a previous POS session can interfere with the current session. In this case, a device might not be usable until you close the dynamic-link library (DLL) host that is managing the previous session. To close the DLL host, follow these steps:
  1. From the **Start** menu, type **Task manager**.
  2. In the search results, click **Task manager**.
  3. In Task manager, on the **Details** tab, click the column header that is labeled **Name** to sort the table alphabetically by name.
  4. Scroll down until you find dllhost.exe.
  5. Select each DLL host, and then click **End task**.
  6. After the DLL hosts have been closed, restart Modern POS.

## Additional resources

[Peripheral simulator for Commerce](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Peripheral simulator for Commerce

2/18/2021 • 28 minutes to read • [Edit Online](#)

The peripheral simulator is a utility that Microsoft provides as part of Microsoft Dynamics 365 Commerce and as a standalone utility. The utility has two primary components, a *virtual peripheral simulator* and a *point of sale (POS) simulator*.

The virtual peripheral simulator is provided primarily to support testing of scenarios that usually require physical POS peripheral devices. The POS simulator is used to test the compatibility of physical peripheral devices without having to deploy the POS client.

## Install the peripheral simulator

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**.
2. Click **Download**, and then click **PeripheralSimulator**.

### NOTE

You must turn off pop-up blockers before you can download the peripheral simulator.

3. After the download is completed, open the **Downloads** folder, and double-click **VirtualPeripherals.msi** to start the installer.
4. Install the peripheral simulator by using the default settings.

Besides the peripheral simulator, you must install the common control objects from Monroe Consulting Services. Otherwise, the peripheral simulator won't work correctly. To download the common control objects, go to [http://monroecs.com/oposccos\\_current.htm](http://monroecs.com/oposccos_current.htm).

## Virtual peripheral simulator

The virtual peripheral simulator helps you set up, test, and troubleshoot peripheral devices. It can be used to streamline the testing of peripherals, and to isolate issues that are caused by incorrect setup or malfunctioning device drivers. The peripheral simulator includes a desktop program that features virtual versions of devices that Commerce supports. A section for each virtual device shows the interaction between the device and the POS. You can also use it to provide input that is valid for various POS scenarios. The peripheral simulator supports interaction between the POS and the following virtual devices:

- **Printer** – The virtual peripheral simulator can show receipts that are configured for a POS printer.
- **Line display** – You can configure a virtual line display to show activity on a physical line display.
- **Magnetic stripe reader (MSR)** – You can send simulated magnetic stripe events to the POS from the virtual peripheral simulator.
- **Drawer** – You can simulate a physical cash drawer.
- **Drawer 2** – By setting up a second cash drawer in the peripheral simulator, you can simulate scenarios that involve a single POS register that has two active shifts.
- **Scanner** – The virtual bar code scanner that the virtual peripheral simulator supports can issue bar code scan events.

- **Scale** – A virtual scale lets you simulate the interaction of weighed items with the POS.
- **Personal identification number (PIN) pad** – You can simulate PIN pad operations.

#### NOTE

You must implement support for a physical PIN pad through the payment connector.

- **Signature capture** – The virtual peripheral simulator includes a virtual signature capture device that you can set up to prompt for signatures that are required for some tenders, such as customer account payments.
- **Payment terminal** – A virtual payment terminal can be used in conjunction with a custom payment connector to test application programming interface (API) calls from the POS to a virtual payment device. To use the virtual payment terminal, you must implement a payment connector. For more information, see <Implementing a payment connector and payment device (white paper)>.

You can also use the virtual peripheral simulator to simulate keyboard wedge events that originate from a bar code scanner and MSR. The virtual peripheral simulator specifically supports Object Linking and Embedding for Retail POS (OPOS) devices.

### Key scenarios – Virtual peripheral simulator

#### Troubleshooting

You can use the peripheral simulator to troubleshoot device setup. If you don't have the peripheral simulator or a second device of the same class, it can be difficult to determine where issues originate. However, when you have the peripheral simulator, you can set up virtual devices, and run the same code paths and business logic that are used for physical devices. From the perspective of the peripheral simulator, the main difference between virtual devices and physical devices is the service object, or device driver. For physical devices, the service object is provided by the device manufacturer. By contrast, for virtual devices, the service objects are provided as part of the peripheral simulator. When the peripheral simulator is working correctly, if a device doesn't work correctly after the device name in the hardware profile is changed to the name of a real device, you can assume that there's an issue with the service object that the manufacturer provided.

#### Training

You can use the peripheral simulator to add a realistic layer to cashier training when a physical hardware setup isn't available. When the peripheral simulator is included in training scenarios, the cashier can more effectively interact with the POS by providing input such as product bar code scans and gift card swipes, and by observing which receipts are printed for a specific transaction.

#### Testing

You can use the peripheral simulator to test product bar codes, receipt formats, and so on, without having to deploy physical hardware in a virtual environment. Because physical hardware isn't required, and you don't have to deploy a POS client on a hardware station or a physical computer, you can more quickly test changes that are made in the back office.

### Set up the virtual peripheral simulator

#### Set up a hardware profile

1. To set up the peripheral simulator, go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**.
2. Click **New** to create a profile.
3. Enter values in the **Profile number** and **Description** fields.
4. Use the following table to set up the virtual devices that must be tested. Here is an explanation of the columns in the table:

- **Device** – This column gives the name of the FastTab where you set up the device.
- **Device type** – This column gives the value that you select in the field that is labeled with the name of the device.
- **Device name** – This column gives the exact value that you enter for the device name.

**IMPORTANT**

The device names that are given here are required, because the hardware station uses these specific names to address the devices. If you don't use the following specific names, the device won't be usable.

No specific setup in the hardware profile is required in order to simulate keyboard wedge events from the bar code scanner and MSR.

DEVICE	DEVICE TYPE	DEVICE NAME
Printer	OPOS	MockOPOSPrinter
Line display	OPOS	MockOPOSLineDisplay
MSR	OPOS	MockOPOSMSR
Drawer	OPOS	MockOPOSDrawer1
Drawer2	OPOS	MockOPOSDrawers
Scanner	OPOS	MockOPOSScanner
Scale	OPOS	MockOPOSScale
PIN Pad	OPOS	MockOPOSPinPad
Signature capture	OPOS	MockOPOSSignatureCapture

**Assign the hardware profile to a register**

1. After the hardware profile is created, go to **Retail and Commerce > Channel setup > POS setup > Registers**.
2. In the **POS registers** list, click the link in the **Register number** field for the register that should use the peripheral simulator.
3. Click **Edit**.
4. In the **Profiles** section, in the **Hardware profile** field, select the hardware profile that you created for virtual peripherals.
5. Click **Save**.

**Synchronize changes to the channel database**

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Select the **1090** distribution schedule.
3. Click **Run now** to synchronize changes to the POS.

After the data is synchronized, the new hardware profile and changes on the register are available in the channel database.

# Using the virtual peripheral simulator

To start the virtual peripheral simulator, click **Start** on your computer, type **Peripheral simulator**, and then select the app when it appears in the search results. After you start the peripheral simulator, click **Use virtual peripherals**. The supported devices will be listed as tabs on the left side of the window. To view a specific device, click the tab for that device.

## Line display capabilities

When the virtual line display is configured, it shows line items as they are scanned in the POS transaction. In addition to line items, the display shows the total that is due when a tender is selected at the POS. It also shows the balance that is due if a tender is entered but a balance is still due for the transaction. When the POS isn't being used, a message can be shown to indicate that the till is closed. You must configure the message on the **Line display** FastTab in the hardware profile.

## Cash drawer capabilities

When the hardware profile is configured to use virtual cash drawers, the POS opens the cash drawer for the active shift in response to drawer operations such as tender declarations. The POS also opens the cash drawer so that the cashier can make change or deposit cash during standard cash-and-carry transactions. The virtual cash drawers have the labels **Main drawer** and **Secondary drawer**. These labels represent Drawer and Drawer 2 in the hardware profile, respectively. When a drawer is closed, an image of a closed cash drawer is shown, and the button on the closed cash drawer is labeled **Open drawer**. If you click this button, the image is replaced with an image of an open cash drawer to indicate that the drawer is now open. The button on the open cash drawer is labeled **Close drawer**.

Several operations at the POS can cause the cash drawer to open. Most operations can't proceed while the cash drawer is open. The exceptions are some end-of-day procedures. If the POS user receives an error message that states that an operation can't be performed while the cash drawer is open, the user must close the virtual or physical cash drawer to proceed. If a cash drawer is marked as **Shared** in the hardware profile, the system doesn't verify that the drawer is closed before an operation. The operation proceeds as usual, even if the cash drawer is open. This behavior supports scenarios where cash drawers are shared among sales associates, and where one associate uses a cash drawer while another associate performs unrelated tasks on their own POS device. Changes that are made to the cash drawer aren't evident until the current shift is closed and a new shift is opened.

## MSR capabilities

The peripheral simulator provides robust support for virtual MSR operations by working in either OPOS mode or keyboard wedge mode. OPOS mode requires that the MSR be configured in the hardware profile to work as an OPOS device. Keyboard wedge mode just sends keyboard wedge data events to Microsoft Windows. Besides differences in setup, OPOS and keyboard wedge modes differ in the following ways:

- The POS client enables OPOS MSR devices for specific scenarios, such as scenarios that allow for magnetic stripe data for loyalty or gift card entry.
- In keyboard wedge mode, the peripheral simulator sends keyboard wedge data to the field that is active when the data is sent. This behavior resembles the behavior that occurs if the data is entered by using a keyboard. To use the MSR as a keyboard wedge, the user must switch to Modern POS (MPOS) to make sure that data is received in the correct field. Therefore, you can configure a delay, so that the user has time to make sure that the data will be sent to the correct field.

## Testing gift and payment card swipes

The virtual MSR that the peripheral simulator provides also lets you configure specific MSR data to test scenarios for gift and payment card swipes. To create a card, click the plus sign (+) button, and select the type of card. Then specify the card number or track data that should be sent to the POS, together with the expiration month and year for the card that you're defining. The value that you select in the **Type of card** field is just a label that can be mapped to a card. This label makes it easier to identify cards when they are swiped through the

peripheral simulator. You can select cards that have been configured in the peripheral simulator by using the left arrow (<) and right arrow (>) buttons above the image of the card. You can edit and delete cards by using the **Edit** and **Delete** buttons next to the plus sign (+) button.

### **PIN pad**

You can configure the PIN pad simulator to simulate an OPOS PIN pad. When an electronic funds transfer (EFT) transaction is performed at the POS and requires that a PIN be entered, the hardware station calls the PIN device to prompt for PIN entry. To work, the PIN pad in the peripheral simulator must support an EFT payment connector.

### **Printer**

The virtual peripheral printer just shows receipts as they are printed from the POS. If a print operation produces multiple receipts, you can scroll through the receipts.

#### **Configure receipt printing**

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**.
2. Select the hardware profile that you created for virtual peripherals.
3. On the **Printer** FastTab, click **Edit**.
4. In the **Receipt profile ID** field, select a receipt profile.
5. Click **Save**.

#### **Scale**

When a scale product is added to the POS transaction, and a scale is configured, the POS retrieves the weight from the scale. For both the virtual and physical scale, the product or weight should be specified before the product is added to the transaction. Before you add the scale product to the transaction, go to the scale in the peripheral simulator, and use the plus sign (+) and minus sign (–) buttons to adjust the weight that the scale should report. You can also enter the desired weight directly in the **Current value** field. You can adjust the units of weight for the scale by using the plus sign (+), **Edit**, and **Delete** buttons. In this way, units can be specified based on the products that are weighed or the locale where the scale is used.

#### **Configure a scale product**

1. Go to **Retail and Commerce > Products and categories > Released products by category**.
2. Open the product record.
3. Select the product to weigh.
4. On the **Retail and Commerce** FastTab, set the **Scale product** option from **No** to **Yes**.

#### **Synchronize changes to the channel database**

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Select the **1040** distribution schedule.
3. Click **Run now** to synchronize changes to the POS.

After the data is synchronized, when a scale product is added to the POS transaction, the POS checks the scale for the weight.

#### **Signature capture**

The virtual signature capture device prompts the user to provide a signature on the virtual signature capture pad when the tender that is used requires a signature. The user can accept the signature to show it at the POS. The cashier can then accept the signature. The signature is then saved together with the tender and is synchronized to the back office together with other transaction data.

#### **Set up a tender to require a signature**

1. Go to **Retail and Commerce > Channels > Stores > All stores**.
2. Select the store.
3. Click **Edit**.
4. Click **Set up**, and then, in the **Set up** section, click **Payment methods**.

5. Click **Edit**.
6. Select the payment method that requires a signature.
7. In the **General** section, under **Signature Capture**, set the **Use signature capture device** option to **Yes**.
8. In the **Signature capture minimum amount** field, enter the minimum amount that should trigger signature capture.

Synchronize changes to the channel database

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Select the **1070** distribution schedule.
3. Click **Run now** to synchronize changes to the POS.

After the data is synchronized, when a tender is used that requires a signature, and the amount meets the signature threshold, the POS prompts for a signature on the virtual signature capture device.

#### **Additional configuration**

You can edit the peripheral simulator's configuration file to more appropriately address the scenarios that you're testing. You can find the configuration file at C:\Program Files (x86)\Microsoft Dynamics 365\70\VirtualPeripherals\Microsoft.Dynamics.Commerce.VirtualPeripherals.Client.exe.config. The configuration file defines the units that are available for testing on the scale, the card types that are available for testing, and bar code types. For example, by modifying the text values in the configuration file, you can add a new card type or a new unit of measure that can be selected at runtime. The new values will appear after the app is restarted.

#### **Troubleshooting**

Activities for the peripheral simulator are logged in the peripheral simulator. You can find the log at C:\Program Files (x86)\Microsoft Dynamics 365\70\VirtualPeripherals\Microsoft.Dynamics.Commerce.VirtualPeripherals.Client.exe.config. The peripheral simulator also reports issues to the Windows event log, which you can access at **Application and Services Logs > Microsoft > DynamicsAX**.

If changes that you made to the hardware profile or other areas aren't evident when you use MPOS or the peripheral simulator, check the distribution scheduler jobs that you used to synchronize the data to the channel database. If the changes were synchronized but still aren't evident at the POS, restart the POS client.

Changes to configured cash drawers aren't effective until a new shift is created. Therefore, if you make changes to cash drawers, make sure that you always close the existing shift to test the new cash drawer setup.

Sometimes, if a manufacturer's driver is installed after the common control objects from Monroe Consulting Services, the driver can cause the common control objects to stop working correctly. In this case, you should reinstall the common control objects.

At install time, it's possible that certain assemblies related to the virtual peripheral simulator were registered incorrectly. This issue is often associated with an 'OPOS\_E\_CLOSED' error when attempting to use a virtual device. This can be corrected by running the Windows Assembly Registration tool. To register the assembly (called Microsoft.Dynamics.Commerce.VirtualPeripherals.ServiceObjects.dll), open a command prompt as administrator and run 'regasm /codebase "C:\Program Files (x86)\Microsoft Dynamics 365\70\Peripheral simulator for Retail\Microsoft.Dynamics.Commerce.VirtualPeripherals.ServiceObjects.dll"'.

## **POS simulator**

The POS simulator lets device manufacturers, independent software vendors (ISVs), and retailers test peripheral devices without having to deploy the POS. By using the same business logic for peripherals as MPOS and the standalone hardware station, the POS simulator can determine device driver compatibility with the POS as a standalone utility. Therefore, device selection can occur independently of POS setup and deployment.

The POS simulator is also provided as a standalone utility that is independent of Commerce. As a standalone utility, the POS simulator is primarily used to test peripheral device compatibility. Only devices that are tested by

using the POS simulator are acceptable for new deployments of the POS. Testing should be driven by the device manufacturers themselves. Parts of the POS simulator overview are intended to explain how the POS simulator is used for compatibility testing. Manufacturers that are interested in testing their devices for compatibility with the POS or standalone hardware station should send an email to [drpc@microsoft.com](mailto:drpc@microsoft.com) to request information about the program.

### Using the POS simulator

1. Click **Start** on your computer, type **Peripheral simulator**, and then select the app when it appears in the search results.
2. Click **Use virtual peripherals**. The supported devices will be listed as tabs on the left side of the window. To view a specific device, click the tab for that device.

The POS simulator supports the following devices:

- Line display
- Cash drawer
- MSR
- PIN pad
- Printer
- Scale
- Signature capture pad
- Bar code scanner
- Payment terminal

#### NOTE

A payment terminal requires that a payment connector be present. For more information, see [Create an end-to-end payment integration for a payment terminal](#).

Below the list of supported devices, there is a **Settings** tab. You can use the **Settings** tab to specify how the POS simulator should communicate with the devices that are being tested. If **Runtime** is selected, the method that the POS simulator uses to communicate with the device resembles the method that MPOS that has a built-in hardware station communicates. If **Win32** is selected, the POS simulator communicates directly with the device. This communication method resembles the method that a standalone hardware station communicates.

On the **Settings** tab, you can also provide details about the user who is performing the tests. These details are important for manufacturers that perform compatibility testing.

### Configuring the POS simulator

For each supported device class, the POS simulator lets you set up multiple devices. For example, several printers can be configured in the POS simulator. Then, on the **Printer** tab, the user can cycle through the configured devices and test them as required.

The setup parameters that are available for a device depend on the type of device. To set up a new device, select the class of device to test, and then click the plus sign (+) button. A menu of the available device parameters appears.

To edit a device that has already been created, use the left arrow (<) and right arrow (>) buttons to find the appropriate device, and then click **Edit**.

### Configure an OPOS device

1. Select **OPOS** as the device type.
2. Select the name of the device driver. This value is required. This list includes all the OPOS service objects that are installed on the local machine where devices are being tested. You can also manually enter the device driver name.

**IMPORTANT**

Before you can test OPOS devices by using the POS simulator, OPOS common control objects must be installed. These objects are separate from the service objects that the manufacturer provides.

3. Enter data in the rest of the required fields.

**NOTE**

When you configure devices for testing, all fields are required, so that the POS simulator can confirm compatibility with the POS.

4. When the device is configured to the level that is required for either casual testing or compatibility testing, click **Save device**.

**Configure a network device**

The POS simulator can be used to test network devices. The following network devices are supported out of the box:

- **Cash drawer:** APG Atwood
- **Receipt printer:** Star TSP650II
- **Payment terminal:** Although a payment terminal can be configured as network devices, any testing of a payment terminal requires a custom payment connector.

**NOTE**

No payment terminals are supported out of the box.

1. Select **Network** as the device type.
2. Enter data for the rest of the fields.

**NOTE**

The fields for the device driver name, model, and version can help identify the version of the device-specific implementation that is being tested. Because devices tend to have their own communication protocol over IP, custom implementations should be labeled with specific attributes.

3. When the device is configured to the level that is required for either casual testing or compatibility testing, click **Save device**.

**Windows devices**

The POS simulator can also be used to test Windows printers. The setup parameters are the same as the parameters for OPOS devices.

1. Select **Printer** as the device type.
2. Enter data in the rest of the required fields.



#### NOTE

When you configure devices for testing, all fields are required, so that the POS simulator can confirm compatibility with the POS.

3. When the device is configured to the level that is required for either casual testing or compatibility testing, click **Save device**.

### Testing devices

Each device class has unique testing capabilities. In addition to device-specific tests, each device has a **Self-test** function.

When you're testing a device for compatibility, set up the device, and then click the green arrow to begin the self-test. For each device, a different set of tests is performed to determine the device's compatibility with the POS (in **Runtime** mode) or the standalone hardware station (in **Win32** mode). For some devices, the test requires user interaction. For example, the self-test for a bar code scanner requires that a bar code be scanned. Therefore, the self-test will instruct the user to scan a bar code.

Results from each self-test and each manual operation are shown in the **Log** section of the device test page. You can clear the **Log** section, or you can export the results and save them to a file. For information about how the exported log results can be used for official compatibility testing, see the "Instructions for device manufacturers" section later in this topic.

To stop a self-test, click the red square. For example, you might have to stop a self-test if the device becomes non-responsive. The red square can be used only when a self-test is in progress.

### Device-specific settings to be aware of

This section lists settings that you might require some help to complete.

#### Line display > Settings tab

To view the values that are entered in a free text field, click **Lock and claim** to claim the device.

1. Click **Display text**, and view the text that was sent to the device.
2. Click **Clear text** to clear the text from the line display.
3. Click **Release** to release the line display, so that it can be used for other processes.

#### Line display > Advanced tab settings

- **Binary conversion** – Some line displays require that text be converted into binary format. To determine whether this conversion is required, see the device's documentation.
- **Character set** – The code page for the characters that are sent to the device. For the identifiers of specific code pages, see [https://msdn.microsoft.com/library/windows/desktop/dd317756\(v=vs.85\).aspx](https://msdn.microsoft.com/library/windows/desktop/dd317756(v=vs.85).aspx).

#### MSR

- **Open and claim MSR** – Prepare the POS simulator to receive data from the MSR.
- **Release and close MSR** – Close the MSR device when testing is completed.
- **Card info** – Data from the card that was scanned on the MSR device.

#### IMPORTANT

Actual credit cards should never be used for device testing. Even expired credit cards should not be used for testing.

### PIN pad

#### Settings that appear on both tabs

- **Lock** – Lock the PIN pad device so that it can be used only with the POS simulator.
- **Get entry** – Enable the POS simulator to receive PIN data.
- **Cancel operation** – Cancel the request that was sent to the PIN pad.
- **Release** – Release the PIN pad device.

#### Settings tab

- **Amount** – The amount to send to the PIN pad device for customer acceptance.
- **Account number** – Specify the account number, if it's required.
- **Encrypted PIN** – The encrypted PIN that is received from the device.
- **Additional security data** – Specify the cryptography that is used for the encrypted PIN.

#### Advanced tab

- **Timeout** – Specify the number of seconds to wait for a response from the device.
- **Exclusive** – Require that the PIN pad device be claimed before it can be enabled.
- **Override** – Override previous commands that were sent to device when the device isn't responding.

#### Scale

- **Timeout** – The time-out interval. Product should be put on the scale before the weight is read. If the scale doesn't respond within the specified interval, the request is canceled.

#### Signature capture

##### Settings tab

- **Form name** – Some signature capture devices require a form name when the signature request is sent.
- **Signature (in HEX)** – The hex value for the signature data that is received from the device.
- **Rendered signature** – The image of the signature that is received from the device.

##### Advanced tab

- **Timeout** – The time-out interval. If the signature capture device doesn't respond within the specified interval, the request is canceled.
- **Exclusive** – Require that the PIN pad device be claimed before it can be enabled.
- **Override** – Override previous commands that were sent to device when the device isn't responding.
- **Lock** – Lock the device and claim it for use by the POS simulator.
- **Get entry** – Request the signature from the device.
- **Cancel operation** – Cancel the signature request.
- **Release** – Release the device so that it can be used by other processes.

#### Bar code scanner

- **Open and claim scanner** – Open and claim the scanner. The POS simulator can receive scan events after this operation is completed successfully.
- **Release and close the scanner** – Make the scanner available for other processes.
- **Scanned information** – The data that was received from the bar code scanner.

#### Payment terminal

##### Settings that appear on every tab

- **Select an operation** – Select a specific operation to perform on the device. Options include **All**, **Pay by card**, **Refund by card**, and **Void payment**.
- **Lock and claim** – Prepare the device so that it can be used by the POS simulator.
- **Update line items** – Send specified line item details to the device.
- **Authorize payment** – Instruct the payment connector to request payment authorization.

- **Capture payment** – Instruct the payment connector to capture the previous authorization.
- **Release** – Release the device.

### Settings tab

The **Settings** tab contains properties that are sent to the device when a payment authorization is requested.

- **Invoice number** – Specify the invoice number that is generated at the POS.
- **Test mode** – A value that indicates that a test transaction is being performed.
- **Payment connector** – The name of the payment connector to use.
- **Debit cashback limit** – The maximum amount of cash back that can be requested on the device when a debit transaction is performed.
- **Locale** – The locale that is used. Locales are defined in the configuration file for the POS simulator.
- **Maximum amount allowed** – The maximum amount that can be processed for a given transaction.
- **Minimum amount allowed** – The minimum amount that can be processed for a given transaction.
- **Signature capture minimum amount** – The lowest amount that the customer is prompted to provide a signature for.
- **Terminal ID** – The terminal ID that is included in the transaction properties that the processor provided.

### Lines tab

The **Lines** tab contains data that can be sent to the device when it's used as a line display.

- **Description** – The product description for the transaction line to show on the device.
- **Discount** – The discount amount that is applied to the transaction line.
- **Extended line with tax** – The product line amount. This amount includes tax.
- **Line item ID** – The product ID for the transaction line.
- **Quantity** – The quantity for the transaction line.
- **Stock keeping unit** – The unit of measure to show on the device.
- **Unit price** – The selling price for the transaction line.
- **Universal product code** – The bar code that was scanned for the transaction line.
- **Is voided** – A value that indicates that the transaction line has been voided from the transaction.

### Advanced tab

The **Advanced** tab contains subtotal information that can be shown on the device.

- **Discount amount** – The total amount for discounts on the transaction.
- **Subtotal amount** – The subtotal for the transaction. This amount doesn't include tax.
- **Tax amount** – The tax amount for the transaction.
- **Total amount** – The total amount that is due. This amount includes tax and discounts.
- **Currency** – The currency that is used for the transaction.

### Payment info tab

The **Payment info** tab contains data that is received from the payment connector after an authorization has been processed.

- **Is approved** – A value that indicates whether the authorization request is approved.
- **Card number masked** – The masked card number that is provided by the device. Typically, only the last four digits of the card number are shown.
- **Card type** – The issue for the card that is used in the transaction.
- **Approved amount** – The amount that is authorized for the card payment.
- **Payment SDK data** – Additional data that the payment software development kit (SDK) can provide, such as

the authorization code and other data that the payment processor provides.

- **Signature data** – Hexadecimal signature data that is provided by the device.
- **Payment errors** – Any errors that occurred during the authorization request.

## Instructions for device manufacturers

To submit a device so that it can be listed for compatibility with Commerce, follow these steps.

1. On the **Settings** tab, specify all contact information that is requested. All fields are required. The personal contact information is for Microsoft internal use. It will be used only if there are issues with devices that have been submitted as compatible.
2. For each device, on the **Devices** tab, set up the device, and provide values for all fields. For each permutation, a new device must be specified. For example, if a given device mode is tested for USB and serial, two compatibility reports must be submitted. Any variance from one device to the next should be indicated. Then, when devices are listed for compatibility, very specific information can be provided to people who implement the devices. The more information that can be provided about device compatibility, the higher the success rate for implementations and the fewer support issues will be raised.
3. After you've entered the required information for devices and settings, test the device.
  - a. Clear the log for the device that you're testing.
  - b. Click **Self-test**.
  - c. When a test has been completed successfully, export the test results, and save them. Give the file a name that will make the test easy to identify. We suggest that you use an abbreviated form of the manufacturer's name and the device model as the file name.

The following example shows the data that is included in the exported file.

```

<?xml version="1.0" encoding="utf-8"?>
<Report xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.datacontract.org/2004/07/Microsoft.Dynamics.Commerce.VirtualPeripherals.Modules">
  <CreationDateTime>05/27/2017 19:34:50</CreationDateTime>
  <ManufacturerInfo>
    <ManufacturerName>Contoso </ManufacturerName>
    <ManufacturerWebsite>http://www.contoso.com</ManufacturerWebsite>
    <SupportEmail>support@contoso.com</SupportEmail>
    <SupportTelephoneNumber>555-555-5555</SupportTelephoneNumber>
    <SupportWebsite>http://www.contoso.com</SupportWebsite>
    <TechnicalContactEmail>karen@contoso.com</TechnicalContactEmail>
    <TechnicalContactName>Karen Berg</TechnicalContactName>
    <TechnicalContactPhone>555-555-5555</TechnicalContactPhone>
  </ManufacturerInfo>
  <DeviceInfo i:type="OposDevice">
    <DeviceDriverName>MockOPOSDrawer1</DeviceDriverName>
    <DeviceModel>Model1</DeviceModel>
    <DriverVersion>V2</DriverVersion>
    <FirmwareVersion>V1</FirmwareVersion>
    <HardwareType>OPOS</HardwareType>
    <ConnectionType>USB</ConnectionType>
    <DriverDownloadLink>http://model1.drivers.contoso.com</DriverDownloadLink>
  </DeviceInfo i:type="OposDevice">
  <LogItems>
    <LogMessage>
      <LogType>Info</LogType>
      <Message>The cash drawer is opened successfully.</Message>
      <Timestamp>05/27/2017 19:48:14</Timestamp>
    </LogMessage>
    <LogMessage>
      <LogType>Info</LogType>
      <Message>The cash drawer open operation elapsed time: 00:00:00.053</Message>
      <Timestamp>05/27/2017 19:48:14</Timestamp>
    </LogMessage>
    <LogMessage>
      <LogType>Info</LogType>
      <Message>The cash drawer status check operation is completed successfully.</Message>
      <Timestamp>05/27/2017 19:48:14</Timestamp>
    </LogMessage>
    <LogMessage>
      <LogType>Info</LogType>
      <Message>The cash drawer status check operation elapsed time: 00:00:00.045</Message>
      <Timestamp>05/27/2017 19:48:14</Timestamp>
    </LogMessage>
    <LogMessage>
      <LogType>Info</LogType>
      <Message>Test finished successfully.</Message>
      <Timestamp>05/27/2017 19:48:14</Timestamp>
    </LogMessage>
  </LogItems>
</Report>

```

Eventually, the data that is included in the report (except personal contact information) will be listed on a device compatibility website. It will also be listed in design and deployment tools that are used to manage customer environments.

Successful logs should be sent to [drpc@microsoft.com](mailto:drpc@microsoft.com). Include the manufacturer's name and the device model in the subject line.

For support if you're performing compatibility tests, and for other inquiries, send an email to [drpc@microsoft.com](mailto:drpc@microsoft.com).

## Additional resources

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Connect peripherals to the point of sale (POS)

2/18/2021 • 11 minutes to read • [Edit Online](#)

This topic covers how to connect peripherals to your Retail POS.

## NOTE

For specific installation instructions, see [Configure and install Retail hardware station](#) and [Configure, install, and activate Modern POS \(MPOS\)](#).

## Key components

Several components are used to define the relationships among a store, the point-of-sale (POS) registers or channels within the store, and the peripherals that those registers or channels use to process transactions. This section describes each component and explains how it should be used in a store deployment.

### POS registers

Navigation: Click **Retail and Commerce** > **Channel setup** > **POS setup** > **Registers**.

The POS register is an entity that is used to define the characteristics of a specific instance of the POS. These characteristics include the hardware profile or setup for peripherals that will be used at the register, the store that the register is mapped to, and the visual experience for the user who logs on to that register.

### Devices

Navigation: Click **Retail and Commerce** > **Channel setup** > **POS setup** > **Devices**.

A device is an entity that represents a physical instance of a device that is mapped to a POS register. When a device is created, it's mapped to a POS register. The device entity tracks information about when a POS register is activated, the type of client that is being used, and the application package that has been deployed to a specific device. Devices can be of two types: **Retail modern POS (MPOS)** or **Retail Cloud POS (Cloud POS)**.

### MPOS

MPOS is a POS client application that is installed on Windows 8.1 or a later PC-based operating system. If the **Retail modern POS** application type is mapped to a device, the download package can be specified for a particular device. The download package can be customized to include different versions of the installation package. The ability to deploy different packages provides flexibility in cases where different POS registers might need different integrations. MPOS is deployed together with a built-in hardware station.

### Cloud POS

Cloud POS is a browser-based POS. Because it runs in the browser, Cloud POS doesn't require Windows 8.1 or a later PC-based operating system. If the **Retail Cloud POS** application type is mapped to a specific device in Headquarters, that device can be used through the browser with no need to download or install a package. Cloud POS requires a hardware station to use hardware beyond keyboard wedge based bar code scanning.

### Hardware profile

Navigation: Click **Commerce** > **Channel setup** > **POS setup** > **POS profiles** > **Hardware profiles**.

A hardware profile identifies the hardware that is connected to a POS register or a hardware station. The hardware profile is also used to specify the payment processor parameters that should be used during communication with the payment software development kit (SDK). (The payment SDK is deployed as part of the hardware station.)

## Hardware station

Navigation: Click **Retail and Commerce > Channels > Stores > All stores**. Select a store, and then click the **Hardware stations** FastTab.

A hardware station is an instance of business logic that drives POS peripherals. A hardware station is automatically installed together with MPOS. Alternatively, the hardware station can be installed as a stand-alone component, and then accessed by MPOS or Cloud POS through a web service. The hardware station must be defined at the channel level.

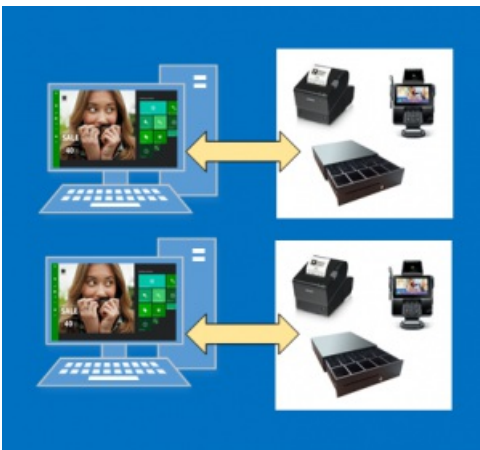
## Hardware station profile

Navigation: Click **Commerce > Channel setup > POS setup > POS profiles > Hardware station profiles**.

Whereas the hardware station itself is specified at the channel level includes instance-specific information, such as the URL for the hardware station, the hardware station profile includes information that can be static or shared across multiple hardware stations. The static information includes the port that should be used, the hardware station package, and the hardware profile. The static information also includes a description of the type of hardware station that is being deployed, such as **Checkout** or **Returns**, depending on the hardware that is required for each specific hardware station.

## Scenarios

### MPOS with connected peripheral devices



To connect MPOS to POS peripherals in a traditional, fixed POS scenario, first navigate to the register itself, and assign a hardware profile to it. You can find the POS registers at **Retail and Commerce > Channel setup > POS setup > Registers**.

After you've assigned the hardware profile, sync changes to the channel database by using the **Registers** distribution schedule. You can find the distribution schedules at **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.

Next, set up a "local" hardware station on the channel. Click **Retail and Commerce > Channels > Stores > All stores**, and select a store.

Then, on the **Hardware stations** FastTab, click **Add** to add a hardware station. Enter a description, enter **localhost** as the host name, and then sync the changes to the channel by using the **Channel configuration** distribution schedule. You can find the distribution schedules at **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.

Finally, in MPOS, use the **Select hardware station** operation to select the **localhost** hardware station. Set the hardware station to **Active**. The hardware profile that is used in this scenario should come from the POS register itself. A hardware station profile isn't required for this scenario.

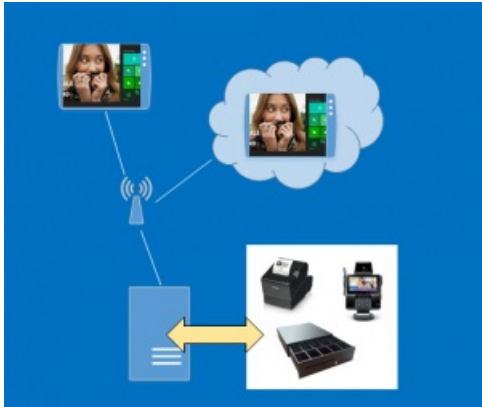


## NOTE

Some hardware profile changes, such as changes to cash drawers, require that a new shift be opened after the changes have been synced to the channel.

Cloud POS must use the stand-alone hardware station to communicate with peripherals.

### MPOS or Cloud POS with a stand-alone hardware station



In this scenario, a stand-alone hardware station is shared among MPOS and Cloud POS clients. This scenario requires that you create a hardware station profile to specify the download package, port, and hardware profile that the hardware station uses. You can find the hardware station profile at **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware station profiles**.

After you've created the hardware station profile, navigate to the specific channel (**Retail and Commerce > Channels > Stores > All stores**), and add a new hardware station. Map this new hardware station to the hardware station profile that was previously created.

Next, provide a description that will help the cashier identify the hardware station. In the **Host name** field, enter the host machine URL in the following format: `https://<MachineName:Port>/HardwareStation`. (Replace **<MachineName:Port>** with the actual machine name of the hardware station and the port that is specified in the hardware station profile.) For a stand-alone hardware station, you should also specify the electronic funds transfer (EFT) terminal ID. This value identifies the EFT terminal that is connected to the hardware station when the payment connector communicates with the payment provider.

Next, from the actual hardware station machine, navigate to the channel, and select the hardware station. Then click **Download**, and install the hardware station.

Next, from MPOS or Cloud POS, use the **Select hardware station** operation to select the hardware station that was previously installed. Select **Pair** to establish a secure relationship between the POS and the hardware station. This step must be completed once for every combination of a POS and a hardware station.

After the hardware station is paired, the same operation is used to make the hardware station active while it's used. For this scenario, the hardware profile should be assigned to the hardware station profile rather than the register itself. If for some reason a hardware station does not have a hardware profile directly assigned, then the hardware profile assigned to the register is used.

## Client maintenance

### Registers

POS registers are managed primarily through the registers themselves, and also through the profiles that are assigned to registers. Attributes that are specific to an individual register are managed at the register level. These attributes include the store where the register is used, the register number, the description, and the EFT terminal ID that is specific to the register itself.

## POS profiles

You can find the POS profiles at **Retail and Commerce > Channel setup > POS setup > POS profiles**. It's useful to manage many aspects of a register through profiles, because the profiles can be shared among many registers. Profiles can be mapped either to an individual register or, if a profile is effective on a store-wide basis, to the store. The following sections describe the POS profiles and how they are used.

### Offline profile

The offline profile is set at the store level. It's used to specify the upload settings for transactions that are performed on a POS register while that register isn't connected to the channel database.

### Functionality profile

The functionality profile is set at the store level. It's used to specify store-wide settings about the functions that can be performed at the POS. The following capabilities are managed through the functionality profile. These capabilities are arranged by FastTab.

- **General** FastTab:
  - International Organization for Standardization (ISO).
  - Create a customer in offline mode.
  - Email receipt profile.
  - Central staff logon authentication.
- **Functions** FastTab:
  - Management of logon and extended logon.
  - Financial and currency-related aspects of the POS, such as the ability to key in prices and whether decimals are required for minor currency.
  - Enabling time registration through the POS.
  - How products and payments appear in the POS and on receipts.
  - End-of-day management.
  - Channel database transaction retention parameters.
  - How customers are looked up and created from the POS.
  - How discounts are calculated.
- **Amount** FastTab:
  - Maximum and minimum prices that are allowed.
  - Discount application and calculation.
- **Info codes** FastTab:
  - All aspects of how info codes are managed at the POS. For details, see [Info codes and info code groups](#).
- **Receipt numbering** FastTab:
  - Specify receipt numbering masks, which might include segments for the store number, terminal number, constants, and whether sales, returns, sales orders, and quotations are printed in separate sequences, or whether they all following the same sequence.

### Receipt profiles

Receipts profiles are assigned to printers within the hardware profile. They are used to specify the receipt types that are printed at a specific printer. The profiles include settings for the receipt formats, and settings that determine whether the receipt is always printed, or whether the cashier is prompted to decide whether the receipt must be printed. Different printers might also use different receipt profiles. For example, printer 1 is a standard thermal receipt printer, and therefore has smaller receipt formats. However, printer 2 is a full-size receipt printer that is used to print only customer order receipts, which require more space.

### Hardware profiles

Hardware profiles are explained as a component for client setup earlier in this article. Hardware profiles are assigned directly to the POS register or to a hardware station profile. They are used to specify the types of devices a specific POS register or hardware station uses. Hardware profiles are also used to specify the EFT settings that are used to communicate with the payment SDK.

#### **Visual profiles**

Visual profiles are assigned at the register level. They are used to specify the theme for a specific register. The profiles include settings for the type of application that is used (MPOS or Cloud POS), the accent color and theme, the font scheme, the logon background, and the POS background.

#### **Custom fields**

You can create custom fields to add fields that aren't provided out of the box to the POS. For more information about how to use custom fields, see the [Working with custom fields blog post](#).

#### **Language text**

You can override default strings in the POS by using language text entries. To override a string in the POS, add a new language text line. Then specify an ID, the default string that should be overridden, and the text that should be shown at the POS instead of the default string.

#### **Hardware station profiles**

Hardware station profiles are explained earlier in this article. They are used to assign non-instance-specific information to hardware stations.

#### **Channel reports configuration**

You set up the reports that are available at the channel on the **Channel reports configuration** page. You can create new reports by providing the XML definition of the report and assigning the report to a specific permission group at the POS.

#### **Devices**

Devices are explained earlier in this article. They are used to manage the activation of a specific POS register. Devices are also used to specify the application that is used for a specific register and the installation package that should be used to install the MPOS client. Here are the device activation states:

- **Pending** – The device is ready to be activated.
- **Activated** – The device has been activated.
- **Deactivated** – The device has been deactivated either in Headquarters or through the POS.
- **Disabled** – The device has been disabled.

Additional activation-related information includes the worker who changed the activation status for the device, a time stamp for the activation, and whether the device configuration has been validated.

#### **Client data synchronization**

All changes to a POS client, except changes in the device activation status, must be synced to the channel database to take effect. To sync changes to the channel database, navigate to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**, and run the required distribution schedule. For client changes, you should run the **Registers** and **Channel configuration** distribution schedules.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Health check for POS peripherals and services

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes the health check operation in the point of sale (POS).

## Overview

Retail stores can be complex environments where many applications and devices are used. As operations grow, it can become difficult to ensure that operations always run smoothly, because of dependencies on, for example, peripherals that can break or accidentally become unplugged over the course of a day. Troubleshooting for issues that are related to devices and services can be costly for larger merchants and equally frustrating for smaller operations.

Microsoft Dynamics 365 Commerce versions 10.0.10 and later include a health check operation that can help prevent some of this cost and frustration. This operation provides a method for testing devices directly from the POS outside of normal operations. Therefore, it can help retailers detect issues before they occur.

## Key terms

TERM	DESCRIPTION
Peripheral	Any device that the POS application uses to facilitate transactions and other operations in the store. Examples include cash drawers, bar code scanners, and payment terminals.
Service	In this topic, a service is an ancillary application that the POS application depends on to perform transactions and daily operations. Examples include apps that help with tax or shipping calculations.

## Health check operation

The health check operation is operation 717 on the **POS Operations** page in Commerce Headquarters. It can be used while the POS is in non-drawer mode. However, a hardware station must be active.

By default, the health check tests only devices that are configured in the hardware profile for the hardware station that is currently active for a register. If a register uses multiple hardware stations over the course of a day, to do health checks for all of them, it must connect to one hardware station at a time. There is no store-level health check. However, it's possible that this type of check can be done through Commerce Server extensibility.

### Out-of-box health checks

TYPE	CONNECTION	DETAILS
------	------------	---------

TYPE	CONNECTION	DETAILS
Printer	OPOS	<p>This check tests basic object linking and embedding for POS (OPOS) functions. Here are some examples:</p> <ul style="list-style-type: none"> <li>• Open: <b>Open &gt; ClaimDevice &gt; DeviceEnabled=True</b></li> <li>• Close: <b>DeviceEnabled=False &gt; ReleaseDevice &gt; Close</b></li> </ul>
Line display	OPOS	<p>This check tests basic OPOS functions. Here are some examples:</p> <ul style="list-style-type: none"> <li>• Open: <b>Open &gt; ClaimDevice &gt; DeviceEnabled=True</b></li> <li>• Close: <b>DeviceEnabled=False &gt; ReleaseDevice &gt; Close</b></li> </ul>
Dual display	Windows	<p>This check ensures that the operating system detects a second Windows display.</p>
MSR	OPOS	<p>This check tests basic OPOS functions. Here are some examples:</p> <ul style="list-style-type: none"> <li>• Open: <b>Open &gt; ClaimDevice &gt; DeviceEnabled=True</b></li> <li>• Close: <b>DeviceEnabled=False &gt; ReleaseDevice &gt; Close</b></li> </ul>
Drawer	OPOS	<p>This check tests basic OPOS functions. Here are some examples:</p> <ul style="list-style-type: none"> <li>• Open: <b>Open &gt; ClaimDevice &gt; DeviceEnabled=True</b></li> <li>• Close: <b>DeviceEnabled=False &gt; ReleaseDevice &gt; Close</b></li> </ul>
Scanner	OPOS	<p>This check tests basic OPOS functions. Here are some examples:</p> <ul style="list-style-type: none"> <li>• Open: <b>Open &gt; ClaimDevice &gt; DeviceEnabled=True</b></li> <li>• Close: <b>DeviceEnabled=False &gt; ReleaseDevice &gt; Close</b></li> </ul>
Scale	OPOS	<p>This check tests basic OPOS functions. Here are some examples:</p> <ul style="list-style-type: none"> <li>• Open: <b>Open &gt; ClaimDevice &gt; DeviceEnabled=True</b></li> <li>• Close: <b>DeviceEnabled=False &gt; ReleaseDevice &gt; Close</b></li> </ul>

TYPE	CONNECTION	DETAILS
PIN pad	OPOS	<p>This check tests basic OPOS functions. Here are some examples:</p> <ul style="list-style-type: none"> <li>• Open: <code>Open &gt; ClaimDevice &gt; DeviceEnabled=True</code></li> <li>• Close: <code>DeviceEnabled=False &gt; ReleaseDevice &gt; Close</code></li> </ul>
Payment terminal	Payments SDK	<p>This check tests basic payment terminal functions provided by the Payments SDK.</p> <ul style="list-style-type: none"> <li>• Lock</li> <li>• BeginTransaction</li> <li>• EndTransaction</li> <li>• ReleaseDevice</li> <li>• Close</li> </ul>

### Using the health check operation in the POS

When the health check operation is initiated in the POS, a pane on the right lists the configured devices and shows the status of each device. To do a health check for a single device, select the device, and then select **Test selected**. To do a health check for all devices, select **Test all**. The **Test all** function tests all the devices, one at a time, and updates the status of each device in the **Status** column.

The **Last check** column shows when the health check was last done for each device.

If the health check for a device passes (that is, if no errors are encountered), the device's status will be **OK**. If the health check fails, the status will indicate that there was an error. In this case, the pane on the right provides details that are related to the error, or it instructs the user to contact the system admin.

Some devices, such as the OPOS keylock, don't have out-of-box health check tests. If a health check test isn't detected for any device that is used, the status will be **Not supported**.

### Extending health checks

The out-of-box health check tests are configured to provide some user-friendly messages for typical errors. However, not all scenarios are covered. Through extensibility, merchants can map user-friendly messages to errors that might be specific to their environment.

Custom health checks can also be created to test devices that aren't supported out of the box, or to test any services that the POS depends on.

## Related articles

[Modern POS \(MPOS\) triggers and printing](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Support for network peripherals

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes the support for and setup of network peripherals in the store.

## Key terms

TERM	DESCRIPTION
Register	The entity that is used to configure an instance of a point of sale (POS) register.
Device	A representation of the physical instance of a POS register and the Modern POS application that is assigned to it.
Dedicated hardware station	The hardware station business logic that is built into the Modern POS for Windows and Modern POS for Android applications.
Drawer kick (d/k) port	A traditional method for connecting a cash drawer to a receipt printer.
Network peripherals	Built-in support for network-enabled payment terminals, receipt printers, and cash drawers.
ESC/P	Epson Standard Code for Printers, also known as "Escape/P," is a printer control language created by Epson that is commonly used to send commands to point of sale printers.

## Supported POS clients and devices

Functionality for network peripherals is supported by the Modern POS for Windows and Modern POS for Android POS clients.

This functionality supports network-enabled payment terminals and receipt printers. You can provide cash drawer support by connecting the cash drawer to the network-enabled receipt printer via the d/k port.

Out-of-box support for this functionality is provided by the [Microsoft Dynamics 365 Payment Connector for Adyen](#). However, other payment connectors might be supported via the Commerce software development kit (SDK). Supported receipt printers include network-enabled receipt printers from Star Micronics and Epson.

Out-of-box support is provided for network protocols for Epson and Star Micronics receipt printers. Cash drawers that are connected to those printers via the d/k port are supported via ESC/P protocols.

## Set up network peripherals

### Adyen payment terminal

For information about how to set up an Adyen payment terminal, see the "POS payment terminal" section in [Dynamics 365 Payment Connector for Adyen](#).

### Epson or Star Micronics receipt printer and a cash drawer

#### Epson prerequisite

If your printer supports Epson ePOS-Print, see Epson's product manuals for information about how to access the Epson Printers Configuration Website and enable ePOS-Print. When ePOS-Print is configured, the printer will print a chit that shows its IP address after a power cycle.

#### Star Micronics prerequisite

Network-enabled Star Micronics printers that support Ethernet can be configured to use network protocols through the Star Micronics Printer Utility. For information about how to set up Star Micronics devices to support network printing, see the documentation that is provided by Star Micronics. When a device is correctly configured, the IP address of the printer can be obtained through the user interface (UI) of the printer utility.

#### Set up a hardware profile

1. In Dynamics 365 Commerce, search for **Hardware profiles**.
2. Select **New**.
3. Assign a hardware profile number, and then enter a description.
4. On the FastTabs for different device types, set up the following device types.

DEVICE	TYPE	DEVICE NAME	ADDITIONAL DETAILS
Printer	Network	Epson or Star	The device name is case-sensitive. Assign a <b>Receipt profile ID</b> .
Cash drawer	Network	Epson or Star	The device name is case-sensitive. Set the <b>Use shared shift</b> option to <b>Yes</b> if the cash drawer will be shared by multiple POS devices.

5. Select **Save**.

#### Set up Modern POS for Windows or Modern POS for Android clients that have built-in hardware station logic

1. In Dynamics 365 Commerce, search for **Registers**.
2. Select a register by selecting the register number, and then select **Edit**.
3. Assign the hardware profile that you just created to the register that should use a dedicated payment terminal. The device that is mapped to this register must use either the Modern POS for Windows application or the Modern POS for Android application.
4. Select **Save**.
5. On the Action Pane, on the **Registers** tab, select **Configure IP addresses**.
6. On the **Printer** FastTab, enter the IP address of the printer. Leave the field for the port number blank.
7. On the **Cash drawer** FastTab, enter the IP address of the printer. Leave the field for the port number blank.
8. Select **Save**.
9. Search for **All stores**.
10. Select a store by selecting its **Retail Channel Id** value, and then select **Edit**.
11. On the **Hardware stations** FastTab, select **Add**.
12. Set the **Hardware station type** field to **Dedicated**.
13. Enter a description, but don't specify a hardware profile or set any other values for this hardware station.
14. Select **Save**.
15. Search for **Distribution schedules**.
16. Select distribution schedule **1090**, and then select **Run now**.
17. Select distribution schedule **1070**, and then select **Run now**.



The Modern POS for iOS and Modern POS for Cloud applications don't have built-in hardware station logic. If you're using either of these applications, follow these steps.

1. In Dynamics 365 Commerce, search for **All stores**.
2. Select a store by selecting its **Retail Channel Id** value, and then select **Edit**.
3. On the **Hardware stations** FastTab, select **Add**.
4. Set the **Hardware station type** field to **Shared**.
5. Enter a description. This hardware station can be shared by multiple POS clients, including POS clients that have built-in hardware station logic.
6. In the **Hardware profile** field, use the drop-down arrow to assign the hardware profile for network peripherals to this hardware station.
7. Select **Save**.
8. While the hardware station for the receipt printer and cash drawer is still selected, select **Configure IP addresses**.
9. On the printer FastTab, enter the IP address of the printer. Leave the field for the port number blank.
10. On the cash drawer FastTab, enter the IP address of the printer. Leave the field for the port number blank.

#### **NOTE**

For detailed information about how to set up shared hardware stations, see [Configure and install Retail hardware station](#).

11. Select **Save**
12. Search for **Distribution schedules**.
13. Select distribution schedule **1090**, and then select **Run now**.
14. Select distribution schedule **1070**, and then select **Run now**.

### **Sharing network peripherals**

Network receipt printers and cash drawers can be shared by multiple POS devices. To share them, you can use a shared hardware station to broker the connection to the devices. Alternatively, if you're using Modern POS for Windows or Modern POS for Android, you can configure the same devices directly in the register properties.

Payment terminals can be shared only if a shared hardware station is deployed to broker the connection to the payment terminal. You can't share a payment terminal by setting the same payment terminal IP address directly at the register level. If you try to use this approach, issues will occur when individual POS clients try to lock and claim the device.

## **Related articles**

- [Set up POS hybrid app on Android and iOS](#)
- [Dynamics 365 Payment Connector for Adyen](#)
- [Dedicated payment terminals and prompts for a printer and cash drawer](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Demo data screen layouts in Modern POS (MPOS) and Cloud POS

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic provides information about the screen layouts that are included with the demo data set for the point of sale (POS) experiences in Dynamics 365 Commerce.

## Overview

The sample screen layouts that are included with Commerce demo data provide content that is optimized for various retail segments, store worker roles, and devices. A single layout can contain several layout sizes and combinations of button grids, to help ensure coverage as store workers move between devices and stations. This topic highlights the differences between these layouts, the operations that they provide, and the overall experiences that they deliver.



## Anatomy of a screen layout ID

To find screen layouts, go to **Retail and Commerce > Channel setup > POS setup > POS > Screen layouts**.

Screen layout ID	Name	Layout type	Width	Height
F4MGR	Fabrikam Manager	Modern POS - Full	1024	768
1280x720 - Full		Modern POS - Full	1280	720
1366x768 - Full		Modern POS - Full	1366	768
1440x960 - Full		Modern POS - Full	1440	960
1536x864 - Full		Modern POS - Full	1536	864
480x853 - Compact		Modern POS - Compact	480	853

Screen layout IDs can have a maximum of 10 characters. The ID is a string that consists of three pieces of

information, in this order:

1. Company
2. Layout version
3. Persona

### Company

LETTER	COMPANY
A	Adventure Works
F	Fabrikam
C	Contoso

### Layout version

VERSION NUMBER	DESCRIPTION
3	The base version that supports multiple screen sizes for various devices and aspect ratios
3.1	The base version that has additional support for the <b>Recommended products</b> panel
4	The extended version for extended Fabrikam updated layout

### Persona

ABBREVIATION	PERSONA	CONTENTS
CSH	Cashier	Cashier layouts include all transaction-related operations, such as customer orders, returns, discounts, voids, and gift cards. These layouts also include daily tasks for inventory management, such price checks, inventory lookups, and stock counts. Basic shift management is also provided for start amounts, suspending shifts, and time clock.
MGR	Store Manager	Store Manager layouts include all transaction-related operations that are found in the Cashier layouts but also include tax overrides. These layouts also include daily tasks for inventory management, such as price checks, inventory lookups, and stock counts. Shift management is provided for starting, suspending, and closing shifts. Additionally, the layouts include drawer operations for entries, removals, tender declarations, and safe and bank drops. Finally, these layouts include access to performance reports, and enable X and Z reports to be printed.

ABBREVIATION	PERSONA	CONTENTS
--------------	---------	----------

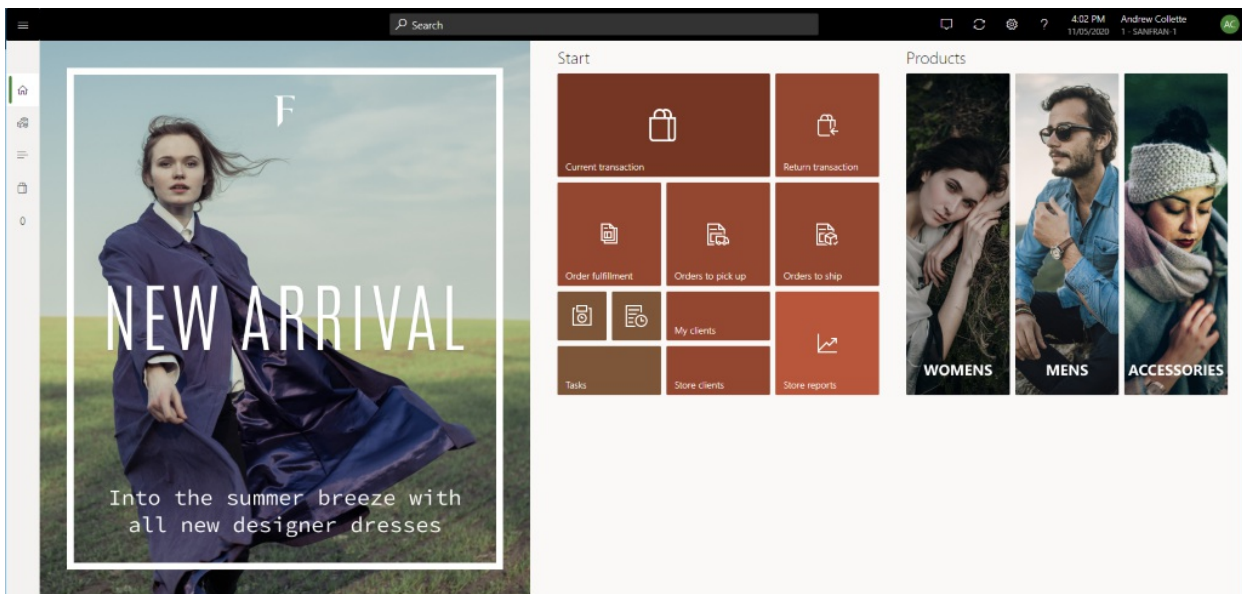
STK	Stock Clerk	<p>Stock Clerk layouts are optimized for inventory management. They include access to daily tasks for price checks, inventory lookups, picking and receiving, stock counts, and kit disassembly. These layouts also provide basic shift operations for time clock and suspending shifts. Although these layouts are intended mainly for back-office tasks, stock clerks have the same operations as cashiers for transaction screens.</p>
-----	-------------	---

### Example layout

Here is an example of a screen layout ID for the Fabrikam company, layout version 4, and the Store Manager persona:

F4MGR

The following illustration shows an example of the Welcome screen for a Fabrikam store manager.



### Layout sizes

## Full vs. compact layouts

A screen layout can have configurations for both full devices and compact devices. Therefore, a user can be assigned to a single screen layout that will work across various sizes and form factors in the store.

- **Modern POS - Full** – Typically, full layouts are best used for larger displays, such as desktop computer monitors or tablets. Users can select the UI elements that the layout includes, specify the size and placement of those elements, and configure their detailed properties. Full layouts support both portrait and landscape configurations.
- **Modern POS - Compact** – Typically, compact layouts are best used for phones or small tablets. Design possibilities are limited for compact devices. Users can configure the columns and fields for the receipt pane and the totals pane.

## Screen resolutions that are provided

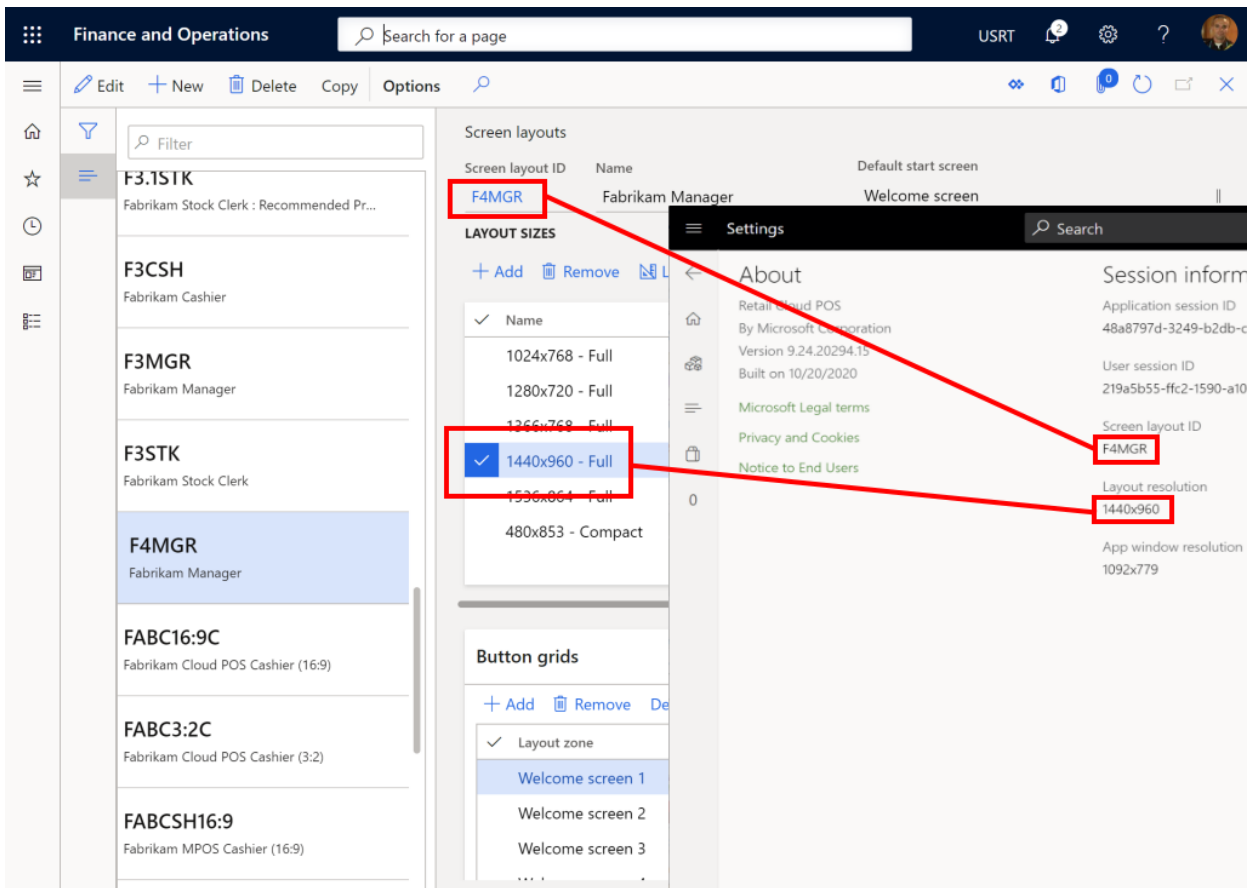
The following table shows the layout sizes that are provided for typical screen resolutions.

LAYOUT TYPE	RESOLUTION	ASPECT RATIO	TARGET DISPLAY
Compact*	480 × 853	16:9	Phones
Full	1024 × 768	4:3	Tablets
Full*	1280 × 720	16:9	Tablets
Full	1366 × 768	16:9	Tablets, larger screens
Full	1440 × 960	3:2	Tablets, larger screens
Full*	1536 × 864	16:9	Tablets, larger screens

\* These additional layout sizes are available only in Adventure Works and Fabrikam layouts.

### TIP

POS automatically selects layout sizes, based on the closest size that is available for the screen resolution of the current app window. To find the screen layout ID and layout resolution that are currently used, in Modern POS (MPOS) or Retail Cloud POS (CPOS), open the **Settings** page, and look in the **Session information** section. You can also see the actual window resolution for your current application or browser frame. After you have this information, you can find the source of the layout content by going to **Channel setup > POS setup > POS > Screen layouts**.



## Companies and brands

Each fictitious company is targeted to a different retail segment and includes product catalogs that are tuned for the company's market. Each company has a unique visual brand that accompanies its products. Branding elements include the accent color, dark or light theme, and accompanying photographs that provide realistic experiences.

### Company segment and visual characteristics

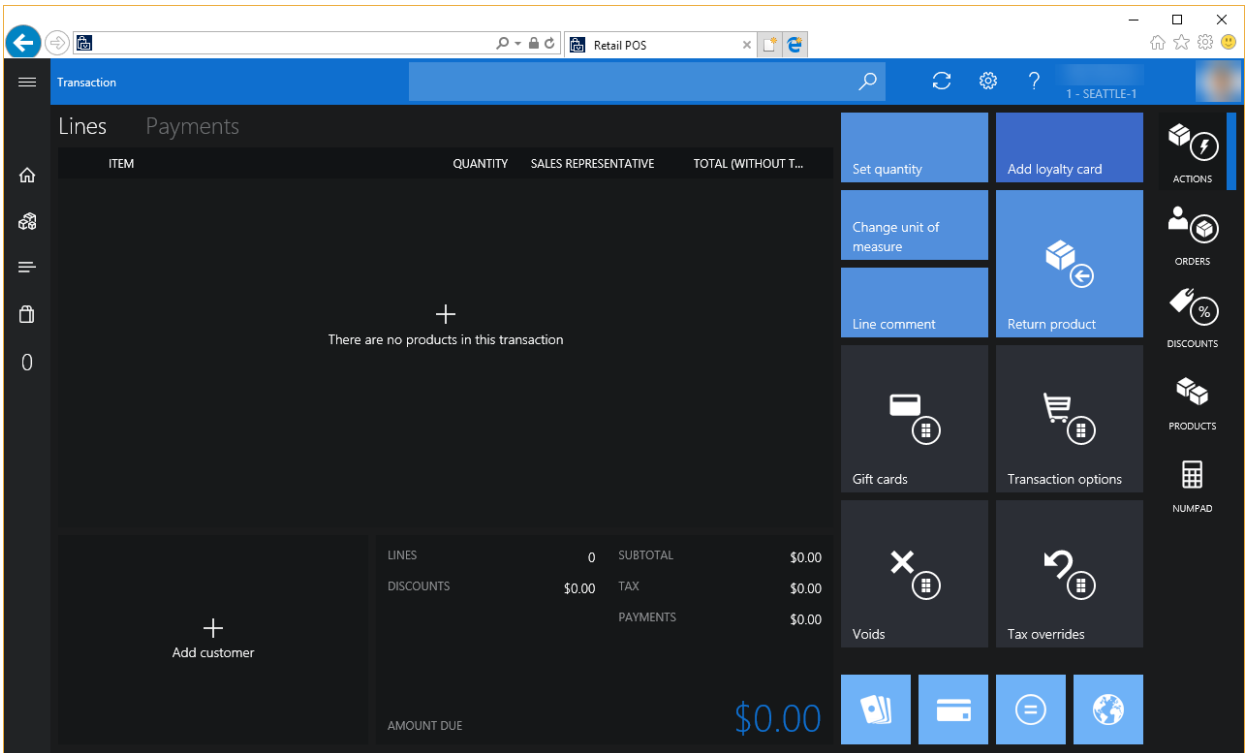
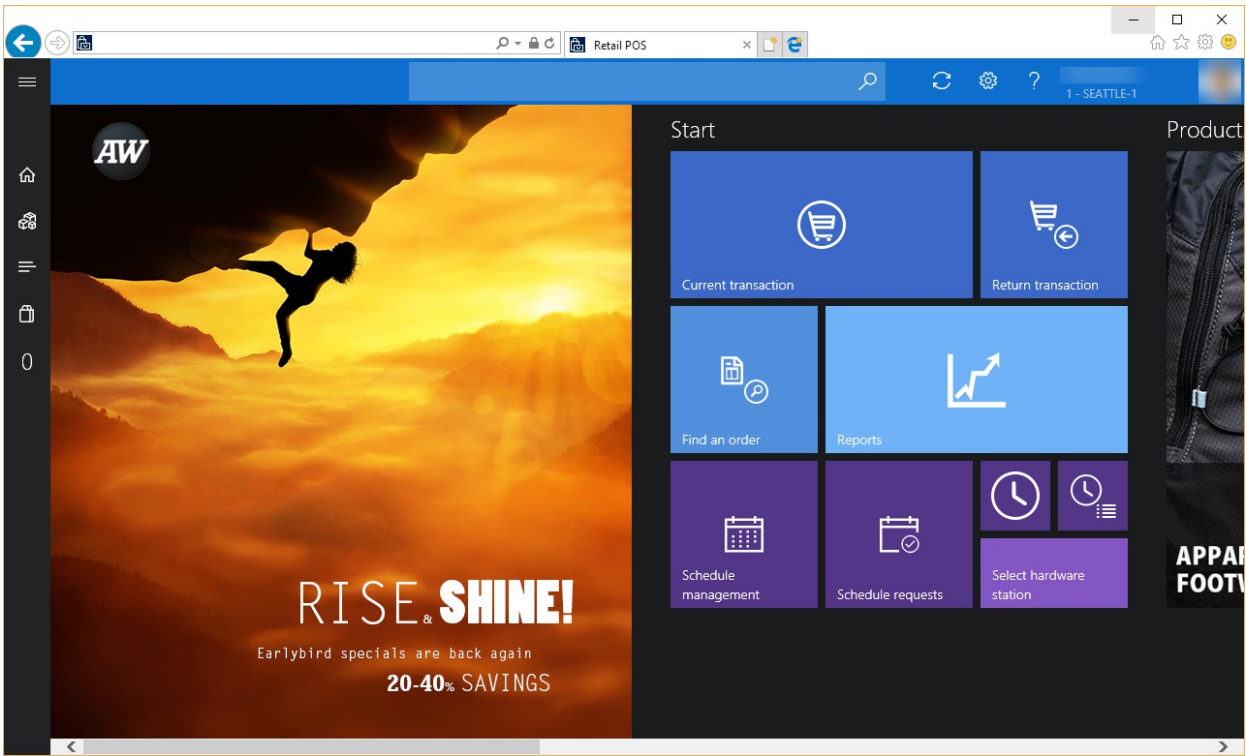
COMPANY	LOCATION	SEGMENT	ACCENT	THEME
Adventure Works	Seattle	Sporting Goods	Blue	Dark
Fabrikam	San Francisco	Fashion	Green	Light
Contoso	Boston	Electronics	Red	Dark

#### NOTE

Adventure Works and Fabrikam are the two flagship brands. Contoso is available, but not all layouts have been provided.

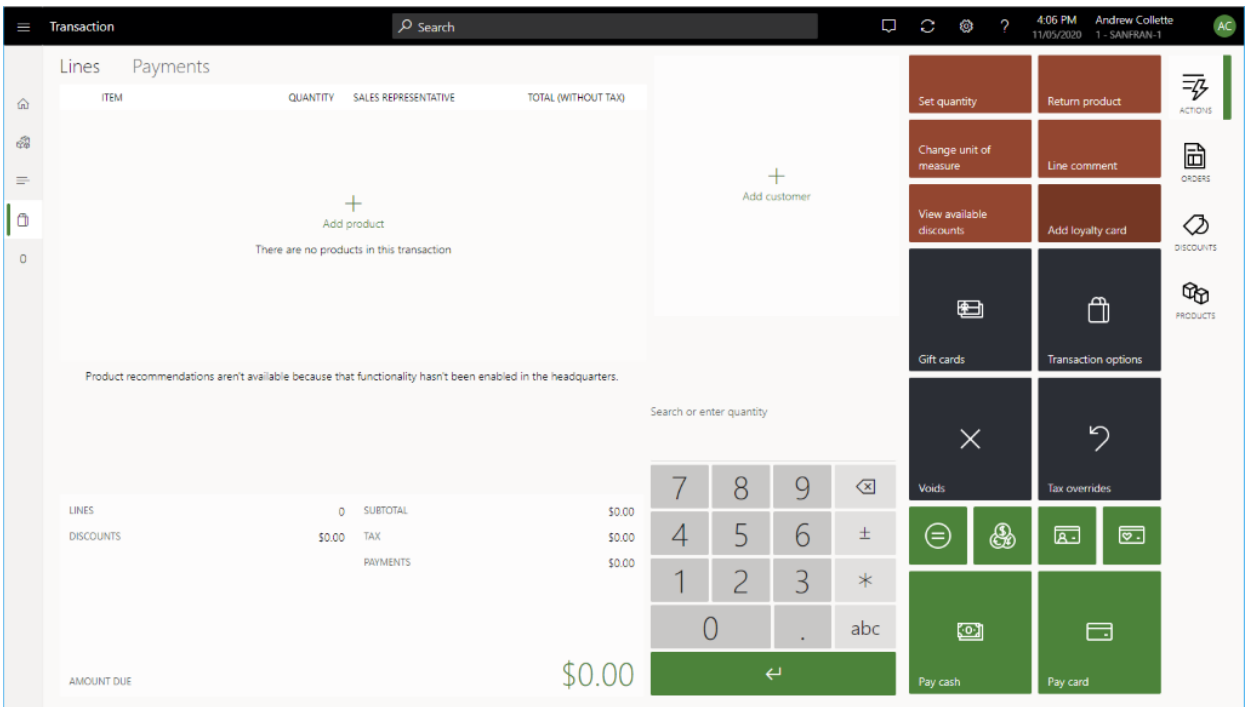
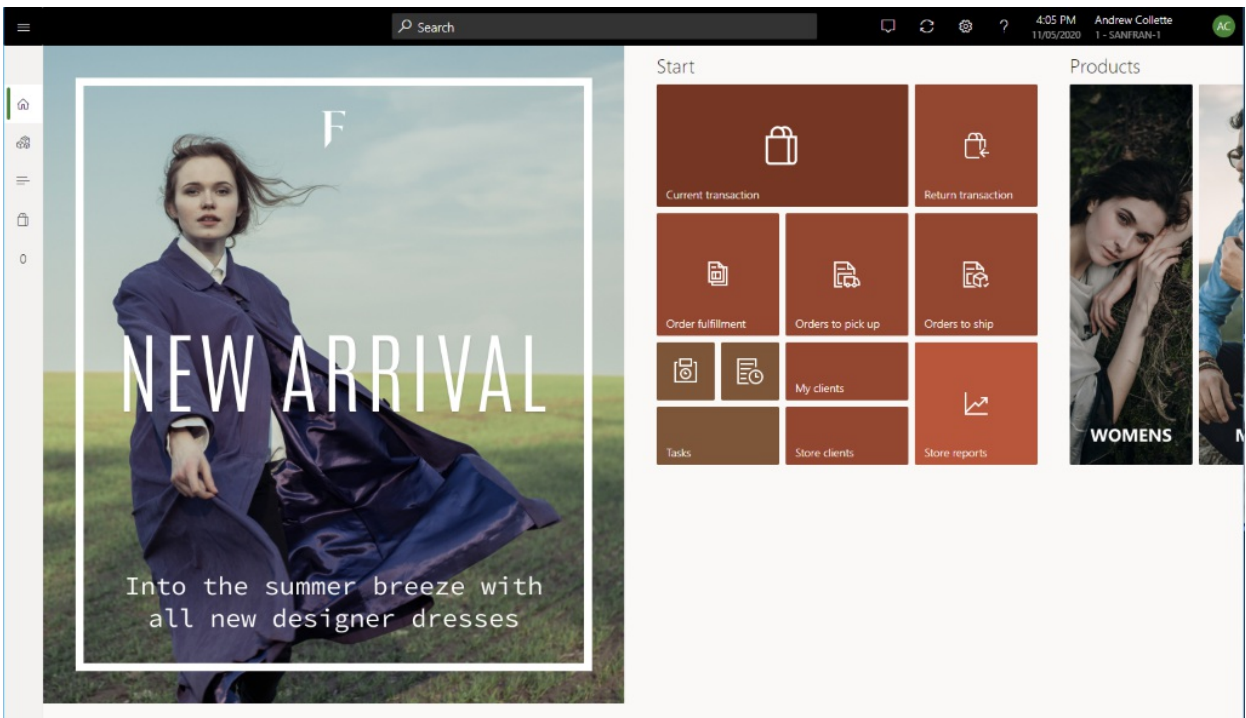
The following illustrations show examples of the welcome page and transaction page for the three fictitious companies.

### Adventure Works

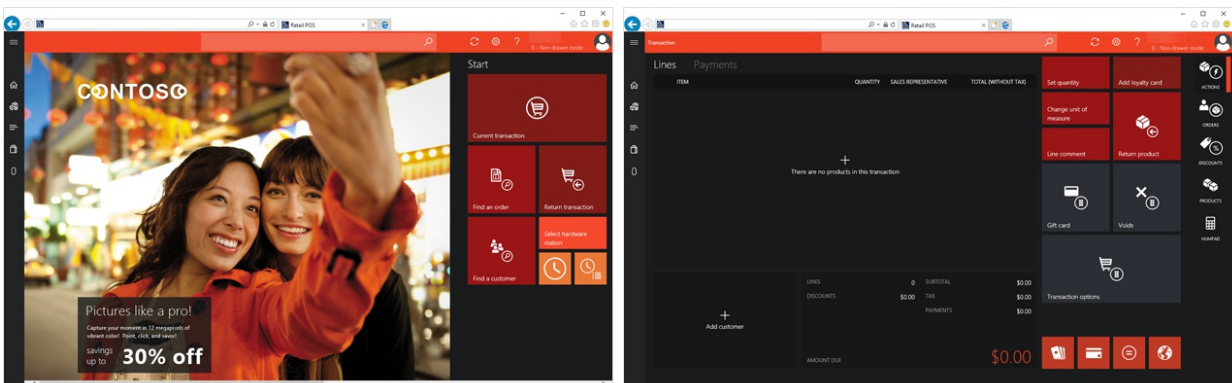


Fabrikam





## Contoso



## User sign in matrix

Users have been provided for the various screen layouts. By using the following table, you should be able to

access any of the screens. Just sign in by using an appropriate operator ID.

COMPANY	SCREEN LAYOUT ID	PERSONA	OPERATOR IDS
Adventure Works	A3MGR	Store Manager	000154, 000137, 000073
Adventure Works	A3CSH	Cashier	000150, 000175, 000165
Adventure Works	A3STK	Stock Clerk	000155, 000181, 000152
Fabrikam	F4MGR	Store Manager	000160, 000713
Fabrikam	F3CSH	Cashier	000161, 000113, 000114
Fabrikam	F3STK	Stock Clerk	000164, 000112, 000123
Contoso	C3MGR	Store Manager	000100, 000111
Contoso	C3CSH	Cashier	000110, 000120
Contoso	Not applicable	Stock Clerk	Not applicable

**TIP**

For best results, activate a register in the corresponding store location, and set the company to the company of the persona that you plan to use when you sign in. In this way, you help guarantee that the visual profile and branding images are aligned across the experience. For example, if you're interested in seeing a Fabrikam layout for a cashier, you should activate a register in the Houston store.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

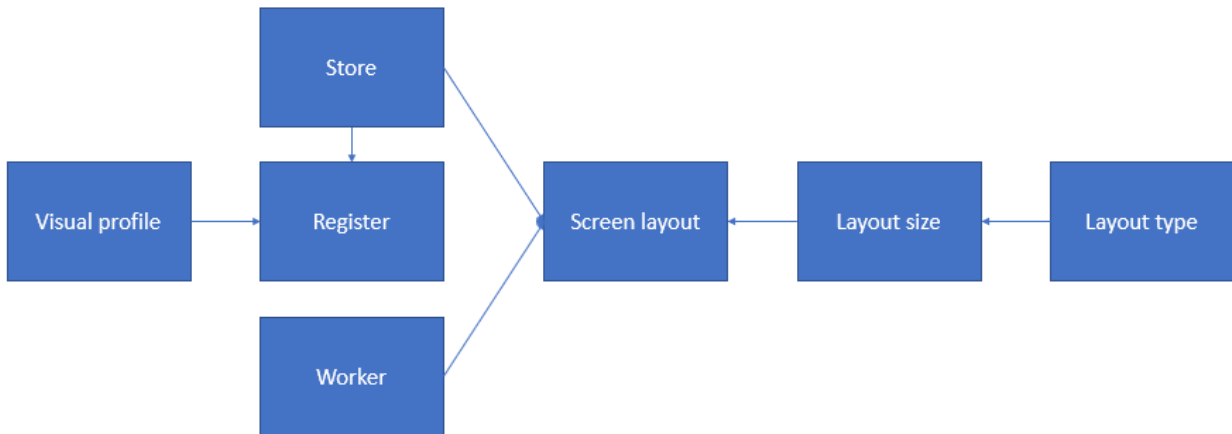
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# POS user interface visual configurations

2/18/2021 • 12 minutes to read • [Edit Online](#)

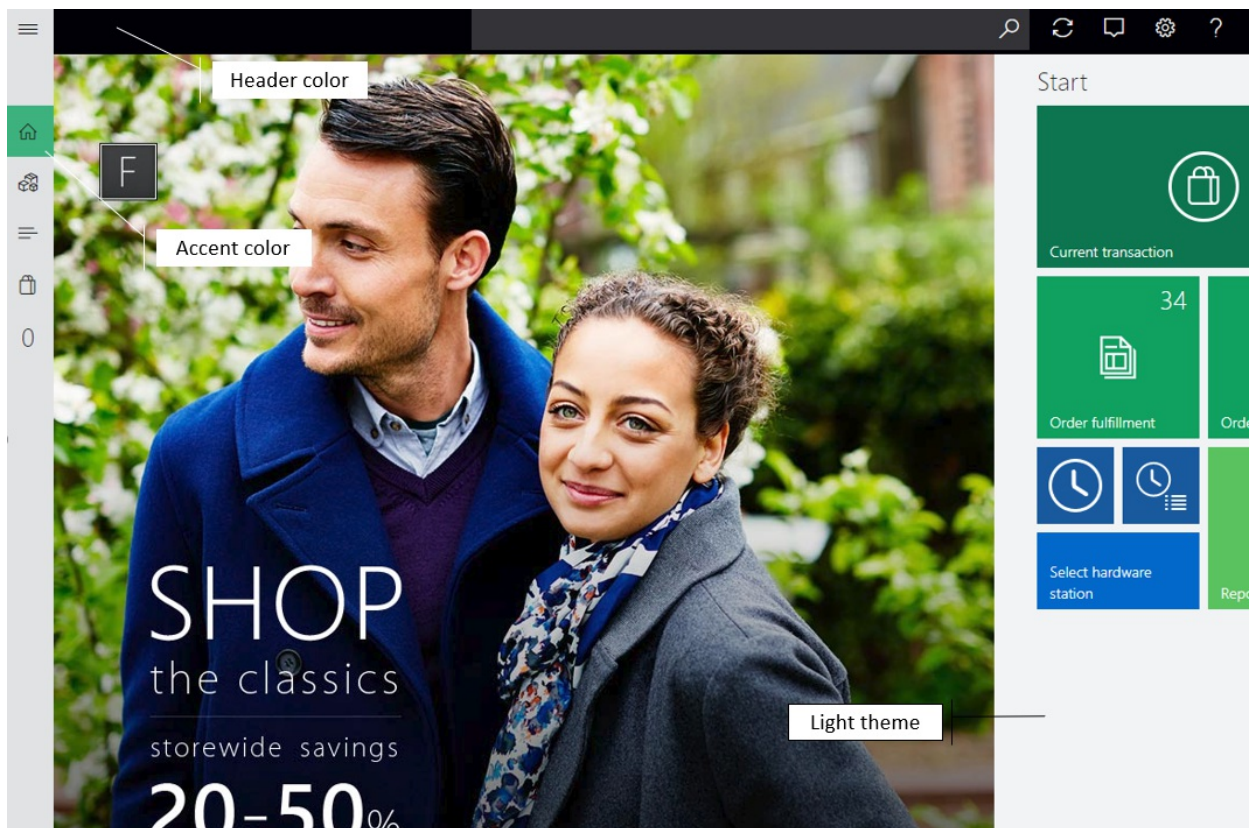
The user interface (UI) of the Microsoft Dynamics 365 Commerce point of sale (POS) can be configured by using a combination of visual profiles and screen layouts that are assigned to stores, registers, and users. This topic provides information about those configuration options.

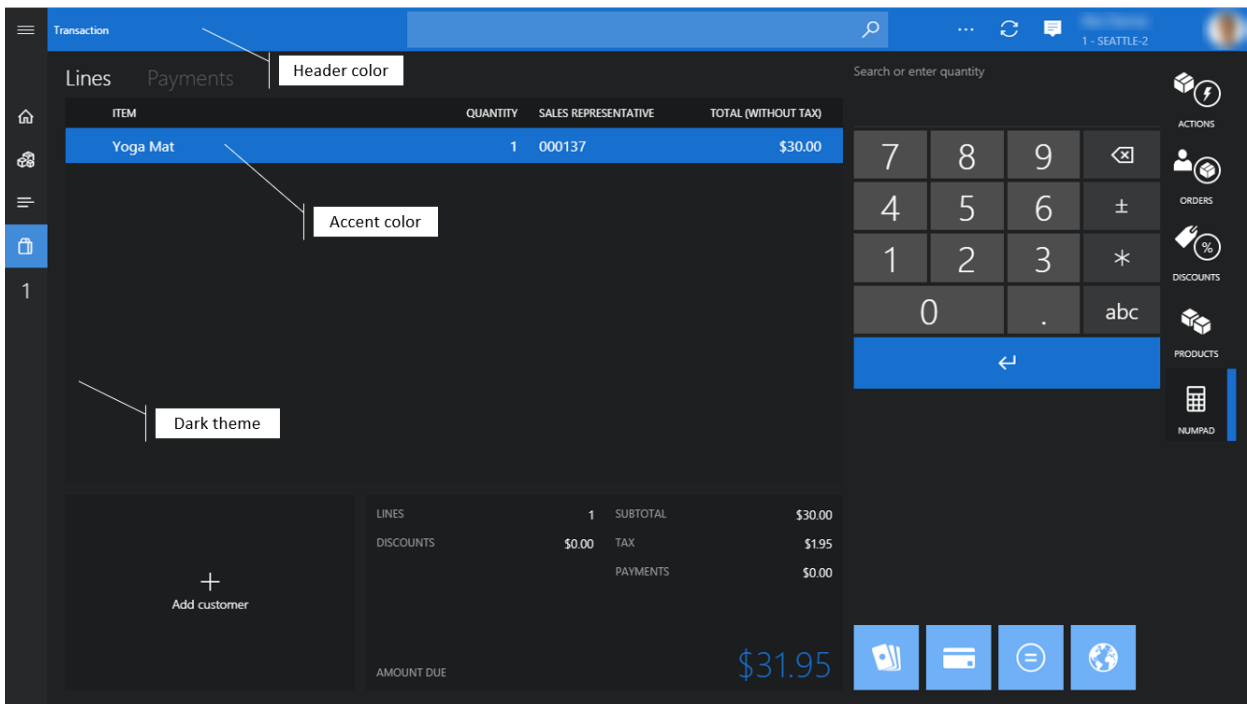
The following illustration shows the relationships among the various entities that make up the configurable aspects of the POS UI.



## Visual profile

Visual profiles are assigned to registers, and they specify the visual elements that are register-specific and shared across users. Every user who signs in to the register sees the same theme, layout, colors, and images.





- **Profile number** – The profile number is the unique identifier of the visual profile.
- **Description** – You can specify a meaningful name that will help identify the correct profile for your situation.
- **Theme** – You can select between the **Light** and **Dark** application themes. The theme affects the font and background colors throughout the application.
- **Accent color** – The accent color is used throughout the POS to differentiate or highlight specific visual elements, such as tiles, command buttons, and hyperlinks. Typically, these elements are actionable.
- **Header color** – You can configure the color of the page header to meet the retailer's branding requirements.
- **Font scheme** – You can select between the **Standard** and **Large** font schemes. The font scheme affects the font size throughout the application. The default selection is **Standard**.
- **Always show application bar labels** – When this option is turned on, the label text is always visible under the application bar buttons.
- **Layout** – You can select between the **Centered** and **Right** layouts. The layout affects the alignment of the sign-in box on the sign-in screen. The default selection is **Centered**.
- **Show date/time** – When this option is turned on, the current date and time are shown in the POS header and on the sign-in screen.
- **Keyboard** – You can select between **Default to OS keyboard** and **Show number pad** to specify the default keyboard that is used for input on the sign-in screen. The number pad is a virtual keyboard that is used primarily for touch-based devices. The default selection is **Default to OS keyboard**.
- **Logo image** – You can specify a logo image that is shown on the sign-in screen. We recommend that you use an image that has a transparent background. The file size should be kept as small as possible, because application behavior and performance can be affected when large files are stored and loaded.
- **Login background** – You can specify a background image for the sign-in screen. The file size of background images should be kept as small as possible.
- **Background** – You can specify a background image that is used instead of the solid theme color throughout the application. As for background images for the sign-in screen, the file size should be kept as small as possible.

#### NOTE

The **Right** layout and date/time display don't apply to the sign-in screen in compact view.

You need to run the **1090 (Registers)** distribution schedule job to synchronize the latest visual profile

configurations to the channel database.

## Screen layouts

Screen layout configurations determine the actions, content, and placement of UI controls on the POS **Welcome** screen and **Transaction** screen.

The screenshot shows the Dynamics 365 Retail POS setup interface for configuring screen layouts. The left sidebar lists various screen layout options, with 'F3MGR Fabrikam Manager' selected. The main area displays the configuration for the selected layout, including a table of layout sizes and a table of button grids.

Screen layout ID	Name	Default start screen
F3MGR	Fabrikam Manager	Welcome screen

Name	Layout type	Width	Height
1024x768 - Full	Modern POS - Full	1024	768
1280x720 - Full	Modern POS - Full	1280	720
1366x768 - Full	Modern POS - Full	1366	768
1440x960 - Full	Modern POS - Full	1440	960
480x853 - Compact	Modern POS - Compact	480	853

Layout zone	Button grid ID	Name
Welcome screen 1	F2W1M	Start
Welcome screen 2	F2W2	Products
Welcome screen 3	F2W3	Inventory
Welcome screen 4	F2W4M	Shift and drawer
Welcome screen 5	F2W5M	Operations
Transaction screen 1	F2T1M	Actions
Transaction screen 2	F2T2	Customer orders

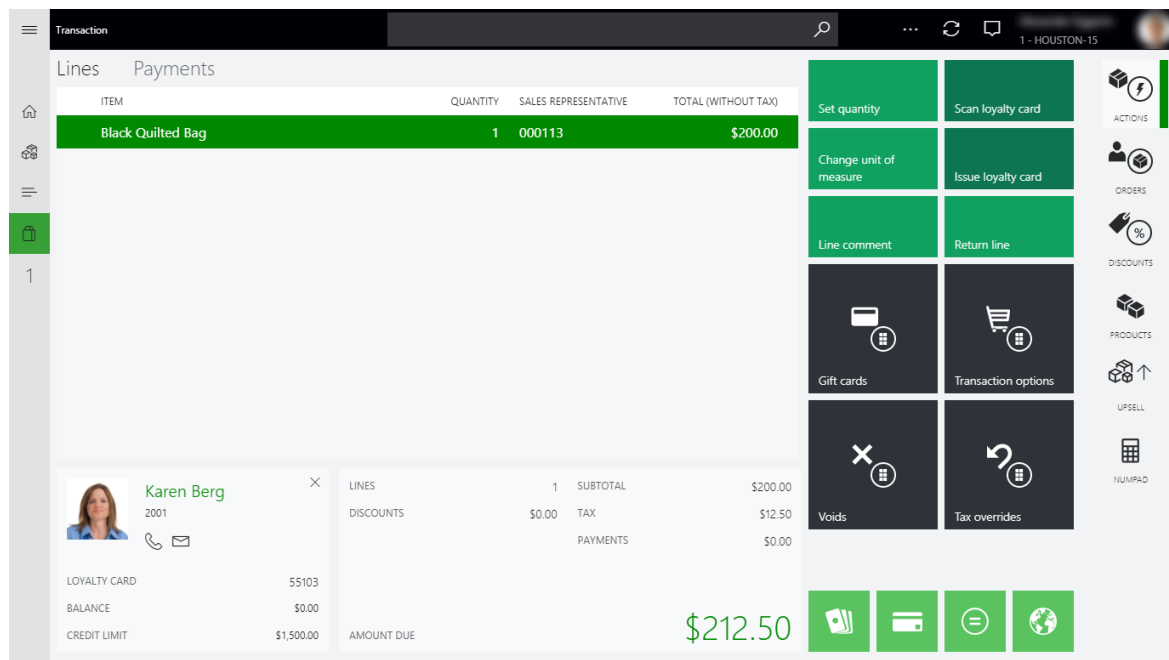
Layout zone	Image
Image 1	2506
Image 2	2507
Image 3	2508

Annotations in the image point to the 'Screen layout' sidebar, the 'Layout sizes' table, the 'Button grids' table, and the 'Images' table.

- **Welcome screen** – In most cases, the welcome screen is the page that users see when they first sign in to the POS. The welcome screen can consist of a branding image and button grids that provide access to POS operations. Typically, operations that aren't specific to the current transaction are put on this screen.

The screenshot shows the POS Welcome screen. The background features a large image of a person climbing a rock at sunrise, with the text 'RISE & SHINE! Earlybird specials are back again 20-40% SAVINGS'. The top right corner shows the user's name '1 - SEATTLE-2'. The main content area is a 'Start' screen with several button grids. The 'Welcome screen button grid' is highlighted, showing buttons for 'Current transaction', 'Return transaction', 'Order fulfillment', 'Orders to pick up', and 'Orders to ship'. The 'Welcome screen image' is also highlighted, showing the 'Select hardware station' and 'Reports' buttons.

- **Transaction screen** – The **Transaction** screen is the main screen in the POS for processing sales transactions and orders. The content and layout are configured by using the screen layout designer.



- **Default start screen** – Some retailers prefer that cashiers go directly to the **Transaction** screen after sign-in. The **Default start screen** setting lets you specify the default screen that appears after sign-in for each screen layout.

### Assignment

Screen layouts can be assigned at the store, register, or user level. The user assignment overrides the register and store assignments, and the register assignment overrides the store assignment. In a simple scenario where all users use the same layout, regardless of register or role, the screen layout can be set only at the store level. In scenarios where specific registers or users require specialized layouts, those layouts can be assigned.

Depending on which level the screen layouts are assigned, you need to run the **1070 (Channel configuration)**, **1090 (Registers)**, and/or **1060 (Staff)** distribution schedule jobs to synchronize the latest screen layout configurations to the channel database.

### Layout sizes

Most aspects of the POS UI are responsive, and the layout is automatically resized and adjusted based on the screen size and orientation. However, the POS **Transaction** screen must be configured for every screen resolution that is expected.

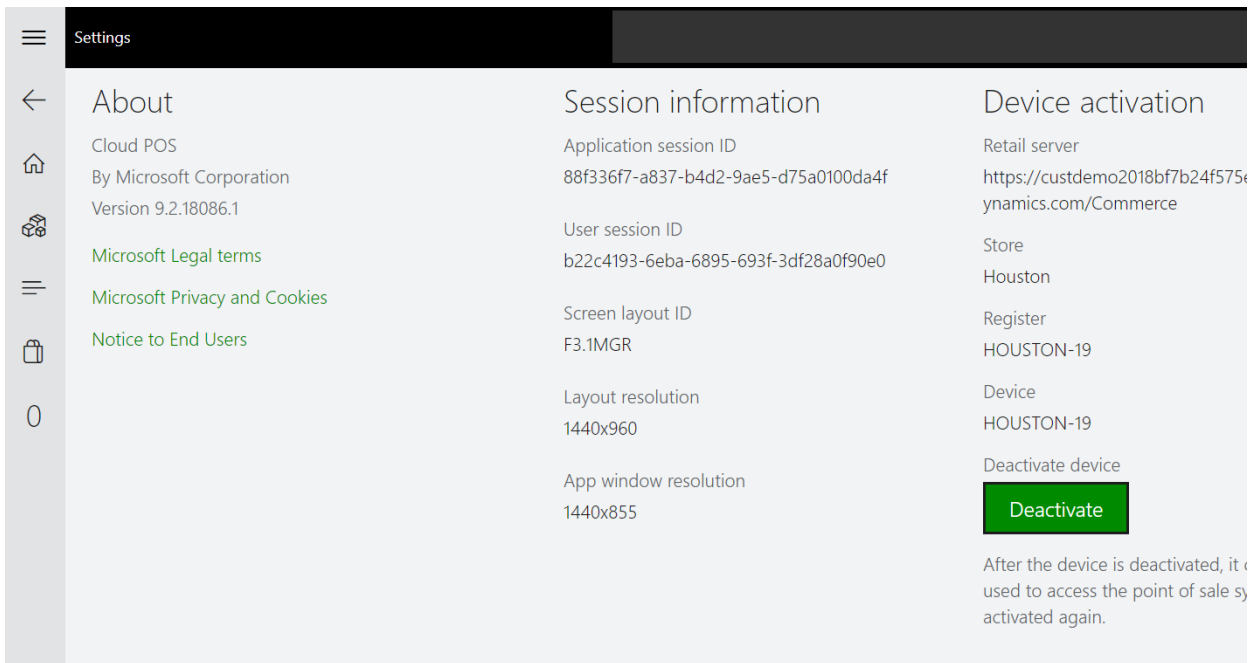
At startup, the POS application automatically selects the closest layout size that is configured for the device. A screen layout can also contain configurations for both landscape and portrait modes, and for both full-size and compact devices. Therefore, users can be assigned to a single screen layout that works across various sizes and form factors that are used in the store.



- **Name** – You can enter a meaningful name to identify the screen size.
- **Layout type** – The POS application can show its UI in various modes to provide the best user experience on a given device.
  - **Modern POS – Full** – Full layouts are typically best for larger displays, such as desktop monitors and tablets. You can select the UI elements to include, specify the size and placement of those elements, and configure their detailed properties. Full layouts support both portrait and landscape configurations.
  - **Modern POS – Compact** – Compact layouts are typically best for phones and small tablets. The design possibilities for compact devices are limited. You can configure the columns and fields for the receipt and totals panels.
- **Width/Height** – These values represent the effective screen size, in pixels, that is expected for the layout. Remember that some operating systems use scaling for high-resolution displays.

**TIP**

You can learn the layout size that is required for a POS screen by viewing the resolution in the app. Start the POS, and go to **Settings > Session information**. POS shows the screen layout that is currently loaded, the layout size, and the resolution of the app window.

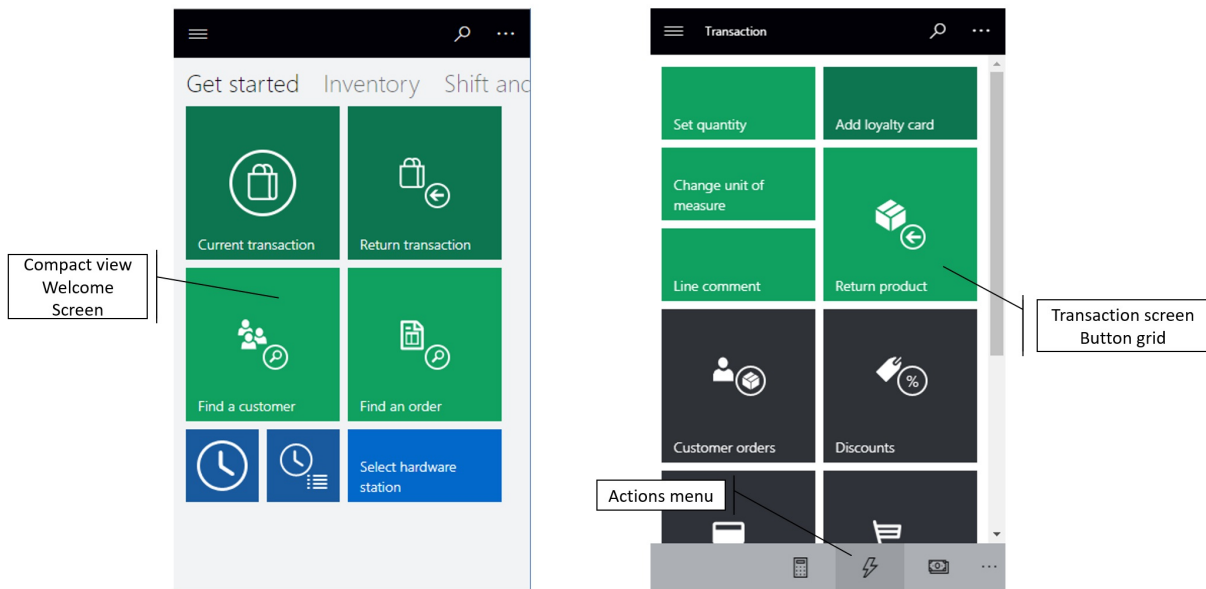


## Button grids

For each layout size in a screen layout, you can configure and assign button grids for the POS welcome screen and **Transaction** screen. Button grids for the welcome screen are automatically laid out from left to right, from the lowest number (Welcome screen 1) to the highest number.

In Full POS layouts, the placement of button grids is specified in the screen layout designer.

In Compact POS layouts, the button grids are automatically laid out from top to bottom, from the lowest number (Transaction screen 1) to the highest number. They can be accessed on the **Actions** menu.



### NOTE

The button sizes in the designer will scale to fit the size of the window, therefore they may not accurately reflect the actual buttons rendered in POS. To best simulate the button grid layout, adjust the designer windows to the same size as the POS.

## Images

For each layout size in a screen layout, you can specify images to include in the POS UI. For Full POS layouts, a



single image can be specified for the welcome screen. This image appears as the first UI element on the left. On the **Transaction** screen, images can be used as tab images or as a logo. Compact POS layouts don't use these images.

## Screen layout designer

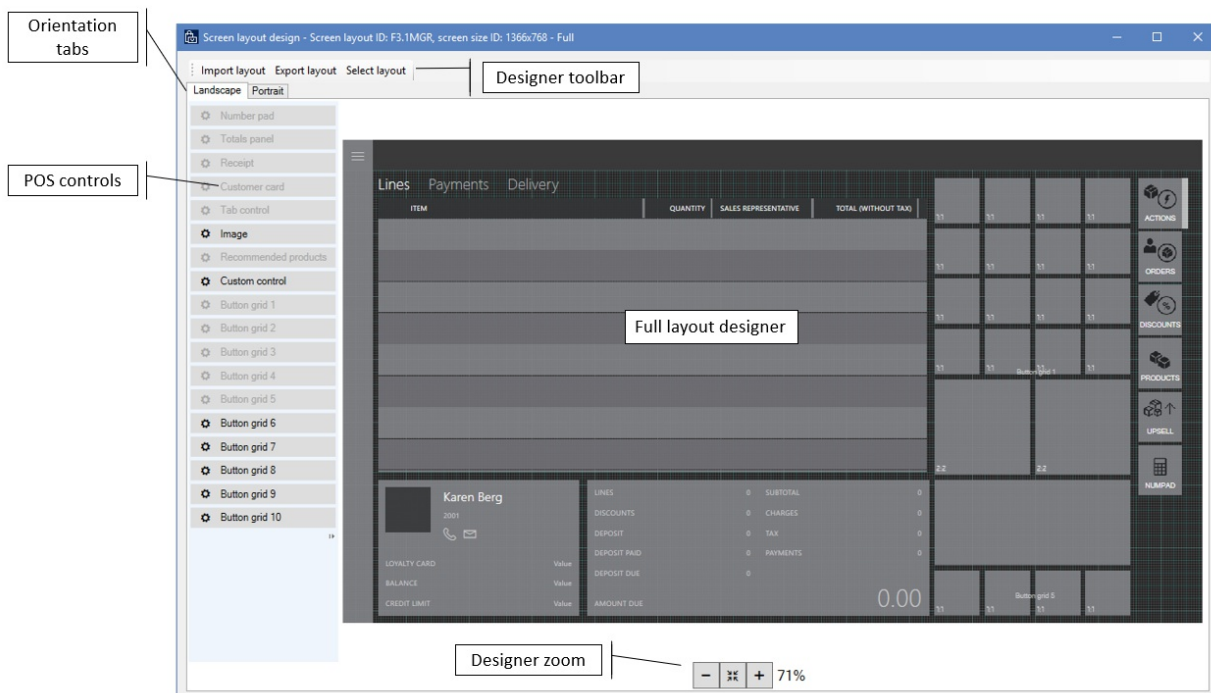
The screen layout designer lets you configure various aspects of the POS **Transaction** screen for each layout size, in both portrait and landscape modes, and for both Full and Compact layouts. The screen layout designer uses the ClickOnce deployment technology to download, install, and start the latest version of the application every time that users access it. Be sure to check the browser requirements for ClickOnce. Some browsers, such as Google Chrome, require extensions.

### IMPORTANT

You must configure a screen layout for each layout size that is defined and that is used by the POS.

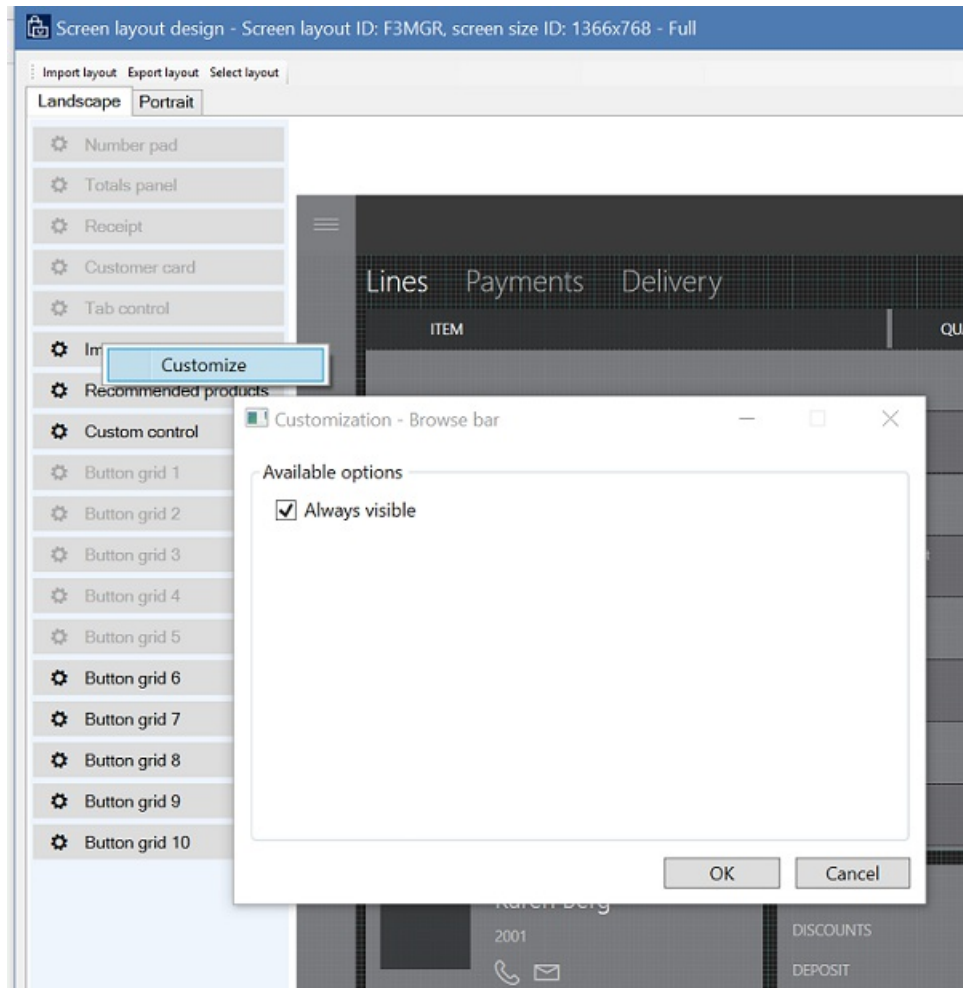
## Full layout designer

The Full layout designer lets users drag UI controls onto the POS **Transaction** screen and configure the settings of those controls.

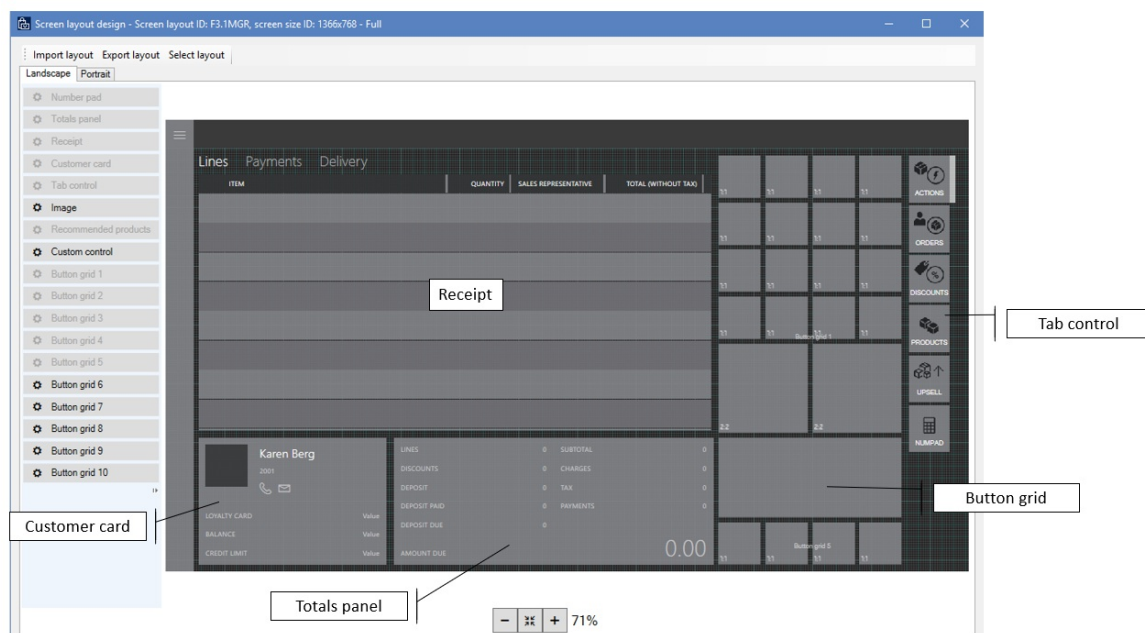


- **Import layout/Export layout** – You can export and import POS screen layout designs as XML files, so that you can easily reuse and share them across environments. It's important that you import layout designs for the correct layout sizes. Otherwise, UI elements might not fit correctly on the screen.
- **Landscape/Portrait** – If the POS device lets users switch between landscape and portrait modes, you must define a screen layout for each mode. The POS automatically detects screen rotation and shows the correct layout.
- **Layout grid** – The POS layout designer uses a 4-pixel grid. UI controls "snap" to the grid to help you correctly align the content.
- **Designer zoom** – You can zoom the designer view in and out to better view the content on the POS screen. This feature is useful when the screen resolution on the POS differs greatly from the resolution of the screen that is used in the designer.
- **Show/hide navigation bar** – For Full POS layouts, you can select whether the left navigation bar is visible on the **Transaction** screen. This feature is helpful for displays that have a lower resolution. To set

the visibility, right-click the navigation bar in the designer, and select or clear the **Always visible** check box. If the navigation bar is hidden, POS users can still access it by using the menu in the upper left.



- **POS controls** – The POS layout designer supports the following controls. You can configure many controls by right-clicking and using the shortcut menu.



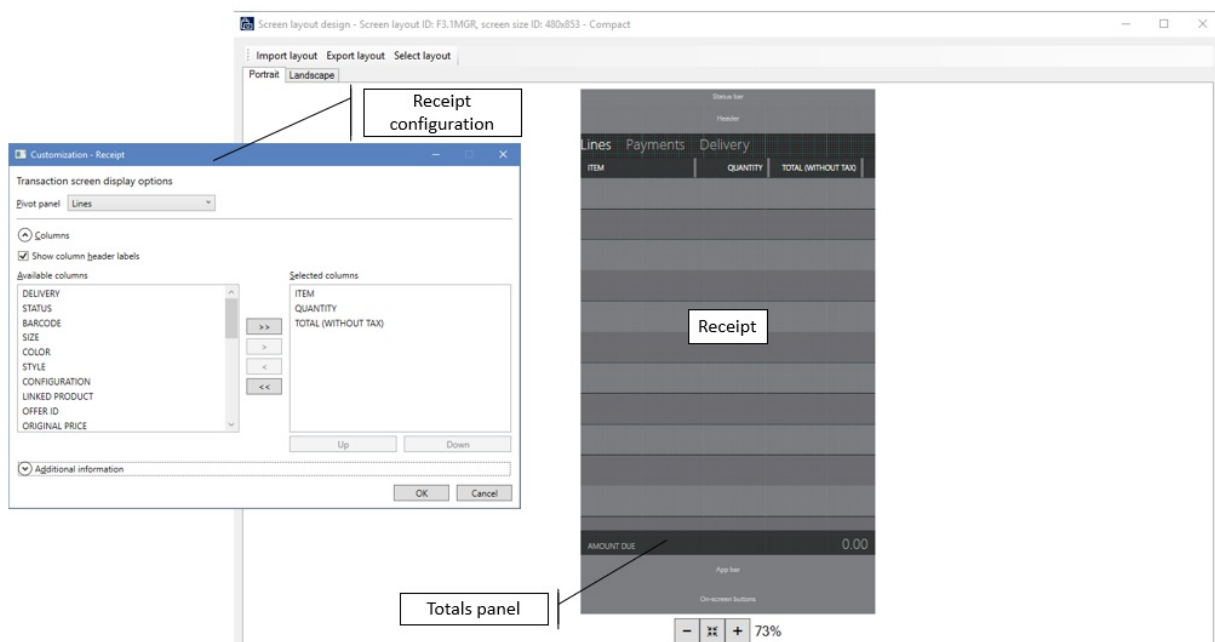
- **Number pad** – The number pad is the main mechanism for user input on the POS **Transaction** screen. You can configure the control so that the full number pad is shown. This option is ideal for touchscreen devices. Alternatively, you can configure it so that only the input field is shown. In this case, a physical keyboard is used for input. The number pad settings are available only for Full layouts.

For Compact layouts, the full number pad is always shown on the **Transaction** screen.

- **Totals panel** – You can configure the totals panel in either one column or two columns, to show values such as the line count, discount amount, charges, subtotal, and tax. Compact layouts support only a single column.
- **Receipt panel** – The receipt panel contains the sales lines, payment lines, and delivery information for the products and services that are processed in the POS. You can specify columns, widths, and placement. In Compact layouts, you can also configure additional information that appears in the row under the main line.
- **Customer card** – The customer card shows information about the customer who is associated with the current transaction. You can configure the customer card to hide or show additional information.
- **Tab control** – You can add the tab control to a screen layout, and then put other controls, such as the number pad, customer card, or button grids, in it. The tab control is a container that helps you fit more content on the screen. The tab control is available only for Full layouts.
- **Image** – You can use the image control to show the store's logo or another branding image on the **Transaction** screen. The image control is available only for Full layouts.
- **Recommended products** – If the recommended products control is configured for the environment, it shows product suggestions, based on machine learning.
- **Custom control** – The custom control acts as a placeholder in the screen layout and lets you reserve space for custom content. The custom control is available only for Full layouts.

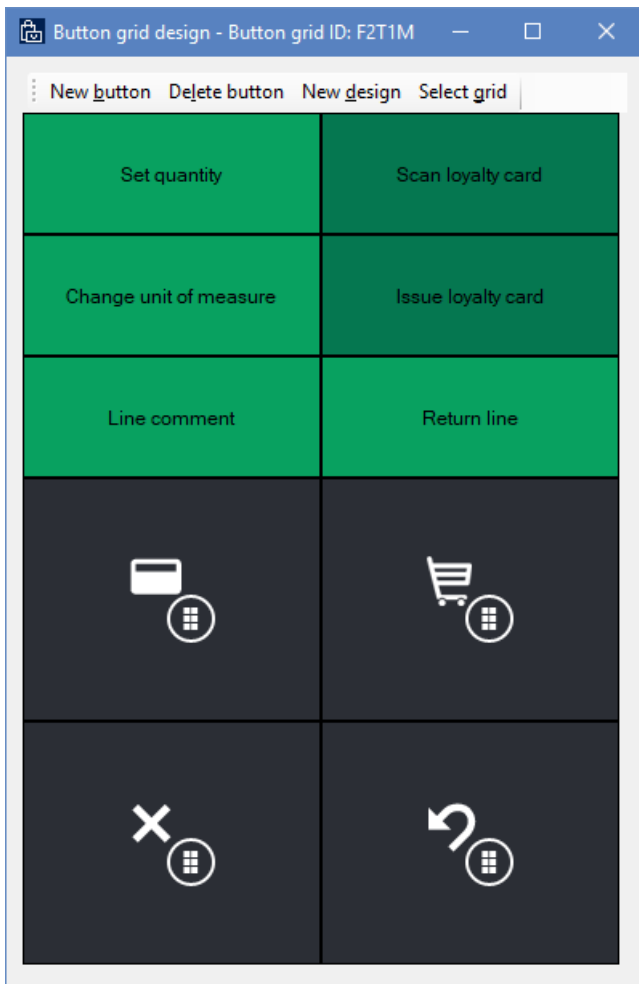
## Compact layout designer

Like the Full layout designer, the Compact layout designer lets you configure the POS screen layout for phones and small tablets. However, in this case, the layout itself is fixed. You can configure the controls in the layout by right-clicking and using the shortcut menu. However, you can't use drag-and-drop operations for additional content.



## Button grid designer

The button grid designer lets you configure button grids that can be used on the POS welcome screen and **Transaction** screen for both Full and Compact layouts. The same button grid can be used across layouts and layout types. Like the screen layout designer, the button grid designer uses the ClickOnce deployment technology to download, install, and start the latest version of the application every time that users access it. Be sure to check the browser requirements for ClickOnce. Some browsers, such as Google Chrome, require extensions.



- **New button** – Click to add a new button to the button grid. By default, new buttons appear in the upper-left corner of the grid. However, you can arrange buttons by dragging them in the layout.

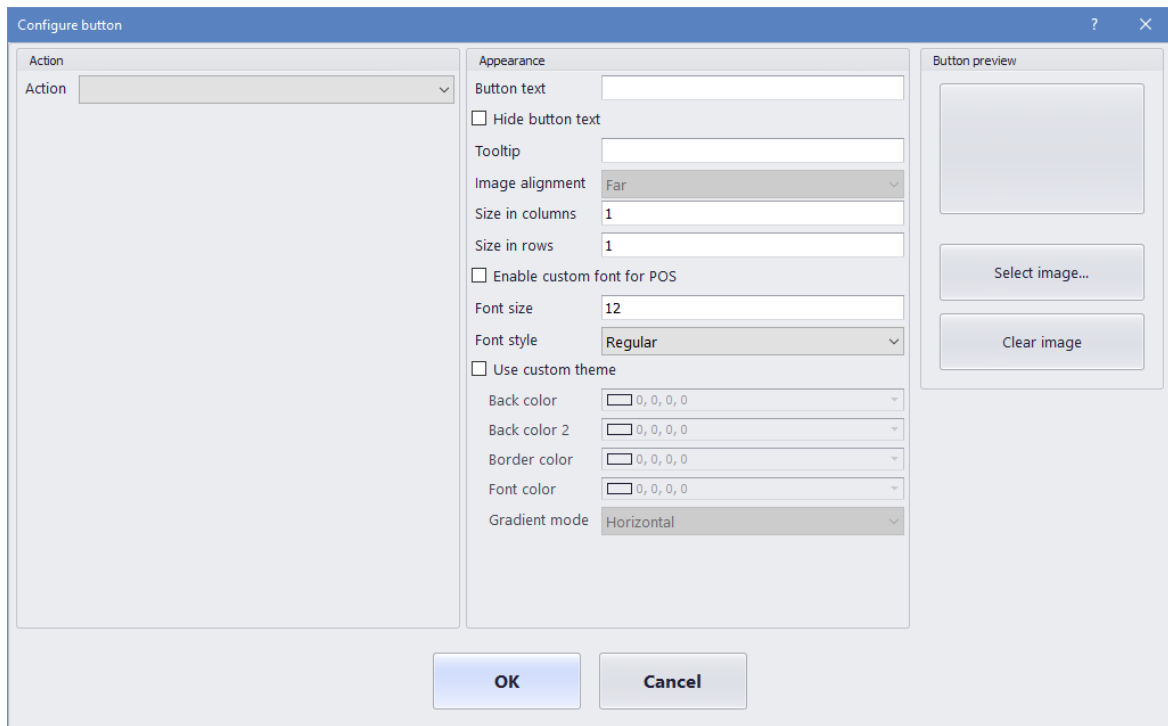
**IMPORTANT**

The contents of the button grid can overlap. When you arrange buttons, make sure that they don't hide other buttons.

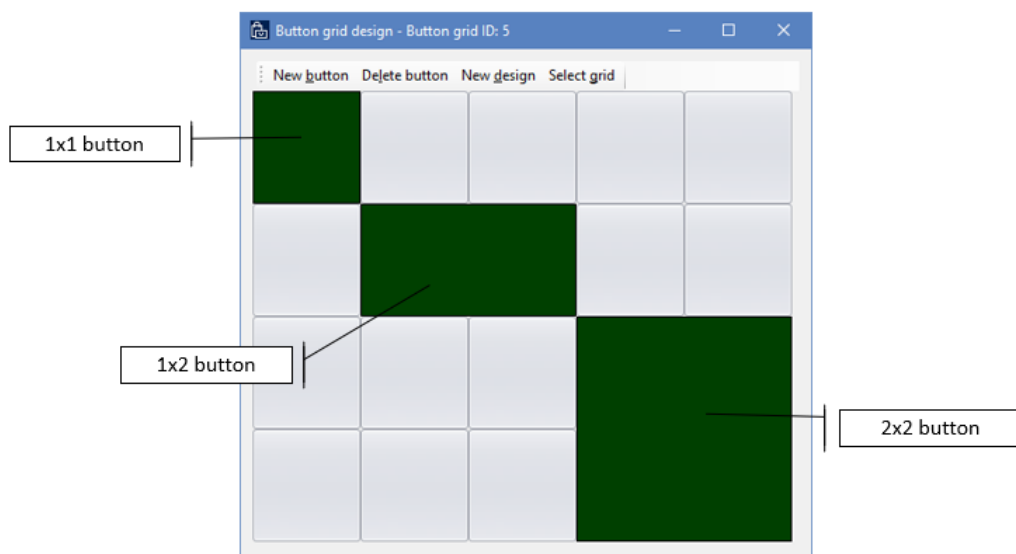
- **New design** – Click to automatically set up a button grid layout by specifying the number of buttons per row and column.
- **Button properties** – You can configure button properties by right-clicking the button and using the shortcut menu.

**IMPORTANT**

Some button grid settings apply only to Enterprise POS, not to Modern POS or Cloud POS.



- **Action** – In the list of applicable POS operations, select the operation that is invoked when the button is clicked in the POS.  
For the list of supported POS operations, see [Online and offline point of sale \(POS\) operations](#).
- **Action parameters** – Some POS operations use additional parameters when they are invoked. For example, for the Add product operation, users can specify the product to add.
- **Button text** – Specify the text that appears on the button in the POS.
- **Hide button text** – Use this check box to hide or show the button text. Button text is often hidden for small buttons that show only an icon.
- **Tooltip** – Specify additional Help text that appears when users mouse over the button.
- **Size in columns/Size in rows** – You can specify how tall and wide the button is.



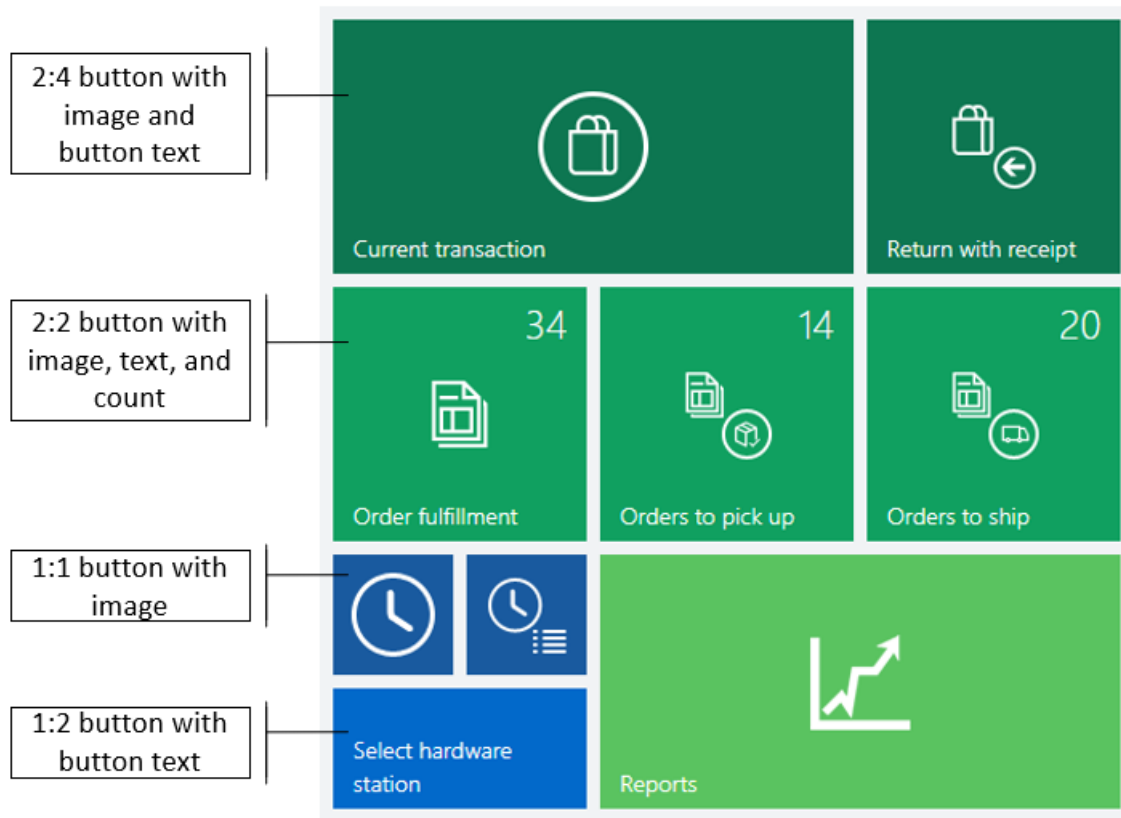
- **Custom font** – When you select the **Enable custom font for POS** check box, you can specify a font other than the default system font for the POS.

- **Custom theme** – By default, POS buttons use the accent color from the visual profile. When you select the **Use custom theme** check box, you can specify additional colors.

**NOTE**

Modern POS and Cloud POS use only the **Back color** and **Font color** values.

- **Button image** – Buttons can include images or icons. Select among the available images that are specified at **Retail and Commerce > Channel setup > POS setup > POS > Images**.



## Additional resources

[Install the Retail point of sale \(POS\) layout designer](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Install the POS layout designer

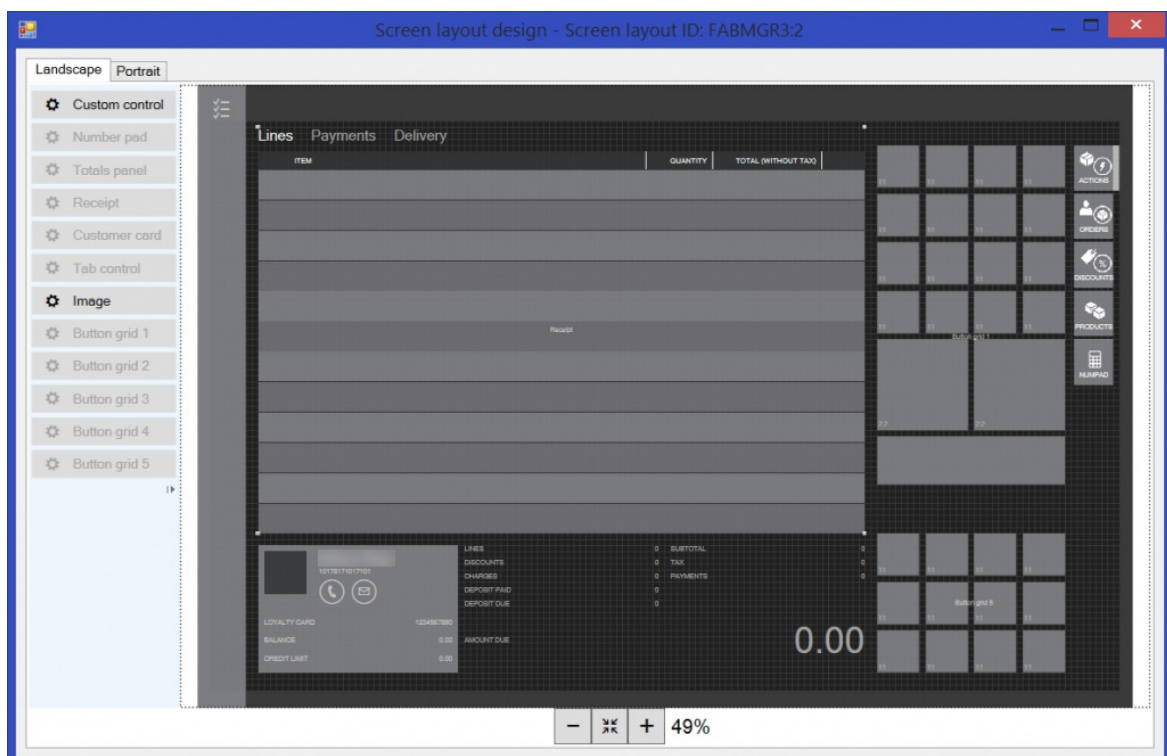
2/18/2021 • 2 minutes to read • [Edit Online](#)

You can use the one-click designer to design different Modern POS (MPOS) and Cloud POS layouts, in either Landscape mode or Portrait mode, for stores, registers, cashiers, and managers.

The graphical design interface for MPOS or Cloud POS is controlled by the till layout. A layout controls the position of various objects. Examples include the total layout, the item grid layout, the customer layout, the payment layout, and the layout of various menu buttons. Layouts also include the overall appearance of the sales interface that is presented to workers.

## Install the one-click designer

1. In Commerce, use the menu in the upper left to navigate to **Retail and Commerce > Channel setup > POS setup > POS > Screen layouts**.
2. Select any layout that has an application type of **Modern POS for Windows** or **Cloud POS**, and then click **Layout designer**.
3. On the notification bar that appears at the bottom of the Internet Explorer window, click **Open** to install the one-click designer. (The notification bar might appear in a different place in other browsers.)
4. In the **Application Run - Security Warning** message box that appears, click **Run** to install the Retail designer host. A progress indicator shows the progress of the installation.
5. After the installation is completed, on the **Sign in** page, enter your Commerce user name and password, and then click **Sign in** to start the designer.
6. After your credentials are validated and the designer starts, you can design your own layout or modify the existing layout.



## Troubleshoot the installation of the Layout designer

- When you click **Designer**, the prompt to download (or run) the installer doesn't appear, or your current security settings don't allow you to download the file.

### Solutions:

- In Internet Explorer, make sure that the pop-up blocker is disabled for this site. Click **Settings > Options > Privacy > Find Pop-up Blocker**, and change the setting, if a change is required.
- In Internet Explorer, add the Commerce URL to your trusted sites. Click **Settings > Options > Security > Trusted sites > Sites > Add**.
- The program doesn't start, and you're instructed to contact the vendor.

**Solution:** In Internet Explorer, add the Commerce URL to your trusted sites. Click **Setting > Options > Security > Trusted sites > Sites > Add**.

**Known issue:** The designer doesn't work correctly in the Google Chrome and Mozilla Firefox browsers. We are working to fix this issue.

## Additional resources

[Configure, install, and activate Retail Modern POS \(MPOS\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Show order notifications in the point of sale (POS)

2/18/2021 • 5 minutes to read • [Edit Online](#)

In the modern retail environment, store associates are assigned various tasks, such as helping customers, entering transactions, doing stock counts, and receiving orders in the store. The point of sale (POS) client provides a single application where associates can perform all these tasks and many others. Because various tasks must be performed during the day, associates might have to be notified when something requires their attention. The notification framework in the POS helps by letting retailers configure role-based notifications. As of Dynamics 365 for Retail with application update 5, these notifications can be configured only for POS operations.

Currently, the system can show notifications only for order fulfillment operations. However, because the framework is designed to be extensible, developers will eventually be able to write a notification handler for any operation and show the notifications for that operation in the POS.

## Enable notifications for order fulfillment operations

To enable notifications for order fulfillment operations, follow these steps.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS > Operations**.
2. Search for the **Order fulfillment** operation, and select the **Enable notifications** check box for it to specify that the notification framework should listen to the handler for this operation. If the handler is implemented, notifications for this operation will then be shown in the POS.
3. Go to **Retail and Commerce > Employees > Workers >**, under Commerce tab, open the POS permissions associated with the worker. Expand the **Notifications** FastTab, add the **Order fulfillment** operation, and set the **Display order** field to 1. If more than one notification is configured, this field is used to arrange the notifications. Notifications that have a lower **Display order** value appear above notifications that have a higher value. Notifications that have a **Display order** value of 1 are at the top.

Notifications are shown only for operations that are added on the **Notifications** FastTab, and you can add operations there only if the **Enable notifications** check box for those operations has been selected on the **POS operations** page. Additionally, notifications for an operation are shown to workers only if the operation is added to the POS permissions for those workers.

### NOTE

Notifications can be overridden at the user level. Open the worker's record, select **POS permissions**, and then edit the user's notification subscription.

4. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. In the **Notification interval** field, specify how often notifications should be pulled. For some notifications, the POS must make real-time calls to the back-office application. These calls consume the compute capacity of your back-office application. Therefore, when you set the notification interval, you should consider both your business requirements and the impact of real-time calls to the back-office application. A value of 0 (zero) turns off notifications.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**. Select the **1060 (Staff)** schedule to synchronize notification subscription settings, and then select **Run now**. Next, select the **1070 (Channel configuration)** schedule to synchronize the permission interval, and then select

Run now.

## View notifications in the POS

After you complete the preceding steps, the workers will be able to view the notifications in the POS. To view notifications, press the notification icon in the top right corner of the POS. A notification center appears and shows notifications for the order fulfillment operation. The notification center should show the following groups in the order fulfillment operation:

- **Store pickup** – This group shows the count of orders that have a delivery mode of **Pickup**, and that are scheduled for pickup from the current store. You can press the number on the group to open the **Order fulfillment** page. In this case, the page will be filtered so that it shows only the active orders that are set up for pickup from the current store.
- **Ship from store** – This group shows the count of orders that have the delivery mode of **Shipping**, and that are scheduled for shipment from the current store. You can press the number on the group to open the **Order fulfillment** page. In this case, the page will be filtered so that it shows only the active orders that are set up for shipment from the current store.

When new orders are assigned to the store for fulfillment, the notification icon changes to indicate that there are new notifications, and the count for the appropriate groups is updated. Even though the groups are refreshed at regular intervals however, POS users can manually refresh the groups at any time by selecting the **Refresh** button next to the group. Lastly, if a group has a new item, that the current worker hasn't viewed, then the group shows a burst symbol to indicate new content.

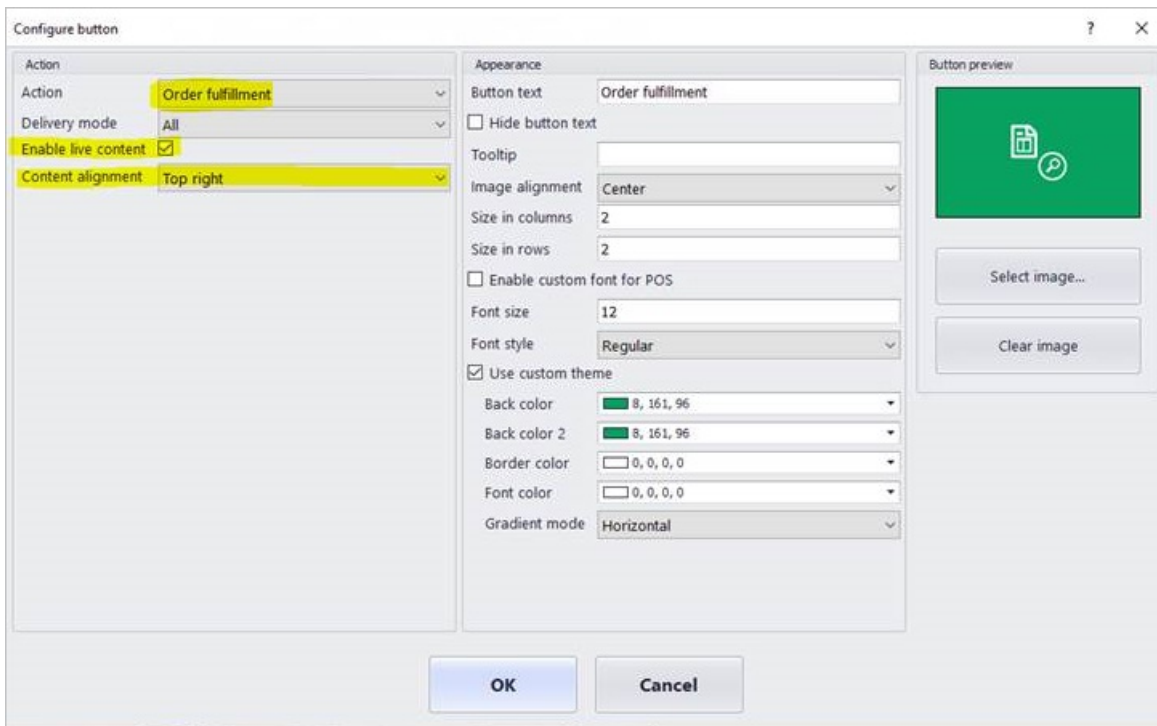
## Enable live content on POS buttons

POS buttons can now show a count to help workers easily determine which tasks require their immediate attention. To show this number on a POS button, you must complete the notification setup that is described earlier in this topic (that is, you must enable notifications for an operation, set up a notification interval, and update the POS permission group for the worker). Additionally, you must open the button grid designer, view the button's properties, and select the **Enable live content** check box. In the **Content alignment** field, you can select whether the count appears in the upper-right corner of the button (**Top right**) or in the center (**Center**).

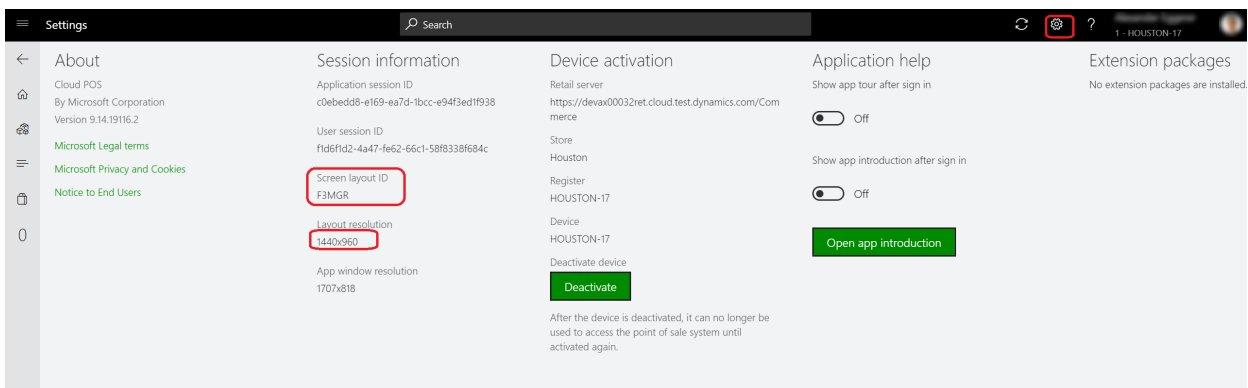
### NOTE

The live content can be enabled for operations only if the **Enable notifications** check box has been selected for them on the **POS operations** page, as described earlier in this topic.

The following illustration shows the live content settings in the button grid designer.



To show the notification count on a button, you need to ensure that the correct screen layout is being updated. To determine the screen layout that is being used by the POS, select the **Settings** icon in upper-right corner and note the **Screen layout ID** and **Layout resolution**. Now using Edge browser, go to the **Screen layout** page, find the **Screen layout ID** and **Layout resolution** identified above and select the **Enable live content** check box. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule** and run the 1090 (Registers) job to synchronize layout changes.



The following illustration shows the effect of selecting **Top right** versus **Center** in the **Content alignment** field for buttons of various sizes.

The image shows a Windows Start menu on the left and a Notifications window on the right. The Start menu features several tiles: 'Current transaction' (shopping cart icon), 'Return transaction' (shopping cart with arrow icon), 'Order fulfillment - ALL' (two tiles, each with a document icon and a count of 1), 'Order fulfillment - Pick Up' (calendar icon, count 1), 'Order fulfillment - Ship' (calendar with checkmark icon, count 0), and 'Select hardware station' (two small tiles, each with a count of 1). The Notifications window is titled 'Order fulfillment' and shows 'Last updated: 01/17/2018 5:43 PM'. It contains two sections: 'Store pickup' with a count of 1, and 'Ship from store' with a count of 0.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Point of sale (POS) application and user language settings

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to change language settings in Modern POS (MPOS) and Cloud POS.

## Overview

Modern POS (MPOS) and Cloud POS support environments where language settings and translations can vary between the store and user settings. For example, the store could be located in a region where English is most common for their customers, but some workers prefer to use the application with French translations.

## Data language

Regardless of the user's settings, MPOS and Cloud POS will always use the store's language settings to determine the translations used for data. This will ensure that all users and customers will have a consistent experience. Examples of data include:

- Products
- Attributes and values
- Category names
- Printed or emailed transaction receipts
- Payment method names
- Line display messages

The store's language will also be used for the main POS login screen, since the user is not known before logging in. If a translation is not available for the store's language, the POS will revert to the company's language.

### Configuring the store's language setting

The store's language setting is set from **All stores** on the **Store** page under **General > Regional Settings > Language**. Use the drop-down list to choose the language for each store.

## User interface language

The POS user's language setting determines the translations used in the application user interface. This includes all labels, menus, and lists that are not considered data. One exception is the text that is displayed on POS button grids. The button grids don't support translations, so they will always show the text as defined on the button. In order to support translated buttons, you'll have to copy and maintain separate button grids and assign them to the users as appropriate.

### Configuring the user's language setting

The POS user's language setting is set from **All workers** on the **Worker** page under **Retail and Commerce > Language**. It is not set on the main Profile tab. This setting is not used by POS. If the user's language is not set or it is set to a language where translations are not available, the POS will revert to the store's language.

	UI LANGUAGE	DATA LANGUAGE (PRODUCTS, RECEIPT FORMATS, LINE DISPLAY, ETC.)
Company	Default	Default
Store	Overrides company	Overrides company
User	Overrides store or company	Never

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up and manage images for Modern POS (MPOS)

2/18/2021 • 14 minutes to read • [Edit Online](#)

This article explains the steps that are involved in setting up and managing images for the various entities that appear in Modern POS (MPOS).

## Setting up the media base URL and defining media templates to configure the format for image URLs

The images that appear in Modern POS (MPOS) must be hosted externally, outside of Commerce. Typically, they are hosted in a content management system, content delivery network (CDN), or media server. MPOS then fetches and displays the images for the appropriate entities, such as products and catalogs, by accessing the target URL. To fetch these externally hosted images, MPOS requires the correct URL format for the images. You can configure the required URL format for the images by setting up the **Media base URL** value in the channel profile and using the **Define media template** functionality for each entity. You can also overwrite the standard URL format for a subset of entities by using the **Edit in Excel** functionality.

### IMPORTANT

In the current version of Commerce, you can no longer set up the URL format by using the **Image** attribute XML for MPOS in the **Default** attribute group for entities. If you're familiar with Microsoft Dynamics AX 2012 R3 and are now using the current version of Commerce, make sure that you always use the new **Define media template** functionality to set up images. Don't use or modify the **Image** attribute in the **Default** attribute group for any entities, including products. Changes that you make directly in the **Default** attribute group for images won't be reflected. This option will be disabled in a future release.

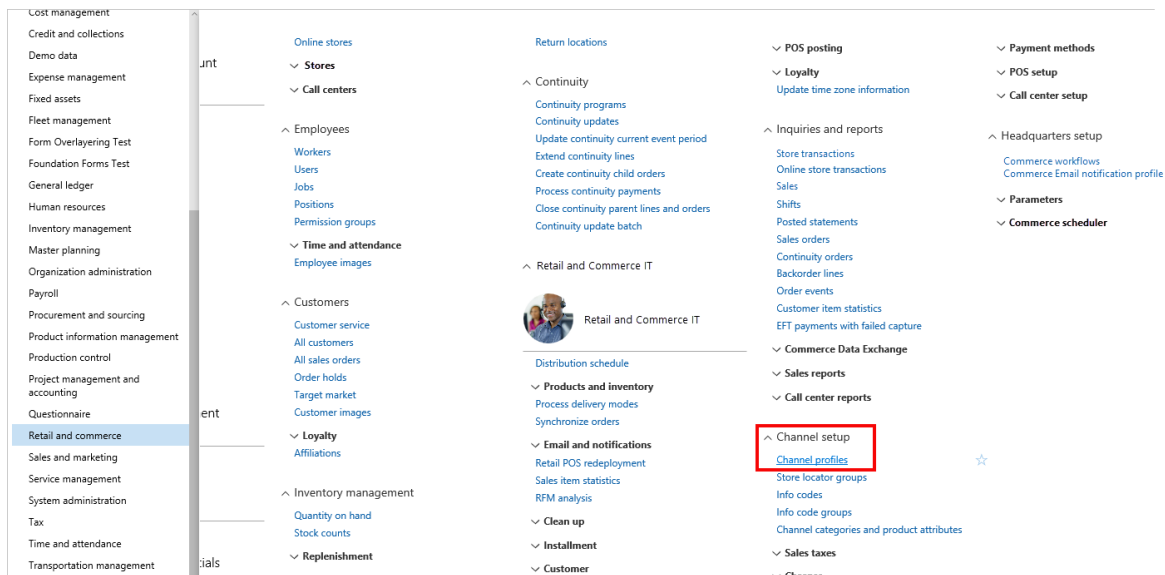
In the following procedures, images are set up for the Catalog entity as an example. These procedures will help guarantee that the correct image destination path is set implicitly for all catalog images that use a common path. For example, if you've set up a media server or CDN externally, and want the images to appear in MPOS for a given store, the **Define media template** functionality helps you set the path where MPOS can look up and retrieve the images.

### NOTE

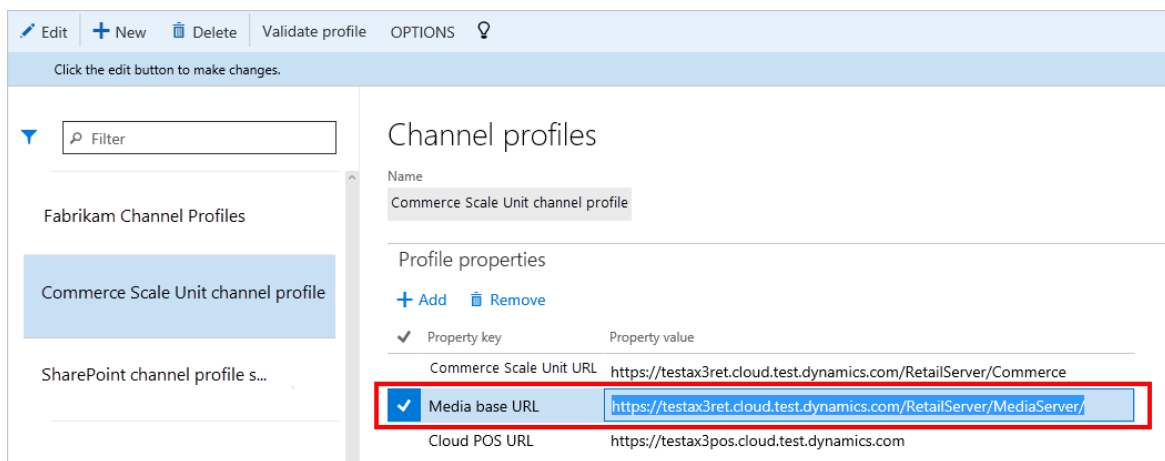
For this demo data example, the media server is deployed on the Commerce Scale Unit. However, you can have it anywhere outside Commerce.

### Set up the media base URL for a channel

1. Open the Commerce HQ portal.
2. Click **Retail and Commerce > Channel setup > Channel profiles**.



3. In the channel profile that your store uses for MPOS, update the **Media base URL** field with the base URL of your media server or CDN. The base URL is the first part of the URL that is shared by all image folders of different entities.



### Define the media template for an entity

1. Click **Retail and Commerce > Catalog management > Catalog images**.
2. On the **Catalog images** page, on the Action Pane, click **Define media template**. In the **Define media template** dialog box, in the **Entity** field, **Catalog** should be selected by default.
3. On the **Media path** FastTab, enter the remaining path of the image location. The media path supports **LanguageID** as a variable. For example, for the demo data, you can create a **Catalogs** folder for all catalog images under the media base URL for your media server ( `https://testax3ret.cloud.test.dynamics.com/RetailServer/MediaServer` ). You can then have a folder for each language, such as en-US or fr-FR, and copy the appropriate images under each folder. If you don't have different images for the various languages, you can omit the **LanguageID** variable from your folder structure and point directly to the **Catalogs** folder that contains the catalog images.

#### NOTE

The current version of Commerce supports the **{LanguageID}** token for Catalog, Product, and Category entities. (The **{LanguageID}** token isn't supported for Customer and Worker entities, according to the existing standard that has been effective since Microsoft Dynamics AX 6.x.)

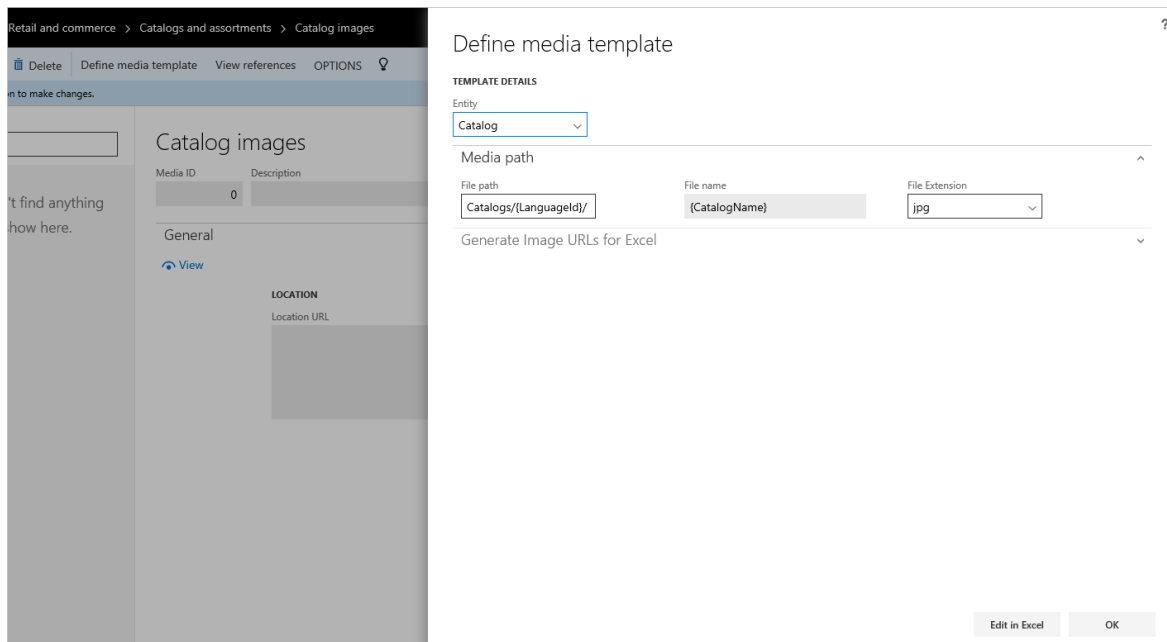
4. For images, the file name format is hard-coded to the catalog name and can't be changed. Therefore, rename your images so that they have appropriate catalog names, to help guarantee that MPOS handles



them correctly.

5. In the **File Extension** field, select the expected file name extension, depending on the type of images that you have. For example, for the demo data, the catalog images are set to the .jpg extension. (The image files are also renamed so that they have catalog names.)
6. Click **OK**.
7. To validate that the media template for images has been saved correctly, on the **Catalog images** page, click **Define media template** again. To validate the template without closing the **Define media template** dialog box, you can use the **Generate Image URLs for Excel** FastTab. Check the appearance of the image URL, and verify that the URL complies with the template standard that was mentioned earlier. The **Define media template** dialog box has now set the image path implicitly for all catalog images that use this common URL path. This URL path applies to all catalog images unless they are overwritten. The first part of the image path is taken from the media base URL that you defined in the channel profile. The remaining part of the path is taken from the path that you defined in the media template. The two parts are concatenated to provide the full URL of the image location. For example, a catalog in the demo data is named Fabrikam Base Catalog. Therefore, the image name must be Fabrikam Base Catalog.jpg so that it uses the catalog name and the .jpg file name extension that is configured in the template. In this case, after concatenation, the URL will be  

```
https://testax3ret.cloud.test.dynamics.com/RetailServer/MediaServer/Catalogs/en-US/Fabrikam Base Catalog.jpg
```
8. Run the synchronization jobs to push the new template to the channel database, so that MPOS can use the template to access the images.
9. To update the media template for catalog images on the channel side, be sure to run **Catalog Job 1150** from **Retail and Commerce IT > Distribution schedule**.

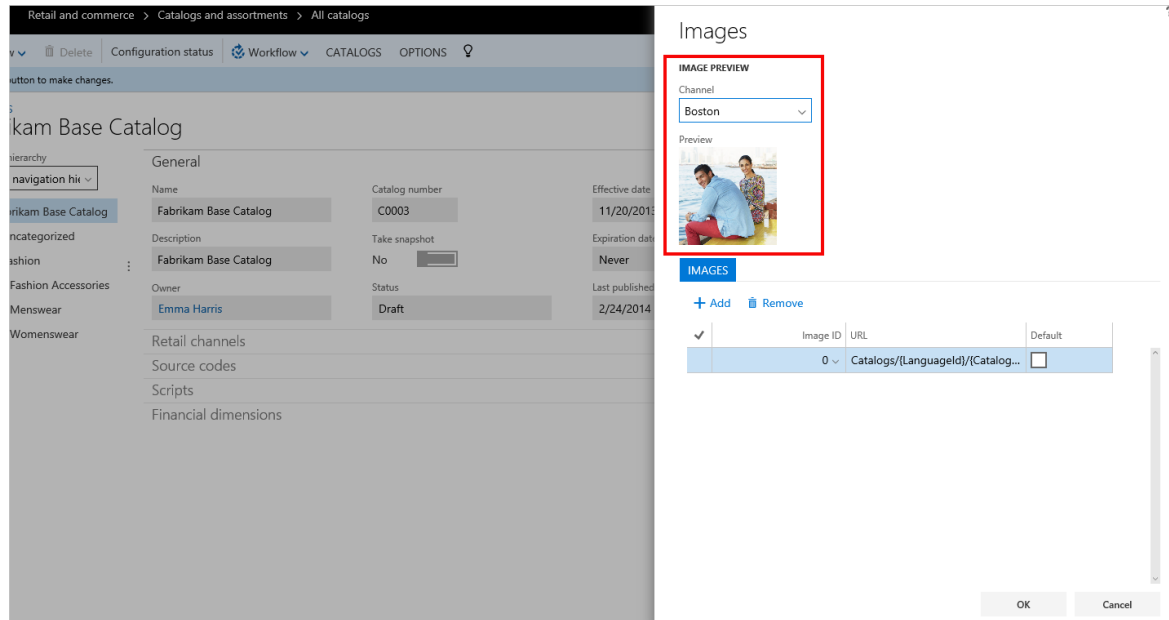


## Previewing an image from the entity level

1. From the page for the entity item in HQ, you can preview the image that uses the image URL that is derived from the media template. For this example, go to the appropriate catalog, and then, on the Action Pane, click **Media > Images**. Use the drop-down list to select different stores that might have different channel profiles.
2. To edit or remove the implicit media template, you must return to the **Define media template** dialog

box for the **Catalog images** page.

3. You can use the **Add** and **Remove** buttons to manually change the path that is based on the implicit template and used for a specific image. For more information, see the [Overwriting the media template for entity items](#) section later in this article.
4. After you've finished previewing an image and making any changes that you require, start the MPOS instance for the appropriate store, and see whether the catalog images are shown.



#### NOTE

You can use the same procedure for all the five entities that are supported: Worker, Customer, Catalog, Category, and Products. "Catalog Products" (products that are set at the catalog level) and "Channel Products" (products that are set at the channel level) use the media template that is set for the Products entity. For the Products media template, you can select the number of product images to show per product. You can also set the default image for a given product. In this way, you can prevent blank images in MPOS and help to control which image is used as the default image for a product item. In the following example, each product has five images, and the first image is set as the default image. Variant products are treated the same way as master products. The file name of the image file should be based on the product number. Some characters are also escaped while the file name is generated. Therefore, it's a good to verify the file name by using the [Generate Image URLs for Excel](#) section. See the [Overwrite by using Edit in Excel](#) section later in this article.

## Synchronization jobs to send a media template to the channel side

For all the five supported entities (Worker, Customer, Catalog, Category, and Products), whenever you update the **Define media template** dialog to set up an image, make sure that you run the Catalog job (1150) from **Retail and Commerce IT > Distribution schedule**. This job will enable the updated media template to be synced to the channel and used by MPOS. Run the Catalog job (1150) after you make any of the following changes:

- You update the Catalog image media template from **Catalog images > Define media template**.
- You update the Employee image media template from **Employee images > Define media template**.
- You update the Customer image media template from **Customer image > Define media template**.
- You update the Product image media template from **Product images > Define media template**.
- You update the Category image media template from **Category images > Define media template**. You must also publish the channel.

# Overwriting the media template for entity items

As you learned in the previous section, the media template for a given entity supports only one common path. This path is based on the media base URL that is configured and the media path that is defined. However, in many cases, a retailer wants to be able to use images from different sources for a subset of items in an entity. For example, a store uses the self-hosted media server for one set of catalog images but uses CDN URLs for another set. To overwrite image URLs that are based on a media template for entity images at the entity level, you can use the Edit in Excel and Manual edit functionality from the **Preview** page.

## Overwrite by using Edit in Excel

1. Click **Retail and Commerce > Catalog management > Catalog images**.
2. On the **Catalog images** page, click **Define media template**. In the **Define media template** dialog box, in the **Entity** field, **Catalog** should be selected.
3. On the **Media path** FastTab, notice the image location.
4. On the **Generate Image URLs for Excel** FastTab, click **Generate**.

### IMPORTANT

Whenever the media template is changed, you must click **Generate** before you can use the Edit in Excel functionality.

You now see a preview of the image URLs that were generated based on the last saved media template.

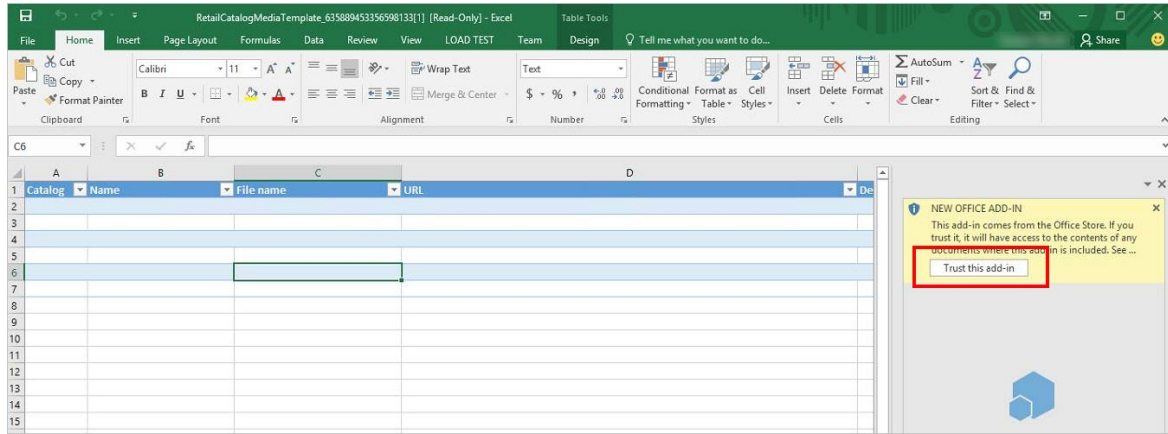
Name	File name	URL
Contoso Catalog	Contoso Catalog.jpg	Catalogs/(LanguageId)/Contoso C
Fashion Catalog	Fashion Catalog.jpg	Catalogs/(LanguageId)/Fashion ...
Fabrikam Base Catalog	Fabrikam Base Catalog.jpg	Catalogs/(LanguageId)/Fabrika...
Fabrikam Semi-Annual Sale	Fabrikam Semi-Annual Sale.jpg	Catalogs/(LanguageId)/Fabrika...
Contoso Appliance Catalog	Contoso Appliance Catalog.jpg	Catalogs/(LanguageId)/Contoso...
Contoso Base Catalog	Contoso Base Catalog.jpg	Catalogs/(LanguageId)/Contoso...

### NOTE

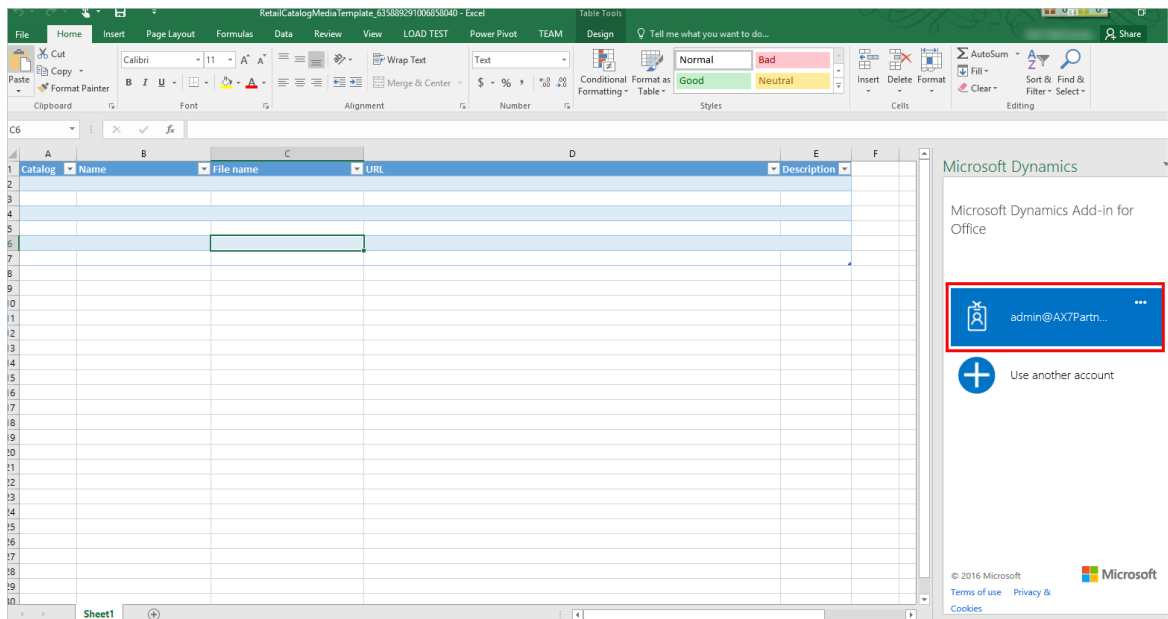
The URLs that are generated for Excel use the path and conventions of the media template that is defined. These conventions include the conventions for file names. The expectation is that you've set up the physical images outside Commerce, and the images can be retrieved from the URLs that are derived from the media template that you defined earlier. You can overwrite these derived URLs by using the Edit in Excel functionality.

5. Click **Edit in Excel**.
6. After the Microsoft Excel worksheet is opened, click **Enable edit** when you're prompted.
7. When you're prompted, click **Trust this add-in** in the right pane, and wait for the add-in to complete the

installation.



8. If you're prompted to sign in, enter the credentials that you used to sign in to HQ.



9. After you sign in, you should be able to see the list of image URLs for the various catalog entries.

10. You edit, add, and remove the image URLs for various entity items.

11. For all entities except Products, you can overwrite the image URLs. Modify the existing image URL, so that it uses the new destination URL of the image, and update the file name with the new file name for the image file. The file name must be unique to help guarantee that the record is unique.

Catalog	Name	File name	URL	Description
5637144576	Contoso Catalog	Contoso Catalog.jpg	www.microsoft.com/azure/machineLearning/POC.jpg	
5637145334	Fashion Catalog	Fashion Catalog.jpg	Catalogs/(Languageid)/Fashion Catalog.jpg	
5637145335	Fabrikam Base Catalog	Fabrikam Base Catalog.jpg	Catalogs/(Languageid)/Fabrikam Base Catalog.jpg	
5637145336	Fabrikam Semi-Annual Sale	Fabrikam Semi-Annual Sale.jpg	Catalogs/(Languageid)/Fabrikam Semi-Annual Sale.jpg	
5637145337	Contoso Appliance Catalog	Contoso Appliance Catalog.jpg	Catalogs/(Languageid)/Contoso Appliance Catalog.jpg	
5637145338	Contoso Base Catalog	Contoso Base Catalog.jpg	Catalogs/(Languageid)/Contoso Base Catalog.jpg	

#### NOTE

When you overwrite image URLs for Products entities by using the Edit in Excel functionality or the entity item page, MPOS always shows all the media template image URLs together with the overwritten image URLs.

- After you've finished making your changes, click **Publish in Excel** to create a new explicit association entry.
- Return to HQ, and click **OK**.
- Run the appropriate synchronization jobs for the entity, and check the preview on the entity page or in MPOS.

#### Creating new records

You can create new records in Excel. However, make sure that you provide the correct information. For example, to create a new entry for a catalog, make sure that the catalog ID and catalog name are correct, and also provide a unique file name. The unique file name is very important, because the uniqueness of records in Excel is validated during publishing. First copy the details from the catalog that you want to create a new record for, and copy the record. You just have to update the file name and URL, because the rest of the information will be same. To create new records for Product entity items, you use the same basic procedure. From the Excel worksheet, copy an existing record for the product that you to create a new record for, and then replace the image URL and filename. Make sure that the file name is unique.

#### Deleting an existing record

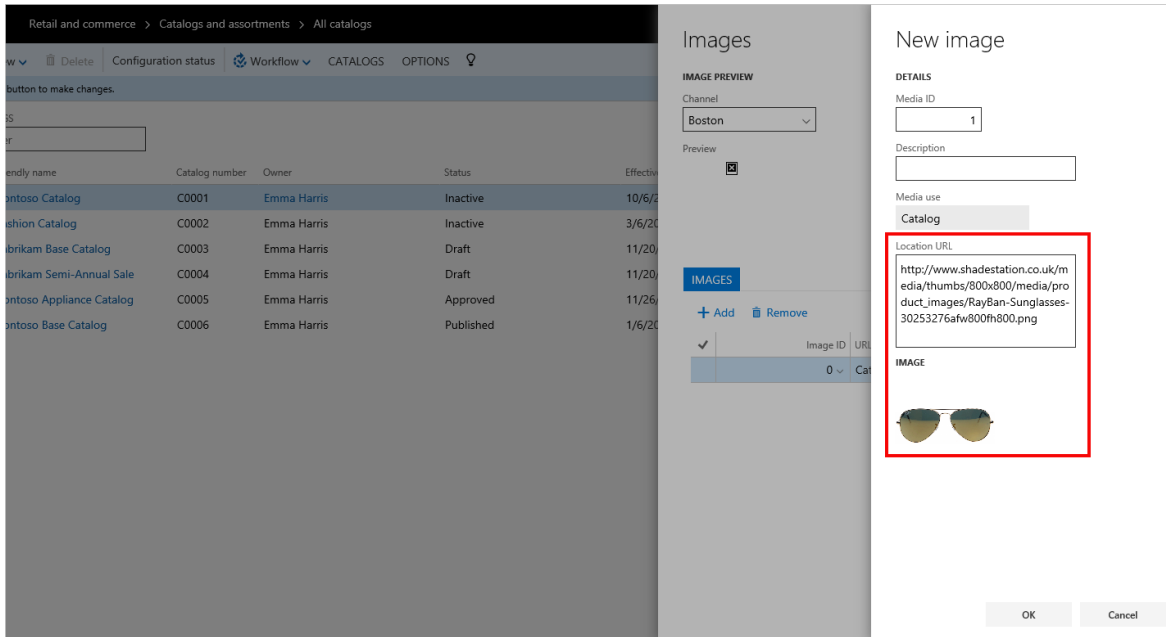
Only the overwritten image URL records can be deleted. After an image is deleted and synchronization is completed, the image will no longer appear on the **Preview** page or in MPOS. Image URL records that are derived from the media template can't be deleted, because these records are always derived from the media template every time.

#### Overwrite from the entity-level Preview page

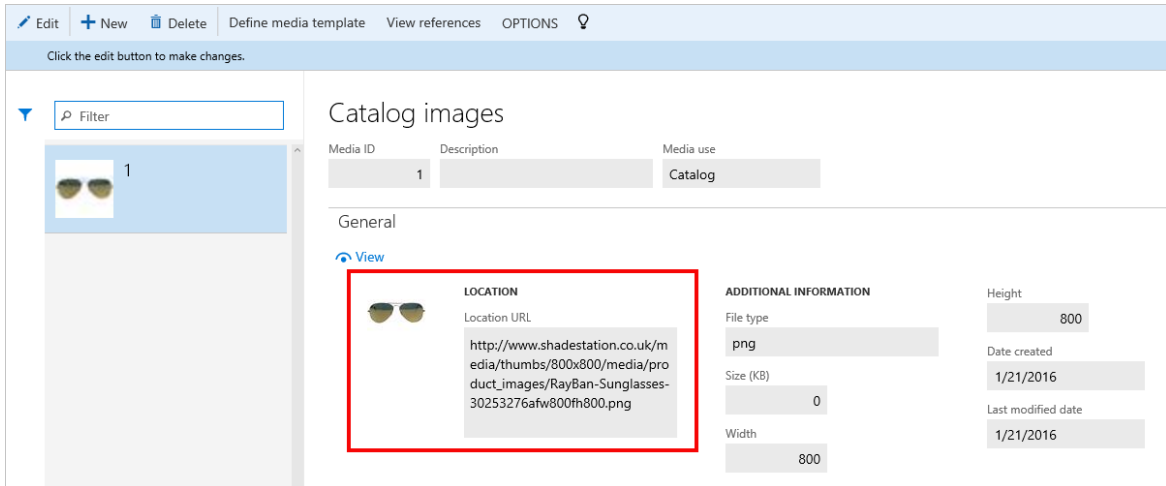
For all entities except Products, you can overwrite the image URL for a given entity item at the entity item level from the **Preview** page. For Products, you can use the "Catalog Products" entity page. This example shows how to overwrite a catalog image.

- Click **Catalogs > Media > Images**, and select the catalog image to update.
- Click **Add**, and enter the image URL to overwrite the media template URL.
- If you want this image to be shown in MPOS for the catalog, you can set it as the default image.

4. Click **OK**. The image URL is updated for this catalog image, and a preview is shown.



5. You can also see the image preview for all overwritten image URLs on the **Catalog images** gallery page.



#### NOTE

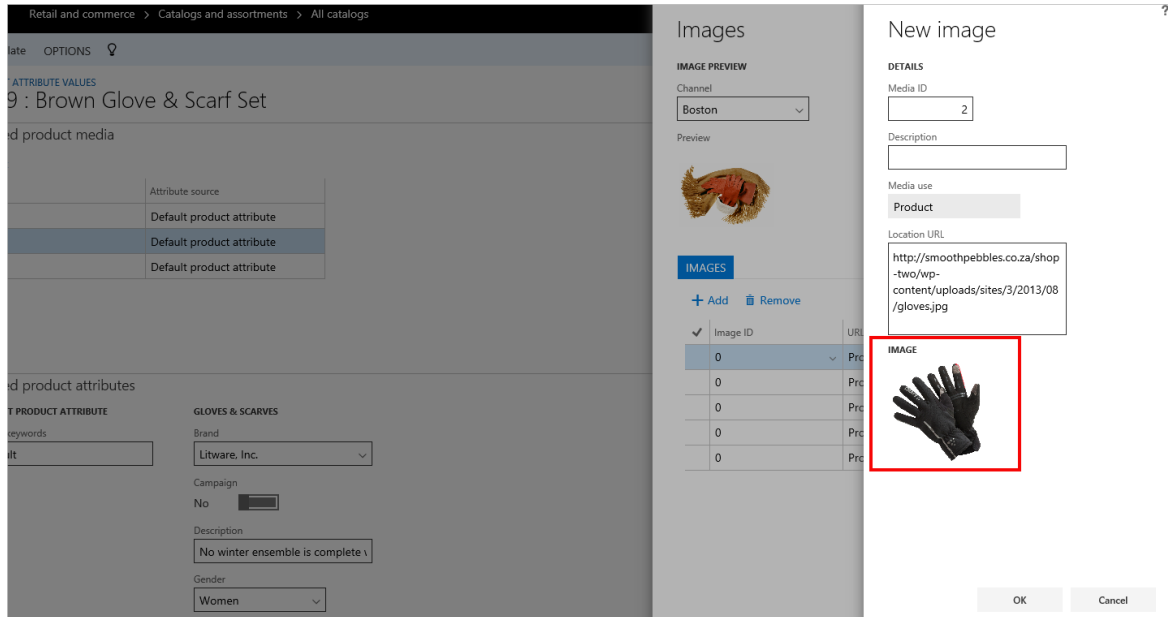
Currently, the gallery doesn't show image previews for media template image URLs. For Catalog, Worker, Customer, and Category entities, if the user explicitly provides a URL through this page, we recommend that you indicate which image is the default image, because Commerce Scale Unit clients show only one image per Catalog, Customer, Worker, and Category. If the user doesn't specify a default image, the system determines the default image and send it to the Commerce service caller (MPOS or Ecommerce).

### Overwrite the image URL for catalog product images from the Preview page

To overwrite image URLs for catalog product images, you must use the **Preview** page. You can't use the Edit in Excel functionality.

1. To overwrite product images at a catalog level, select a catalog, and then select the product to overwrite the image for.
2. Click **Attributes**.
3. On the next page, select **Image**, and then click **Edit**. The **Preview** page opens as a slider dialog box.
4. Click **Add**, and overwrite the image URL with a new URL.

5. Click **OK**. You now see the preview of the new image and can set it as the default image.



#### NOTE

After category image association, you must publish the channel and run the Channel job to help guarantee that the changes are published to the channel database.

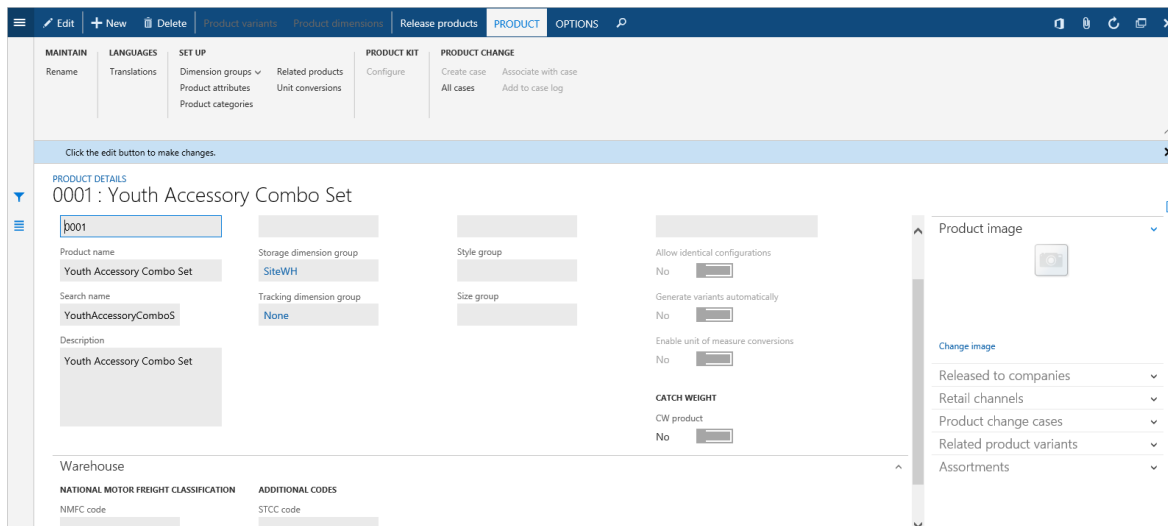
## Setting up images to appear in Offline mode for MPOS

MPOS can run in Online mode (when MPOS connected to Commerce Scale Unit) or Offline mode (when there is no Commerce Scale Unit or network connectivity, and transactions are stored in a local offline database). When MPOS runs in Offline mode, it can't get images from the external image server to display from Commerce Scale Unit, because connectivity has been lost. However, you can still set up images so that they are shown when MPOS runs in Offline mode.

### Set up product images to appear in Offline mode for MPOS

The product images that must be used in Offline mode can be set up by uploading the required physical image into the base product image.

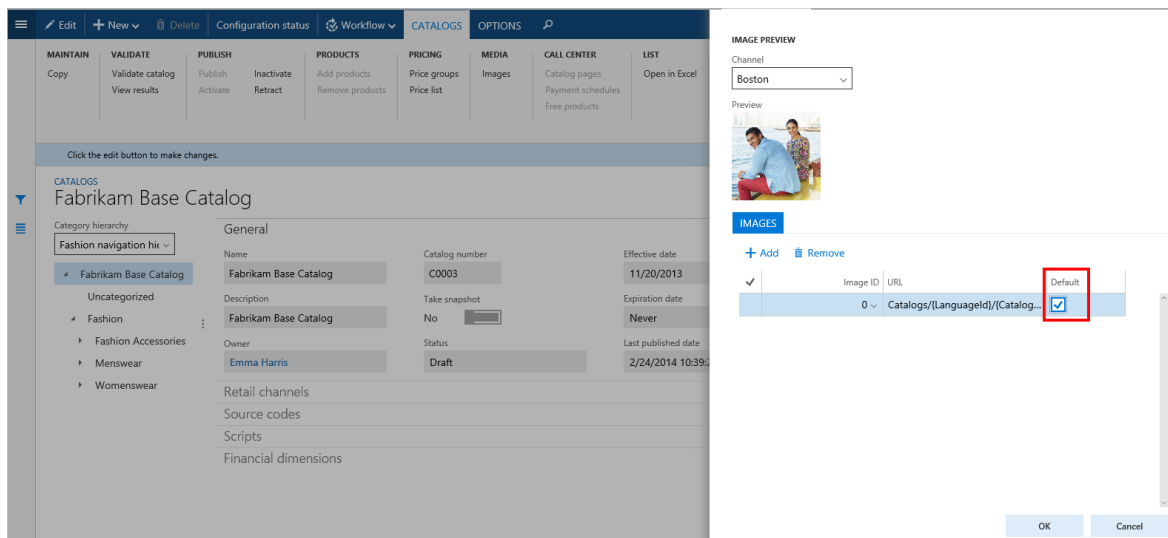
1. Click **Product information management > Products > Products**.
2. Select the product to set the offline image for.
3. Click **Edit**, and then click the arrow in the right corner to show the right pane.
4. On the **Product image** FastTab, click **Change image**, and upload the physical image to use for the selected product in Offline mode.
5. Save and close the page.
6. While MPOS is in Online mode, run the Catalog job in HQ, to make sure that the data is sent at least one time to the offline database.
7. Put MPOS into Offline mode. You should see the image that you uploaded for the specific product in HQ.



## Set up catalog, category, employee, and customer images to appear in Offline mode for MPOS

The catalog, category, employee, and customer images that must be used in Offline mode can be set up by adding the required image's destination link to the gallery and setting the image as the default image for the selected entity.

1. Go to the catalog, and then, on the Action Pane, click **Media > Images**.
2. Follow the steps in the [Overwrite from the entity-level Preview page](#) section to add the external image URL.
3. Mark this image as the default image for the catalog by selecting the check box against the Image listed in the grid.
4. Run the Catalog job. This image will now be used as the Offline image for that catalog in MPOS.
5. Follow a similar process for other entities, such as Category, Employee, and Customer.



### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Client images in POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic is intended for people who implement functionality related to point of sale (POS) client image management in a retail environment. It provides tips and guidance to consider when planning an implementation.

This guidance applies to both Cloud POS and Modern POS, and provides some general information about image file size handling and types of images that can be used to enrich the user experience with the store and support customer-focused scenarios like up-selling, cross-selling, and clienttelling. Welcome screen images, category images, and product images are examples of types of images that you can use.

## Implementation considerations

- **File size** - In order to maintain responsiveness of the POS client user interface (UI) and performance while loading different kinds of images, we recommend that you prepare your images to be an appropriate file size respective to the purpose of use. For example, product and category images in the Contoso demo data are sized at 500 x 500 or 580 x 580 pixels.
- **Image size** - The screens and displays of the POS devices will pre-determine reasonable image sizes (length and width). You should size the image as close to your intended screen size as possible. For an example, see the "Implementation example" section below.
- **Resolution** - An important parameter to consider is the dots per inch (dpi), or for screen resolution, pixel per inch (ppi). Because POS client images will not be printed, the common dpi setting for rendering images on the web is a good guideline (72 to 150 dpi). Contoso demo image files are typically rendered to 96 dpi. For high resolution devices, you should take operating system (OS) scaling into account and use the effective resolution, rather than the actual pixels.
- **File types** – You can use \*.png or \*.jpg image file types. In most cases, \*.jpg are smaller in size.

## Implementation example

To create a welcome screen image that covers two-thirds of the canvas on a common 18.5 inch POS display with a 1366 x 768 pixel resolution and a screen layout similar to Contoso demo data, choose an image with resolution that is not much higher than the length and width of the screen. In this example, 911 x 512 pixel resolution is sufficient. Selecting a relatively high resolution for length and width but keeping a low dpi setting will still result in reasonably small file sizes.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Track commissions in the point of sale (POS) by using sales groups

2/18/2021 • 4 minutes to read • [Edit Online](#)

It's a common retail practice to track sales by the associate who worked with the customer by—providing assistance, up-selling, cross-selling, and processing the transaction.

Tracking sales by sales representative is a measure of the associates selling abilities, while sales by cashier is a measure of speed and efficiency. Sales tracked by sales representative are also often used to calculate commissions or other incentives.

## Configuring a worker to be a sales representative in POS

When a worker is added to a sales group, they become eligible for commission and can be identified as a sales representative in the system. A worker who isn't in a sales group isn't eligible for commission and won't be listed as a sales representative in the point of sale (POS) application. In POS, the list of sales representatives is derived from all sales groups that contain at least one worker assigned to the store. The list is shown in POS as a combination of Sales group ID and Name (ID : Name). A default sales group can be assigned to workers to support scenarios where the retailer chooses to set the sales representative on POS lines automatically. Users can select from any sales group that the worker is a member of.

## Functionality profile settings

There are a number of functionality profile settings for a store that will determine the flow and process in POS that involve sales representatives.

PROFILE	DESCRIPTION
Default to cashier when available	If this option is enabled, POS will automatically populate transaction lines with the current cashier's default sales group. If a cashier doesn't have a default sales group specified, the value won't be set. A user could still manually set the sales group by using a POS button grid button.

PROFILE	DESCRIPTION
Prompt for sales representative	<p>This option has three possible values:</p> <ul style="list-style-type: none"> <li>• <b>No</b> – If this option is selected, the user won't be prompted to select a sales group. The value could still be set by using a cashier's default Sales group or manually by using a POS button grid button.</li> <li>• <b>Start of transaction</b> – If this option is selected, and either the <b>Default to cashier</b> option isn't enabled or the current cashier doesn't have a default sales group, the user will be prompted to select a sales group at the beginning of each transaction. Selecting a sales group from this prompt will default all subsequent lines to the selected sales group. A user could still manually set the sales group by using a POS button grid button.</li> <li>• <b>For each line</b> – If this option is selected, and either the <b>Default to cashier</b> option isn't enabled or the current cashier doesn't have a default sales group, the user will be prompted to select a sales group after adding each line. A user could still manually set the Sales group by using a POS button grid button.</li> </ul>
Require	<p>This option is only applicable when POS is configured to prompt for a sales representative. If enabled, the user will be required to choose a sales group before continuing. Otherwise, the user will be prompted, but can cancel and continue without making a selection. After the line is added, a user with sufficient permissions could still remove the sales group from the line. "Require sales representative" is not enforced in this situation.</p>

## Displaying the Sales representative information on the POS transactions screen

The POS transaction screen layout and contents are configurable using the screen layout designer and assigned screen layouts to stores, registers, or workers. The **Sales representative** field can be added to the Lines tab of the Receipt pane. This will display the ID of the specified Sales group for each line on the transaction screen.

## Adding Sales representative operations to POS button grids

POS allows users to configure button grids, which are included in screen layouts to provide access to POS operations. The following POS operations can be assigned to button grid buttons that pertain to Sales representatives.

OPERATION	DESCRIPTION
Set sales representative on line	This POS operation displays a list of eligible Sales groups (ID : Name) for the store. Selecting a Sales group from this list will set the value on the current transaction line.
Clear sales representative on line	This POS operation removes the current Sales group value from the current transaction line.

OPERATION	DESCRIPTION
Set sales representative on transaction	This POS operation displays a list of eligible Sales groups (ID : Name) for the store. Selecting a Sales group from this list will set the default value on the current transaction. Any existing lines without a sales group assigned will be set, as well as any subsequently added lines.
Clear sales representative on transaction	This POS operation removes the current default Sales group value from the current transaction. It does not impact any lines already existing in the transaction.

## Calculating commissions

Commission is calculated for the workers in the specified sales groups at the time of statement posting or sales order posting. The commission amount is determined based on the worker's commission share, as defined in the sales group and the associated commission calculation settings for the customer and/or products on the transaction.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure, install, and activate Modern POS (MPOS)

2/18/2021 • 21 minutes to read • [Edit Online](#)

This topic describes how to configure, download, and install Modern POS on various platforms. It then describes how to activate Modern POS through device activation.

## NOTE

There are two Modern POS installers: Modern POS and Modern POS with offline (this installer also installs the offline database).

Starting in release 10.0.11, altering customized files that are stored in the ClientBroker folder could cause issues when installing a newer release. These issues might include the inability to go offline or a newer installer failing to complete successfully. A workaround is to remove the files in the ClientBroker folder in the Modern POS directory before performing the installation using the newer installer.

Starting in 10.0.15 release, customizations to files in the Client broker folder for Modern POS can cause an error when updating from a previous version. The known workaround is to delete all files from the Client broker folder prior to running the newer Modern POS installer. For automation, this can easily be scripted as a pre-step for the installer. All files in this folder must be deleted. When this error occurs, the newer installer will update the current installation correctly.

## Technology

The self-service process lets you download the appropriate version of the Modern POS installer and install it on the physical device that you want to use as the point of sale (POS) register. Device activation is the main onboarding step that ties the physical device to a register in Headquarters. Here are the main technical functions of this feature:

- Tie a physical device to a business entity (register).
- Provide enhanced security through Microsoft Azure Active Directory (Azure AD) and a device token/ID.
- Stop unauthorized remote use of Modern POS. (In other words, deactivate a device remotely.)
- Initialize settings for easy Modern POS functioning (number sequence, hardware profile, merchant information) as the first touchpoint of the POS.
- Comply with payment card industry (PCI) standards, and report on device information from Headquarters.

## NOTE

If you are installing Modern POS for use with an on-premises environment, Modern POS does not use Azure Active Directory credentials for device activation.

## Setup

Before you start the steps that are outlined in this topic, follow these steps.

- Verify that you have credentials to sign in to Headquarters.
- Verify that you have administrative or root access to install Modern POS on a device.
- Verify that you can access the Commerce Scale Unit from the device.
- Verify that the environment contains the Commerce permission groups and jobs in the **Human resources**

module. These permission groups and jobs should have been installed as part of the demo data.

## Download and install Modern POS

### Verify that the device is correctly configured

1. In Headquarters, go to **Retail and Commerce > Channels > Channel deployment**.
2. On the **Channel deployment** page, select the **Registers** tile.
3. On the **Registers** page, select a store register. The demo data thoroughly defines the Houston store and registers for self-service. To find the Houston registers, enter **Houston** in the filter at the top of the list of devices.
4. Select a register by selecting the register number in the **Register number** column. In the Houston store, register Houston-3 is well defined and is therefore useful as an example.
5. On the page for the register, under **General**, verify that the **Support offline** option is set to **No**. To use offline support, on the Action Pane, select **Edit**, and then set **Support offline** option to **Yes**.

### Download the Modern POS installer

1. On the **Welcome** page, use the menu in the upper left to go to **Retail and Commerce > Channels > Channel deployment**.
2. On the **Channel deployment** page, select the **Devices** tile.
3. Select a device.

#### NOTE

- The Houston devices are well defined. Houston-3 is useful as an example for a Microsoft Windows desktop or tablet. Houston-21 is useful as an example for a Windows Phone.
- When you select a device, the **Download** button on the Action Pane becomes available.

4. Select **Download**, and then select **Configuration file**. Note the following:
  - Browsers might block the download pop-up that is generated. You must select either **Allow once** or **Options for this site > Always allow**. Then, while the device is still selected, select **Download** again.
  - The configuration file must be saved to the same location as the Modern POS installer. For security reasons, delete this file after installation is completed. If the configuration file is not the same file name as the installer executable, either the executable must be run using the command line to specify the configuration file or you need to rename the XML configuration file to have the same base name as the executable file name.
5. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)
6. Select **Download**, and then select **Retail Modern POS**. Note the following:
  - Browsers might block the download pop-up that is generated. You must select either **Allow once** or **Options for this site > Always allow**. Then, while the device is still selected, select **Download** again.
  - The installation package that you must use varies, depending on whether you require offline support, and whether the device that Modern POS will be installed on is a Windows tablet or a phone device (such as a Windows Phone, an Android device, or an iOS device). The correct package is automatically selected for download, based on the register settings and the application type that is set for the device. If the offline package is selected for a Windows tablet, but Microsoft SQL Server isn't already installed

(or if it doesn't meet the requirements for the offline package), SQL Server is downloaded and installed silently.

7. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)
8. After the setup installer has been saved, on the Notification bar, select **Run**. (This step might differ, depending on your browser.)

### Before running the Modern POS installer

- Make sure that all [system requirements](#) are met.
- It is recommended to temporarily turn off antivirus applications. It has been noted that on aggressive antivirus solutions, the installation may stall due to the antivirus solution checking active files while in use.
- The installer will sideload a modern application. Therefore, a Group Policy entry must be set to allow for sideloaded applications. The installer will change the associated registry key as follows to allow for this installation:
  - **Path:** HKLM:SoftwarePoliciesMicrosoftWindowsAppx
  - **Property:** AllowAllTrustedApps
  - **Value:** 1
- If offline is used (an offline database created), then a default SQL Server instance must exist. If SQL Server instances exist, but none are set as the default, then the installer will fail to install the offline database.

If you are installing Modern POS for use with an on-premises environment, you must start the installer from a command line as follows:

```
ModernPosSetupOffline.exe -UseAdfsAuthentication
```

### Run the Modern POS installer on a Windows computer

The Modern POS installer first extracts the associated files and then starts the installation.

1. The installer validates that all prerequisites are met. Note the following:
  - If a system restart is required, the installer informs you about this requirement, but the installation can typically continue.
  - A sideloaded installation of Modern POS requires a Group Policy change. The installer informs you if this change is required and then makes the change automatically.
2. If you selected offline support, but a valid version of SQL Server isn't found, the installer downloads and installs Microsoft SQL Server 2014 Express with Service Pack 2 (SP2). To meet the prerequisites, SQL Server must have Full-text search installed. Additionally, a minimum of SP2 must be installed for Microsoft SQL Server 2014, or a minimum of Service Pack 3 (SP3) must be installed for Microsoft SQL Server 2012. Note the following:
  - The installer tries to download the correct language. However, if you require a specific language, we highly recommend that you manually install SQL Server. If the installer can't correctly determine the language, it installs the English version of SQL Server 2014 Express with SP2 by default. Typically, after the SQL installation is completed, the system requires a restart before the installation of Modern POS can continue.
  - This process might require a long time, depending on the speed of the computer and the Internet connection. If a prerequisite fails during this step, first retry the installer. If the installer continues to fail, see the [Troubleshooting](#) section of this topic.
3. The installer installs Modern POS.
4. On the page that states that installation was successful, select **Close** to exit the installer.

You can now start the program.

#### NOTE

This installation occurs only for the administrator user who ran the installer. For all other users, a desktop icon to install Modern POS is created. Every time that a user signs in, he or she must double-click this icon. The program will then be installed or updated, as required. If a user doesn't use the desktop icon after an update, the POS client will request that the user run from the desktop icon instead to update correctly prior to running.

#### Run the installer on any other device (Windows Phone, Google Android device, or Apple iOS device)

1. If the application wasn't downloaded directly to the device, transfer the downloaded app file and the associated configuration file to the same folder on the device. Depending on the type of device, the app file will be an APPX, APK, or IPA file.

Note that this step can be done in various ways. For example, the files can be accessed through a shared folder, transferred via USB cable, or securely mailed to the user's device.

2. Use a file explorer on the device to browse to the app directory.
3. Tap the app to begin application installation. (If the configuration file was saved to the same location, the Commerce Scale Unit URL will be automatically entered when you start the application and begin device activation.)

Note that some devices require that you double-tap the file to begin application installation. Some devices might not notify you that an application has been installed. On those devices, we recommend that you look at the application list to verify that the application was correctly installed.

4. When the installation is completed, you should be able to start the application from the application list on the device. For example, after you install the application on a Windows Phone, you can start it from the home screen tiles list.

You can now start the program.

## Create a worker

For this topic, we have already created workers and assigned them to the Houston address book in the demo data that is provided. Therefore, this topic will use pre-generated data.

#### Create a worker

1. Go to **Retail and Commerce > Employees > Workers**.
2. On the Action Pane, select **New** to create a new employee.
3. Enter the first and last name. For example, enter **John** as the first name and **Smith** as the last name.
4. Verify that the **Legal entity** field is set to **USRT**, the **Worker type** field is set to **Employee**, and the **Employment start date** field is set to the current date at 12 AM, so that the worker's employment starts immediately.
5. Select the **Assign a position** check box. Select position number **000544**, which is the Store manager position.
6. Set the **Personnel action type** field to **Hire Action** to hire a new employee immediately.
7. Select **Continue**.
8. On the Action Pane, select **Complete** to finish creating the new worker.
9. Return to the worker list. Search for the newly created worker (for example, John Smith). Select the worker's name to see the details of the new worker.
10. On the Action Pane, select **Edit**.
11. Verify that the language for the worker is **en-us**.



12. Under **Worker summary**, in the **Address books** field, select the **Houston** store.
13. On the **Commerce** tab, you can reset the POS password. For this tutorial, reset the password to **123**.
14. On the **Commerce** tab, under **Screen layout**, assign a screen layout. For example, select **F2MP16:9M (Fabrikam MPOS Manager (16:9))**.
15. On the Action Pane, select **Save**.
16. Go to **Retail and Commerce > Periodic > Distribution schedule**.
17. Select the **1060 – Staff job**, and then, on the Action Pane, select **Run now** to sync the worker data to the channel database.
18. After the new worker has been created and synced to stores, worker John Smith can sign in to any POS device that is used in the HOUSTON store that he is assigned to, and he can perform transactions on that device. However, the device must be activated first. The following section explains how to activate a device for a new worker.

### **Map an Azure AD account to a worker who has POS permissions for device activation**

You must complete this procedure before you activate Modern POS for a new worker.

1. In Commerce, from the **Worker** page, open the **Worker details** page for the worker that you created in the previous procedure.
2. On the Action Pane, select **Edit**.
3. On the **Commerce** tab, select the **POS permissions** link. Under **POS permission group**, verify that the value is **Manager**.
4. When you've finished, return to the **Worker details** page for the new worker.

To return to the **Worker details** page, select the **Close** button (X) on the right side of the Action Pane.

5. On the Action Pane, select **Commerce**, and then select **Associate existing identity**.
6. In the dialog box that appears, select the Azure AD account that is named **admin AX Admin**. (If an alternative administrator Azure AD account has been created, select that account instead.)
7. Select **OK**. In the demo data, the Azure AD account that is associated with the administrator account in Headquarters is your administrator Azure AD account.
8. On the Action Pane, select **Save**, and then refresh the page. The **External identity** section should be now updated with the new information.

Note that the **External identifier** field will remain empty. This behavior is expected. Therefore, you can ignore it.

This procedure should be completed before you activate Retail Cloud POS or Modern POS. For more information, see [Manage activation accounts and validate devices](#).

### **Run the Validate Devices for Activation check**

1. In Headquarters, open the **Device** page (**Retail and Commerce > Setup POS > Devices**).
2. Select the device to validate for device activation, and then select **Validate Devices for Activation**. For example, select device **HOUSTON-3**.
3. In the dialog box that appears, select the worker to validate the device for (that is, the worker that you mapped to the Azure AD account in the previous procedure). For example, select worker **000160**.
4. Select **OK**, and make sure that you receive the following message: "Pre-Activation validation completed for Device HOUSTON-3 and Staff 000160. Validation: Passed"

## **Activate a device**

## NOTE

It is possible for the Safari browser to show an error during device activation of a Cloud POS device due to an Azure Active Directory token being unattainable. You can resolve this issue by utilizing the [Microsoft Enterprise SSO plug-in for Apple devices](#).

1. Start Modern POS on your computer. Read the instructions on the **Before you start** page, and make sure that they are completed. Then select **Next**.
2. Select **Activate**. You're redirected to the Azure AD sign-in page.
3. Enter the Azure AD account that you mapped earlier, such as `admin@<MyCompany>.onmicrosoft.com`, and the password.
4. When activation is completed, select **Get Started**.
5. Sign in to Modern POS by using worker account **000160** and the password **123**.

The device should now be activated and ready to use.

## Update the Modern POS application

### NOTE

To learn more about deployable packages, see [Apply a deployable package](#).

1. After a Modern POS application is uploaded into the environment, the version of the package can be selected on the device. The package listings should include the new uploaded application.
2. To update the Modern POS application, follow the steps in the [Download and install Modern POS](#) section. To do an in-place update, just run the newer version of the self-service installer. Uninstallation isn't required or recommended. Device activation status will be maintained after the update.
3. The installer will use the currently installed configuration settings. If the configuration file has changed, because of various configuration changes in Commerce, an update won't change the Modern POS application settings.

## Troubleshooting

### Troubleshoot installation

- Your browser blocks the download pop-up that is generated.

**Solution:** Select either **Allow once** or **Options for this site > Always allow** (or the equivalent commands in the browser that you're using). Then, while the correct register is still selected, select **Download** again.

- The installation package that you must use depends on whether you require offline support. The correct package is automatically selected for download. For the offline package, SQL Server must be installed and must meet the requirements for the offline package.

**Solution:** No action is required. If SQL Server isn't already installed (or if it doesn't meet the requirements), it's downloaded and installed. The installer gives generic information about the download and installation of SQL Server Express 2014. This installation might require a long time.

- The installation occurs only for the administrator user who ran the installer, but not for any other users.

**Solution:** The installer generates a desktop icon that is used to install, upgrade, and run Modern POS. This icon is generated for every user on the computer. When a user who must install Modern POS double-clicks this icon, the program is installed. The user can then start to use Modern POS.

- SQL Server isn't successfully downloaded and installed through the self-service Modern POS installer.
  - **Solution 1:** A list of reasons shows the prerequisites that failed. If the list includes **SMO** or **SQL Management Objects**, first try to run the installer again. SQL Server Management Objects (SMO) are installed during SQL Server installation. Therefore, it's possible that the operating system didn't pick up the registration of the executable program that you used. When you run the installer a second time, the prerequisites are retested, and the prerequisite check should correctly verify the required executable program. If the installer continues to fail, restart the system to fully complete the registration of SQL Server, and then rerun the installer.
  - **Solution 2:** Manually download and install SQL Server (Microsoft SQL Server Express or another version) by using Advanced Tools. During installation, select **Full-text search** as an additional feature.
- The installation of Modern POS fails, because the registration of performance (perf) counters failed.

**Solution:** Follow these steps to fix this issue:

1. Open a **Command Prompt** window as an administrator.
2. Enter the following command.

```
lodctr /s:"perf_backup.txt"
```

3. Enter the following command.

```
lodctr /R
```

4. If the system doesn't rebuild the performance counter settings from the system backup, rerun the **lodctr /R** command.
  5. Rerun the Modern POS installer.
- If you're using a downloaded virtual hard disk (VHD) instead of a cloud-hosted environment, the downloader might fail.
    - **Solution 1:** In a downloaded VHD, the Azure Storage Emulator must be installed and must be running correctly. Otherwise, the self-service packages can't be downloaded correctly.
    - **Solution 2:** A failure might have occurred during the process of integrating the VHD into Microsoft Hyper-V. You must manually edit permissions before the packages can be downloaded correctly. Follow these steps:
      1. In File Explorer, browse to **C:\Microsoft Dynamics 365\70\Retail Server**.
      2. Right-click the **SelfServicePackages** folder, and then select **Properties**.
      3. On the **Security** tab, select **Edit**.
      4. In the **Permissions for SelfServiceDeployment** dialog box, select **Add**.
      5. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, select **Locations**.
      6. In the **Locations** dialog box, select the first entry in the list (the local computer), and then select **OK**.
      7. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, enter the name **IIS\_IUSRS**, and then select **Check names**. The object name should be changed to **IIS\_IUSRS**. Select **OK**.
      8. In the **Permissions for SelfServiceDeployment** dialog box, select the new **IIS\_IUSRS** user. Under **Permissions for IIS\_IUSRS**, select **Allow** for the **Full control** permission. Select **OK**.
      9. In the **Open permission** dialog box, select **OK**.

- The latest iOS version does not support your self-signed certificate.

**Solution 1:** Utilize a domain and generate a proper domain-based certificate.

**Solution 2:** Download the open source OpenSSL library and perform the following after completing installation:

1. Using PowerShell, create a private key for the root Certificate Authority (CA) using a command such as `$ openssl genrsa -des3 -out rootCA.key 2048`. 2. You will be prompted for a password, which must be remembered for later usage. 3. Next, generate the root certificate using a command such as `$ openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem`. There will be a prompt for the password entered previously and some basic certificate information.

#### NOTE

The number of days the certificate is valid for can be altered. In the above example this is 1024 days.

- d. Create a new **info.ext** file and enter the following details:
  - o `keyUsage = keyEncipherment, dataEncipherment - extendedKeyUsage = 1.3.6.1.5.5.7.3.1 - subjectAltName = @alt_names - [alt_names] - DNS.1 = <FULLY QUALIFIED DOMAIN NAME OF HOST COMPUTER>`
- e. Generate the signing request and private key using a command such as `openssl req -new -nodes -out server.csr -newkey rsa:2048 -keyout server.key`.
- f. Issue the certificate using the previously generated root certificate using a command such as `$ openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out server.crt -days 500 -sha256 -extfile info.ext`. There will be another prompt for the root key password and you will need to specify the number of days that the certificate is valid (500 days in this example).
- g. Generate the IIS certificate using a command such as `$ openssl pkcs12 -inkey server.key -in server.crt -export -out server.pfx`. This command will request a new password, which will be used later when the certificate is imported.
- h. Open **Certmgr.msc** and go to **Trusted Root Certificate Authorities**. Use the **Import** action to import the previously generated **rootCA.pem** root CA file.
- i. In the same window, go to **Personal** and use the **Import** action to import the previously generated **server.pfx**.
- j. Next, open the **IIS Manager**, select the **RetailHardwareStationWebSite** and select **Edit Bindings** from the right-most menu.
- k. In the new window, select the HTTPS site binding, and then select **Edit**. In the final screen, select the newly installed certificate and select **OK**.
- l. Verify the certificate is correctly being used. In a web browser, go to "`https://<hostname>/HardwareStation/ping`".
- m. Install the certificate on the iOS device:
  - o Copy the **rootCA.pem** file and rename the copy to **rootCA.crt**. - Using OneDrive or another file hosting location, upload the **rootCA.crt** and **server.crt** so that they can be downloaded onto the iOS device.
- n. On the iOS device, go to **Settings > General > Profiles** and select the downloaded profile for the **rootCA.crt**. Select **Install**.
- o. Validate that the profile status updates to **Verified**. Repeat the same process for the **server.crt** file.
- p. Go to **Settings > General > About > Certificate Trust Settings** and enable the installed

root certificate.

- q. On the iOS device, use the hardware station ping URL specified previously to verify that the certificate is trusted.
- r. Open the POS application in **Non-drawer mode** and pair to the hardware station as typically performed.

### Troubleshoot device activation for Modern POS

- The Microsoft account (Azure AD) sign-in page doesn't open.

**Solution:** The Azure AD endpoint might be unreachable. Wait a few minutes, and then try again.

- After you enter the Azure AD account, you receive an error message that states that the user isn't authorized.

**Solution:** Verify that the Azure AD user is mapped to a worker who has POS permission to activate devices. The **Manage device** permission for the worker should be set to **Yes**.

- Device activation isn't completed. It fails during one of the steps.

**Solution:** Follow this checklist to verify that all data is correct:

- Complete the Validate Devices for Activation check in Headquarters, and make sure that the device passes validation.
- On the client computer where you're activating the device, access the Commerce Scale Unit URL health check, and make sure that the health check is passed. Use the following format for the URL:  
`https://MyCompanyName.et.axcloud.dynamics.com/commerce/healthcheck?testname=ping`
- The worker must be mapped to an Azure AD account (under **External identity**).
- The Azure AD account that is mapped must belong to the same tenant.
- To map the worker to the Azure AD account, sign in to Headquarters by using the Admin account for Microsoft Dynamics Lifecycle Services (LCS).
- Make sure that the worker is set up as a Commerce user in the Manager role. (This item is checked by validation.)
- Make sure that the channel is published. (This item is checked by validation.)
- Make sure that the channel database has the synced data from Headquarters, and that download jobs are running.
- Set up the hardware profile under **Registers**. (This item is checked by validation.)
- Make sure that the register and store have a screen layout. (This item is checked by validation.)
- Make sure that a primary address is set up for the legal entity.
- Make sure that the language is set up for the Commerce Data Exchange: Real-time Service user profile (JBB in the demo data).
- Make sure that the Real-time Service profile has the correct access.
- Make sure that the electronic funds transfer (EFT) configuration value is present.

### Troubleshoot Modern POS connectivity

On a single-computer system, such as a developer topology or a demo environment, or when Commerce Scale Unit and Modern POS are installed on the same computer, Modern POS can't complete device activation.

**Solution:** This issue occurs because Modern POS can't make network calls to the same computer (that is, calls to itself). To mitigate this issue, you must enable an AppContainer loopback exception so that communications can occur to the same computer. Various applications will help enabling this loopback for Modern POS. For more information about loopback, see [How to enable loopback and troubleshoot network isolation](#).

## Additional resources

## Install the POS layout designer

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up extended logon functionality for MPOS and Cloud POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers your options for setting up extended logon for Cloud POS and Retail Modern POS (MPOS).

## Setting up extended logon

You can find the setup for bar code masks at **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. The **Functions** FastTab includes the following options that are related to extended logon.

### Staff bar code logon

When the **Staff bar code logon** option is enabled, workers who have an extended logon assigned to their point of sale (POS) credentials can log on by using a bar code.

### Staff bar code logon requires password

When the **Staff bar code logon requires password** option is enabled, the staff bar code logon selects only the worker who is assigned to the extended logon that is presented. Workers must still enter their password when this option is enabled.

### Staff card logon

When the **Staff card logon** option is enabled, workers who have an extended logon assigned to their POS credentials can log on by using a magnetic stripe.

### Staff card logon requires password

When the **Staff card logon requires password** option is enabled, the staff card logon selects only the worker who is assigned to the extended logon that is presented. Workers must still enter their password when this option is enabled.

## Assigning an extended logon

By default, only managers can assign extended logon to workers. To assign extended logon, go to **Extended log on** in POS. Then search for a worker by entering his or her operator ID in the search field. Select the worker, and then click **Assign**. On the next page, swipe or scan the extended logon to assign to the worker. If the swipe or scan is successfully read, the **OK** button becomes available. Click **OK** to save the extended logon for that worker.

## Deleting an extended logon

To delete the extended logon that is assigned to a worker, search for the worker by using the **Extended log on** operation. Select the worker, and then click **Unassign**. All extended logon credentials that are associated with that worker are removed.

## Extending extended logon

The logon service can be extended to support additional extended logon devices, such as palm scanners. For more information, see the POS extensibility documentation.

## Using extended logon

When extended logon is configured, and a worker has been assigned a bar code or magnetic stripe, the worker just has to swipe or scan his or her card while the POS logon page is displayed. If a password is also required before logon can proceed, the worker is prompted to enter his or her password.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Manage activation accounts and validate devices

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how an IT Pro can set up Commerce activation accounts for workers to activate Modern POS or Cloud POS devices.

## Setting up a device activation account for a single worker

This procedure should be completed before you activate Cloud POS.

1. In Commerce, from the **Workers** page, open the **Worker details** page for the worker to assign AAD device activation privileges to. Click **Edit**.
2. On the **Commerce** tab, click the **POS permissions** link. Make sure that the worker is in the Manager Permission group, or that **Manage Devices** is set to **Yes** for the worker.
3. On the **Commerce** tab, under **External identity**, update the values for the following fields:
  - Alias
  - UPN
  - External identifier
4. You can update the **External identity** fields by using an existing AAD account or creating a new AAD account. To update the fields, access the **External identity** options from the **Commerce** main menu (**Commerce** > **Associate existing identity** or **Commerce** > **Create new identity**).
5. To use an existing AAD account, select **Commerce** > **Associate existing identity**. In the slider, click the AAD account that has the correct name, and then click **OK**. The AAD account that is associated with that name and alias is the user's Activation account for Modern POS.
6. Complete and save the changes on the **Workers** page, and then refresh the page. The section that contains external identity information should be updated with the new information. The mapped AAD account is now your Activation account for Cloud POS and Modern POS. This account is mapped to a worker for the required POS permissions. You can use this AAD account for Modern POS or Cloud POS activation.
7. The Create external identity feature creates a new AAD account for you by using the alias that you enter. To update the fields, access the **External identity** options from the **Commerce** main menu (**Commerce** > **Create new identity**).
8. You can either manually enter the alias to generate or use the **Reset to default** button. Then manually enter a strong password, and click **OK**.
9. If the worker is created successfully, you receive a message on the **Workers** page. The mapped AAD account is now the user's Activation account for Cloud POS and Modern POS. This account is mapped to a worker for the required POS permissions. You can use this AAD account for Modern POS or Cloud POS activation.

## Setting up device activation accounts for multiple workers

You can set up activation accounts for multiple workers in bulk. However, this functionality is supported only if you're creating new external identities, not if you're associating identities.

1. In the workers form, select the list of workers to set the activation account for.

2. Click **Commerce > Create external identity** to update the fields. Any AAD accounts that are associated with the workers appear in this pane.

#### NOTE

These accounts aren't device activation accounts until you map them by using the external identity flow options.

3. If you want to use the existing AAD accounts as activation accounts, you can't map them in bulk. Cancel the selection of those workers, and then map them individually by using **Use existing external identity**.
4. To create new AAD accounts and associate them with the workers, so that they can be used as activation accounts, update the **Alias** and **Password** fields, and then click **OK**. In the main worker form, you receive a message as activation accounts are created for each worker.

## Run the Validate Devices for Activation check at headquarters

Before handing an activation account to a worker, an IT Pro must run the Validate devices check for the devices assigned to the worker. This will help identify any potential failures of device activation in advance and fix it before it is given to the worker.

1. Open the **Device** page in HQ (**Retail and commerce > Channel setup > POS setup > Devices**).
2. Select the device to validate for device activation, and then click **Validate devices for activation**. For example, select device **HOUSTON-2**.
3. In the dialog box that appears, select the worker to validate the device for (that is, the worker that you mapped to the AAD account in the previous procedure). For example, select worker **000160**.
4. Click **OK**, and make sure that you receive a message similar to the following: "Pre-Activation validation completed for Device HOUSTON-2 and Staff 000160. Validation: Passed"

## Checklist to follow before activation

1. Complete the **Validate devices for activation** check in HQ, and make sure that the device passes validation.
2. On the client machine where you're activating the device, access the Commerce Scale Unit URL health check, and make sure that the health check is passed. Use the following format:  
`https://c1xtestax404ret.cloud.test.dynamics.com/en/healthcheck?testname=ping`
3. The worker must be mapped to an AAD account (under **External identity**).
4. The AAD account to map must belong to the same tenant.
5. To map the worker to the AAD account, sign in to HQ by using the Admin account for Microsoft Dynamics Lifecycle Services (LCS).
6. Make sure that the worker is set up as a Commerce user in the Manager role (checked by validation).
7. Make sure that the channel data is present in the channel database.
8. Set up the hardware profile under **Registers > Register** (checked by validation).
9. Make sure that the register and store have a screen layout (checked by validation).
10. Make sure that a primary address is set up for the legal entity.
11. Make sure that the electronic funds transfer (EFT) configuration value is present.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create financial dimensions for POS registers and configure dimension values on registers

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through creating financial dimensions for point of sale (POS) registers, and demonstrates how to configure financial dimension values on registers. This procedure doesn't include other related steps, such as creating dimension sets and account structures. Those tasks can be found in other topics. This recording uses USRT demo company.

1. Go to General ledger > Chart of accounts > Dimensions > Financial dimensions.
2. Click New.
3. In the Use values from field, select an option.
4. In the Dimension name field, type a value.
5. Click Activate.
6. Click Close.
7. Click Activate.
8. Click Dimension values.
9. Close the page.
10. Click Save.
11. Close the page.
12. Go to Retail and Commerce > Channel setup > POS setup > Registers.
13. In the list, find and select the desired record.
14. Toggle the expansion of the Financial dimensions section.
15. Click Edit.
16. In the Terminal field, click the drop-down button to open the lookup.
17. In the list, find and select the dimension value for the register being updated.
18. Click Save.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create POS permission groups

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to create a POS permission group. The demo data company used to create this task is USRT. This task is intended for the Commerce operations manager role.

1. In the navigation pane, go to **Modules > Retail and Commerce > Employees > Permission groups**.
2. Select **New**.
3. In the **POS permission group ID** field, type a value.
4. In the **Description** field, type a value.
5. Select **Yes** in the **View time clock entries** field. You can now enable or disable various permissions for your POS Permission group. For some permission you can set a value that will be used to evaluate if the POS user can perform the action. This task guide enables a few permission that might be given to a cashier.
6. Select **Yes** in the **Allow create order** field.
7. Select **Yes** in the **Allow edit order** field.
8. Select **Yes** in the **Allow retrieve order** field.
9. Select **Yes** in the **Allow password change** field.
10. Select **Yes** in the **Allow blind close** field.
11. Select **Save**. After your changes are saved you need to run the Staff distribution schedule to push the changes to commerce channels.
12. In the navigation pane, go to **Modules > Human resources > Jobs > Jobs**.
13. Next we will assign the POS permission group to a Job. In the list, find and select the desired record.
14. Select **Edit**.
15. Expand the **Job classification** section.
16. In the **POS permission group** field, enter or select a value. All Workers in Positions for this Job will use this POS permission group's settings unless the workers POS permissions have been overridden at their Position level.
17. Select **Save**. After your changes are saved you need to run the Staff distribution schedule to push the changes to channels.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create point of sale (POS) visual profiles

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through creating a new point of sale (POS) visual profile. A visual profile contains basic information that determines the appearance of POS registers. You can create several visual profiles and assign specific profiles to run on specific registers. This procedure uses the USRT demo data company.

1. Go to Retail and Commerce > Channel setup > POS setup > POS profiles > Visual profiles.
2. Click New.
3. In the Profile number field, type a value.
4. In the Description field, type a value.
5. In the Application type field, click the drop-down button to open the lookup.
6. In the list, click the link in the selected row.
7. In the Theme field, click the drop-down button to open the lookup.
8. In the list, click the link in the selected row.
9. In the Accent color field, click the drop-down button to open the lookup.
10. In the list, find and select the desired record.
11. In the list, click the link in the selected row.
12. Toggle the expansion of the Login background section.
13. In the Landscape image ID field, select or enter an image ID.
14. In the Portrait image ID field, select or enter an image ID.
15. Toggle the expansion of the Background section.
16. RequestPopup the Image ID.
17. In the list, click the link in the selected row.
18. Click Save.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and install Retail hardware station

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic explains how to configure, download, and install Retail hardware station by using self-service. It also explains how to uninstall Retail hardware station.

## IMPORTANT

It is critical to note that this component utilizes a server certificate. Server certificates must be managed for expiration. By default, a certificate expires in one calendar year (365 days).

## Download Retail hardware station by using self-service

### Configure a new Retail hardware station

#### NOTE

If you're running the February 2016, non-upgraded version of Retail (Initial release), skip step 6.

1. Use your Azure AD credentials to sign in to the Retail trial.
2. On the **Welcome** page, use the menu in the upper left to go to **Retail > Channels > Retail stores > All retail stores**.
3. On the **All retail stores** page, select the retail channel ID of the desired store. The details view for the store appears.

#### NOTE

The Houston store is the most thoroughly prepared store in the demo data.

4. On the **Retail store details** page, on the **Hardware stations** FastTab, select **Add**.

#### NOTE

The Retail Server URL that is used for the selected store is read-only. This URL will be important during the installation of Retail hardware station.

5. In the **Hardware station type** field, select **Shared** to indicate that this hardware station is an Internet Information Services (IIS), installed hardware station that will be used by external point of sale (POS) systems.

#### NOTE

The value **Shared** signifies that the installation is a truly shared hardware station installation, and that it works through HTTPS communication. By contrast, the value **Dedicated** signifies that the hardware station is a part of Modern POS, and that it works through inter-process communication.

6. Select a hardware station profile.

7. Enter the host name of the computer that you're installing Retail hardware station on. Additionally, enter the electronic funds transfer (EFT) terminal ID that is associated with that computer for merchant account information.
8. To utilize the configuration file or initial installation using mass deployment, enter the certificate thumbprint that is to be used during the installation that's detailed in the next section.

#### Download the Retail hardware station installer

1. Use your Azure AD credentials to sign in to the Retail headquarters or Retail trial.
2. On the **Welcome** page, use the menu in the upper left to go to **Retail > Channels > Retail stores > All retail stores**.
3. On the **All retail stores** page, select the retail channel ID of the desired store. The details view for the store appears.

#### NOTE

The Houston store is the most thoroughly prepared store in the demo data.

4. On the **Retail store details** page, select the **Hardware stations** FastTab.

#### NOTE

The Retail Server URL that is used for the selected store is read-only. This URL will be important during the installation of Retail hardware station.

5. Select the hardware station to download, and then select **Download**.

#### NOTE

- Browsers might block the download pop-up that is generated. You must select either **Allow once** or **Options for this site > Always allow**. Then select **Download** again.

6. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)
7. If needed for mass deployment or command line deployment, repeat the above steps for the configuration file download, which is a button next to the **Download** button that you previously selected.

#### NOTE

- If the configuration file downloaded does not have the same base file name as the installer, either rename the XML configuration file to be the same base name or run the installer using the command line to specify the configuration file.
- Note that the configuration file is not required for the installation of Commerce hardware station.

8. After the files have been saved, run the installer. (This step might differ depending on your browser.)

#### Run the installer

#### NOTE

Before you run the Retail hardware station installer, make sure that all [system requirements](#) are met.



The Retail hardware station installer first extracts the associated files and then begins the installation.

1. The installer validates that all prerequisites are met. If a sideloading key is required, the installer requests it. This key is found on the **Devices** page for each device, under **General**.

**NOTE**

- If a system restart is required, the installer informs you of this requirement but can continue the installation.
- Before you can use hardware that is based on the Object Linking and Embedding for Retail Point of Sale (OPOS) standard, the OPOS Common Control Objects must be installed. If they aren't installed, the installer informs you of this requirement but can continue the installation.

2. Enter the Retail Server URL (for example, `https://MyCompanyNameret.axcloud.dynamics.com/Commerce`), and then select **Next**.

**NOTE**

You can find the Retail Server URL at the top of the **Hardware stations** FastTab on the **Retail store details** page.

3. Select a valid Secure Sockets Layer (SSL) certificate to use for HTTPS communication.

**NOTE**

The certificate must use private key storage, and server authentication must be listed in the enhanced key usage property. Additionally, the certificate must be trusted locally, and it can't be expired. It must be stored in the personal certificate store location on the local computer.

4. The next page requests the user that should be used for the IIS application pool. By default in version 1611 and later, the installer can automatically create and use a service account. If you're on a domain or require more specific controls, clear the check box, and then enter the user name and password that the application pool should run under.
5. Enter the HTTPS port to use.

**NOTE**

- You can find the HTTPS port in Retail. (See the configuration instructions earlier in this topic).
- The installer automatically enters the host name. If, for any reason, you must change the host name for the installation, you can change it here. The host name must be the fully-qualified domain name (FQDN) of the system, and it must be entered in the **Host name** field for the selected hardware station entry.

6. The installer installs Retail hardware station and then indicates whether the installation was successful.
7. When the installation is completed, the Install merchant information tool may start. This installer connects to the environment and installs the merchant account information (such as the EFT ID) for the selected hardware station.

#### NOTE

- If the hardware station that was installed won't be used for payment-related work, don't close the **Install merchant information** window without completing the remaining steps. The hardware station won't work unless this installation is successfully completed.

- For version 10.0.6 and above, the install merchant information tool is no longer used. Instead, the merchant information for the hardware station is set by the POS at the time of logon or when the hardware station is made active. If the retail server is not available when the hardware station is subsequently made active, the last known merchant properties will be used by until the connection to the retail server is re-established. If the POS client is not upgraded to version 10.0.6 at the same time the hardware station is upgraded, merchant properties will not be updated until the POS client is upgraded to an equal or later version.

8. The Install merchant information tool might request Azure AD credentials. Enter the Azure AD credentials of the user who is installing Retail hardware station.
9. The Retail Server URL is determined through the Retail hardware station installation and is entered automatically. The installer uses this URL to load the list of stores that the user is connected to via the address book.
10. Select the retail store that the hardware station was installed for.
11. Select the hardware profile that matches the hardware station that was installed on the current computer.
12. Verify that the host names and EFT terminal IDs are correct, based on the current computer and the Retail hardware station configuration that has already been completed in Retail. After you've verified this information, select **Install**.
13. When you receive a message that states that the merchant account information was installed correctly, exit the installer by selecting the **Close** button.

## Help secure Retail hardware station

Current security standards state that the following options should be set in a production environment:

#### NOTE

The hardware station installer automatically makes these registry edits as part of the installation through self-service.

- SSL should be disabled.
- Only Transport Layer Security (TLS) version 1.2 (or the current highest version) should be enabled and used.

## NOTE

By default, SSL and all version of TLS except TLS 1.2 are disabled. To edit or enable these values, follow these steps:

1. Press the Windows logo key+R to open a **Run** window.
2. In the **Open** field, type **Regedit**, and then select **OK**.
3. If a **User Account Control** window appears, select **Yes**.
4. In the new **Registry Editor** window, go to **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\SecurityProviders\SCHANNEL\Protocols**.  
The following keys have been automatically entered to allow for TLS 1.2 only:
  - TLS 1.2\Server:Enabled=1
  - TLS 1.2\Server:DisabledByDefault=0
  - TLS 1.2\Client:Enabled=1
  - TLS 1.2\Client:DisabledByDefault=0
  - TLS 1.1\Server:Enabled=0
  - TLS 1.1\Client:Enabled=0
  - TLS 1.0\Server:Enabled=0
  - TLS 1.0\Client:Enabled=0
  - SSL 3.0\Server:Enabled=0
  - SSL 3.0\Client:Enabled=0
  - SSL 2.0\Server:Enabled=0
  - SSL 2.0\Client:Enabled=0

- No additional network ports should be open, unless they are required for known, specified reasons.
- Cross-origin resource sharing must be disabled and must specify the allowed origins that are accepted.
- Only trusted certificate authorities should be used to procure certificates that will be used on computers that run Retail hardware station.

## IMPORTANT

- Most common, lower-security software and services will stop working after all lower-security standards are disabled. To use them again, go to the preceding registry keys, and set the **Enabled** key from **0** to **1**.
- It's critical that you review security guidelines for IIS and Payment Card Industry (PCI) requirements.

## Troubleshooting

**Modern POS can detect the hardware station in its list for selection, but it can't complete the pairing**

**Solution:** Verify the following list of potential failure points:

- The computer that is running Modern POS trusts the certificate that is used on the computer that runs Retail hardware station.
  - To verify this setup, in a web browser, go to the following URL:  
`https://<Computer Name>:<Port Number>/HardwareStation/ping`
  - This URL uses a ping to verify that the computer can be accessed, and the browser indicates whether the certificate is trusted. (For example, in Internet Explorer, a lock symbol appears in the address bar. When you select this symbol, Internet Explorer verifies whether the certificate is currently trusted. You can install the certificate on the local computer by viewing the details of the certificate that is shown.)
- On the computer that runs Retail hardware station, the port that will be used by the hardware station is opened in the firewall.

- Retail hardware station has properly installed merchant account information through the Install merchant information tool that runs at the end of the Retail hardware station installer.

### **Modern POS can't detect the hardware station in its list for selection**

**Solution:** Any one of the following factors can cause this issue:

- Retail hardware station hasn't been set up correctly in Commerce headquarters. Use the steps earlier in this topic to verify that the hardware station profile and the hardware station are correctly entered.
- The jobs haven't been run to update the channel configuration. In this case, run the 1070 job for channel configuration.
- The hardware station isn't accessible from that computer. Verify that the hardware station URL ping test is accessible from a web browser. This URL can be found at the end of the hardware station installer and is in the following form: `https://<Computer Name>:<Port Number>/HardwareStation/ping`

## Uninstall Retail hardware station

You can use Control Panel in Microsoft Windows to uninstall Retail hardware station.

1. Press the Windows logo key, and then, in the search box, type **Control Panel**. In the list of search results, select **Control Panel**.
2. In Control Panel, select **Programs > Uninstall a program**. The **Programs and Features** window opens.
3. Select **Microsoft Dynamics 365 for Retail hardware station**, and then select **Uninstall** above the list of programs.
4. Wait for the uninstaller to finish removing the program.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up and design receipt formats

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article describes how to modify form layouts to control how receipts, invoices, and other documents are printed. Dynamics 365 Commerce includes a form layout designer that you can use to easily create and modify various kinds of form layouts.

## IMPORTANT

You must set up form layouts and receipt profiles to print receipts and other documents from Retail Modern POS and Cloud POS. You can include multiple form layouts in a receipt profile. You can then assign the receipt profile to a printer by modifying a hardware profile.

## Set up a receipt format

1. Click **Retail and Commerce** > **Channel setup** > **POS setup** > **POS** > **Receipt formats**.
2. On the **Receipt format** page, click **New** to create a new form layout, or select an existing form layout.
3. In the **Receipt format** field, enter an identifier for the form layout, and then select the type of receipt that this layout is used for. You can also enter a description and a short name for the receipt in the **Title** field.
4. On the **General** FastTab, select an option to define the print behavior:
  - **Always print** – The receipt is printed automatically, as appropriate.
  - **Do not print** – The receipt isn't printed.
  - **Prompt user** – The user is prompted to print the receipt.
  - **As required** – This option is used only for gift receipts. When this option is selected, the user can print a gift receipt from the **Change** page, if a gift receipt is required.

## Print images

The receipt designer includes a **Logo** variable that can be used to specify images to be printed on the receipt. Images that are included in receipts using the **Logo** variable should be monochrome bitmap (.bmp) file types. If a .bmp image is specified in the receipt designer, but is not printing when sent to the printer, the file size may be too large or the pixel dimensions on the image are not compatible with the printer. If this occurs, try reducing the image file resolution.

## Design a receipt format

Use the form layout designer to graphically create the layout of the form document. The **Receipt format designer** page has three sections: **Header**, **Lines**, and **Footer**. Some types of form layouts use elements from all three sections, whereas other types use elements from only one or two sections. To view the elements that are available for each section, click the appropriate button in the navigation pane on the left side of the page.

1. Click **Retail and Commerce** > **Channel setup** > **POS setup** > **POS** > **Receipt formats**.
2. On the **Receipt format** page, select a form layout, and then click **Designer**.
3. Click **Run** to start to install the Commerce designer host.
4. On the Notification bar that appears at the bottom of the Internet Explorer window, click **Open** to start to

install the one-click designer. (The Notification bar might appear in a different location in other browsers.) The progress indicator shows the progress of the installation process.

5. After the installation is completed, enter your Commerce user name and password, and then click **Sign in** to start the designer.
6. After your credentials are validated and the designer starts, you can start to design the receipt format or modify an existing format.
7. To create the elements of the form, select the **Header**, **Lines**, or **Footer** section, and then drag an element from that section to the workspace. Most elements contain variables that are automatically populated with data from the database. Other elements, such as **Text**, let you print custom text on the receipt.

#### NOTE

You can specify how many lines each section spans by adjusting the number in the lower-right corner of that section. To make it easier to modify a section, increase its height by dragging the sizing bar at the bottom of the section. The height of the section on the workspace doesn't affect the number of lines on the actual receipt.

8. After you drag an element to the workspace, set the properties for the part in the **Object information** pane at the bottom of the page. Enter one or more of the following settings:
  - **Align** – Set the alignment of the field to either **Left** or **Right**.
  - **Fill char** – Specify the white space character. By default, an empty space is used, but you can enter any character.
  - **Prefix** – Enter the value that appears at the beginning of the field. This setting applies only to the **Lines** section of the layout.
  - **Characters** – Specify the maximum number of characters that the field can contain if the element contains a variable. If the text in the field is longer than the number of character that you specify, the text is truncated to fit the field.
  - **Variable** – This check box is selected automatically if the element contains a variable and can't be customized.
  - **Font type** – Set the font style to either **Regular** or **Bold**. Bold letters use two times as much space as regular letters. Therefore, some characters might be truncated.
  - **Font size** – Set the font size to either **Regular** or **Large**. Large letters are two times higher than regular letters. Therefore, using large letters may lead to overlapping text in the receipt.
  - **Delete** – Click this button to remove the selected part from the form layout.

## Assign receipt profiles

Receipt profiles are assigned directly to printers through the hardware profile.

1. Open the hardware profile by clicking **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profile**.
2. Select the printer, and then, in the **Receipt profile** field, assign the receipt profile to use on the register.

#### NOTE

If two printers are used, one printer can be used to print standard 40-column thermal receipts. The second printer is typically used to print full-page receipt types that require more information. These receipt types include customer order receipts and customer invoices.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Send email receipts from Modern POS (MPOS)

2/18/2021 • 4 minutes to read • [Edit Online](#)

In Modern Point of Sale (MPOS), you can send receipt emails when a transaction is tendered at the point of sale (POS).

## Prerequisite

To send email receipts, you must configure a Simple Mail Transfer Protocol (SMTP) server.

## Set up email receipts

### Set default options for email receipts

1. Select **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**.
2. On the **Posting** tab, on the **Email receipt** FastTab, in the **Receipt option** field, select a default option:
  - **Standard receipt** – Print receipts from the POS register.
  - **Email** – Send receipts to customers in email messages.
  - **Both** – Print receipts from the POS register, and send receipts to customers in email messages.
3. In the **Subject** field, enter the text that should appear by default on the subject line of a receipt that is sent as an email message.

### Set email receipt options for a customer

1. Go to **Retail and Commerce > Customers > All customers**.
2. On the **All customers** list page, select a customer, and then select **Edit**.
3. On the customer details page, on the **Commerce** FastTab, in the **Receipt option** field, select an option:
  - **Standard receipt** – The customer will receive only printed receipts. The printed receipt is generated from the POS register.
  - **Email** – The customer will receive only email receipts.
  - **Both** – The customer will receive both printed receipts and email receipts.
4. If you selected either **Email** or **Both** in the **Receipt option** field, enter the customer's email address in the **Receipt email** field.

### Set up an email receipt profile

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Receipt profiles**.
2. Press **Ctrl+N** to create a receipt profile.
3. Enter values in the **Receipt profile ID** and **Description** fields.
4. On the **General** FastTab, select **Add** to add a receipt type.
5. Select **Receipt** as the receipt type, and select the receipt format to use for email receipts.

### Add an email receipt profile to the functionality profile

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**.
2. Select **Edit**.
3. On the **General** FastTab, in the **Receipt profile ID** field, specify an email receipt profile.



## Set up an email template for receipts

1. Go to **Organization email templates** under either **Retail and Commerce > Headquarters setup > Setup > Organization email templates** or **Organization administration > Setup > Organization email templates**.
2. Select **New**.
3. Enter information in the following fields:
  - In the **Email ID** field, enter **EmailRecpt**.
  - In the **Email description** field, enter an optional description.
  - In the **Sender name** field, specify the name that should appear as the sender of the email. Customers will see this name as the **From** name on the email.
  - In the **Sender email** field, specify a valid email address. Customers will see this email address as the **From** email address on the email.
4. Under **General**, enter information in the following fields:
  - In the **Default language code** field, select a language. This is the language that the receipt will be sent in when templates for multiple languages are configured and the store or customer's preferred language doesn't match any of those additional languages.
5. In the **Email message content** pane, select **New** to create a new template instance. Enter information in the following fields:
  - In the **Language** field, specify the language this template will be localized in. Note that this only applies to emailed receipts that contain HTML with static content above and/or below the `%message%` placeholder.
  - In the **Subject** field, enter a title for the email receipts.
  - Select the **Has body** check box.
  - Select **Edit** to upload your template HTML. At a minimum, your template instance must contain the following code:

```
<pre>
%message%
</pre>
```

You can also add HTML to display a header, footer, logo or any other static content that you want included in the receipt email. For more information about creating HTML receipt templates, see [Create a template for emailed receipts](#).

5. Depending on the settings that you configured, you must run the appropriate distribution schedule jobs to synchronize the changes to MPOS.
  - **1010** – Customer
  - **1070** – Channel configuration
  - **1090** – Registers
  - **1110** – Global configuration

## MPOS transactions

After the changes are synchronized to the store, MPOS prompts the user for an email address for each transaction (if this feature is enabled). If an email address is already on file for the customer, that address appears in the email address prompt. If a customer hasn't been named, or if an email address hasn't been entered for a named customer, enter an email address, and then select **Send**. When the transaction is finalized,

the real-time service will send the customer an email that has the receipt in the body of the message, as you configured earlier.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Reset receipt numbers

2/18/2021 • 4 minutes to read • [Edit Online](#)

## NOTE

We require that you select the **Independent sequence** property for all receipt types in the functionality profile before using this feature. Also, the system time zone of the device, where the POS is being used, should match the corresponding store time zone. Due to these limitations, we recommend that you do not use this feature in production while we work to fix these issues in a future release.

Retailers generate receipt numbers for various actions in the store, such as cash and carry transactions, return transactions, customer orders, quotations, and payments. Although retailers define their own receipt formats, some countries or regions have regulations that put restrictions on these receipt formats. For example, these regulations might limit the number of characters on the receipt, require consecutive receipt numbers, restrict some special characters, or require a reset of receipt numbers at the beginning of the year. Microsoft Dynamics 365 Commerce makes the process of managing receipt numbers very flexible, to help retailers meet regulatory requirements. This topic explains how to use the functionality for resetting receipt numbers.

In Commerce, receipt formats can be alphanumeric. You can put both static content and dynamic content in them. Static content includes alphabetic character, numbers, and special characters. Dynamic content includes one or more characters that represent information such as the store number, terminal number, date, month, year, and number sequences that are automatically incremented. The formats are defined in the **Receipt numbering** section of the functionality profile. The following table describes the characters that represent the dynamic content.

CHARACTERS	DESCRIPTION
S	The character <b>S</b> is used for the store number. For example, if a store is numbered HOUSTON1, the format <b>SSS</b> shows "ON1" on the receipt. The format <b>SSSSS</b> shows "STON1" on the receipt.
T	The character <b>T</b> is used for the terminal number. For example, if a terminal is numbered 0001, the format <b>TTTT</b> shows "0001" on the receipt.
C	The character <b>C</b> is used for the staff ID number. For example, if a staff member has an ID of 000160, the format <b>CCCC</b> shows "0160" on the receipt.
ddd	The characters <b>ddd</b> correspond to the day of the year, from 1 through 366. For example, on January 15, the format <b>ddd</b> shows "015" on the receipt.
MM	The characters <b>MM</b> are used for the two-digit month. For example, in January, the format <b>MM</b> show "01" on the receipt.
DD	The characters <b>DD</b> are used for the two-digit day of the month. For example, on January 15, the format <b>DD</b> shows "15" on the receipt.

CHARACTERS	DESCRIPTION
YY	The characters <b>YY</b> are used for the two-digit year. For example, in any month during the year 2020, the format <b>YY</b> shows "20" on the receipt.
#	A number sign ( <b>#</b> ) is used for sequential numbering. For example, the format <b>####</b> shows "0001," "0002," "0003," and so on, on the receipt.

You can reset the sequential numbering of the receipt on a specific date. Then, for the first transaction that occurs after 12:00 AM on the selected reset date, the system resets the receipt's number sequence to 1. You can also specify whether the reset occurs only one time, or whether it recurs every year. If yearly recurrence is specified, the reset automatically occurs every year until the retailer chooses to stop it.

To turn on the reset, follow these steps.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**.
2. On the **Receipt numbering** FastTab, select **Reset number reset date**.
3. In the drop-down dialog box, in the **Reset date** field, select a future date when the reset should occur.
4. In the **Reset receipt type** field, select **One time only** or **Yearly**.
5. Select **OK**.

After you select a date, it appears in the **Next receipt number reset date** column. The reset date is applicable to all receipt transaction types. Therefore, the receipt number sequence will be reset for all receipt types.

When the reset date arrives, the receipt number is reset for the first transaction of each type. In addition, in the functionality profile, the reset date is moved from the **Next receipt number reset date** column to the **Current receipt number reset date** column. This change indicates that if a register isn't used on the reset date, the receipt number will be reset the next time that the register *is* used. For example, on December 3, 2019, you select **January 1, 2020**, as the reset date. On January 1, when the registers make their first transaction, the receipt number are reset. However, one register isn't used at all during December and January, but then starts to be used in February. In this case, because a reset action was defined, the receipt number for that register will be reset when the register makes its first transaction in February.

You can use the **Clear reset date** functionality to clear future reset dates. However, if the reset date occurred in the past, it can't be undone. Therefore, the reset will still occur for all registers where the reset hasn't yet occurred.

**NOTE**

Depending on the reset date that you select, and the receipt format, you might have duplicate receipt numbers. Although the point of sale (POS) system can handle these situations, they increase the amount of time that is required to process returns, because sales associates must select among the duplicate receipts. Other complications that are related to data cleanup can occur if the duplicate receipts weren't a planned consequence. Therefore, we recommend that you use dynamic date characters (for example, **ddd**, **MM**, **DD**, and **YY**) to help prevent duplicate receipt numbers after a reset.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Embed processor credit card receipts in customer receipts

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to embed credit card receipts from payment processors into a customer's itemized transaction receipt. This capability is available in Microsoft Dynamics 365 Commerce version 10.0.8 and later.

## Key terms

TERM	DESCRIPTION
Customer's receipt	The receipt that is generated for a cash-and-carry transaction at the point of sale (POS).
Customer's credit card receipt	The credit card receipt that is printed as a record of the credit card payment or other electronic payment that is used in a transaction.

## Overview

This topic describes the steps that are required to embed the credit card receipt from a payment processor directly into a customer's receipt. In Dynamics 365 Retail version 10.0.7 and earlier, several elements from the customer's credit card receipt could be embedded into the customer's itemized transaction receipt. However, the actual receipt that comes from the payment processor could not be included. That solution wasn't acceptable for all retailers, because the configurable receipt fields in the customer's credit card receipt didn't always include all the details that are stipulated by local statutory requirements.

## Prerequisites

The following items are required to embed processor credit card receipts into customer receipts:

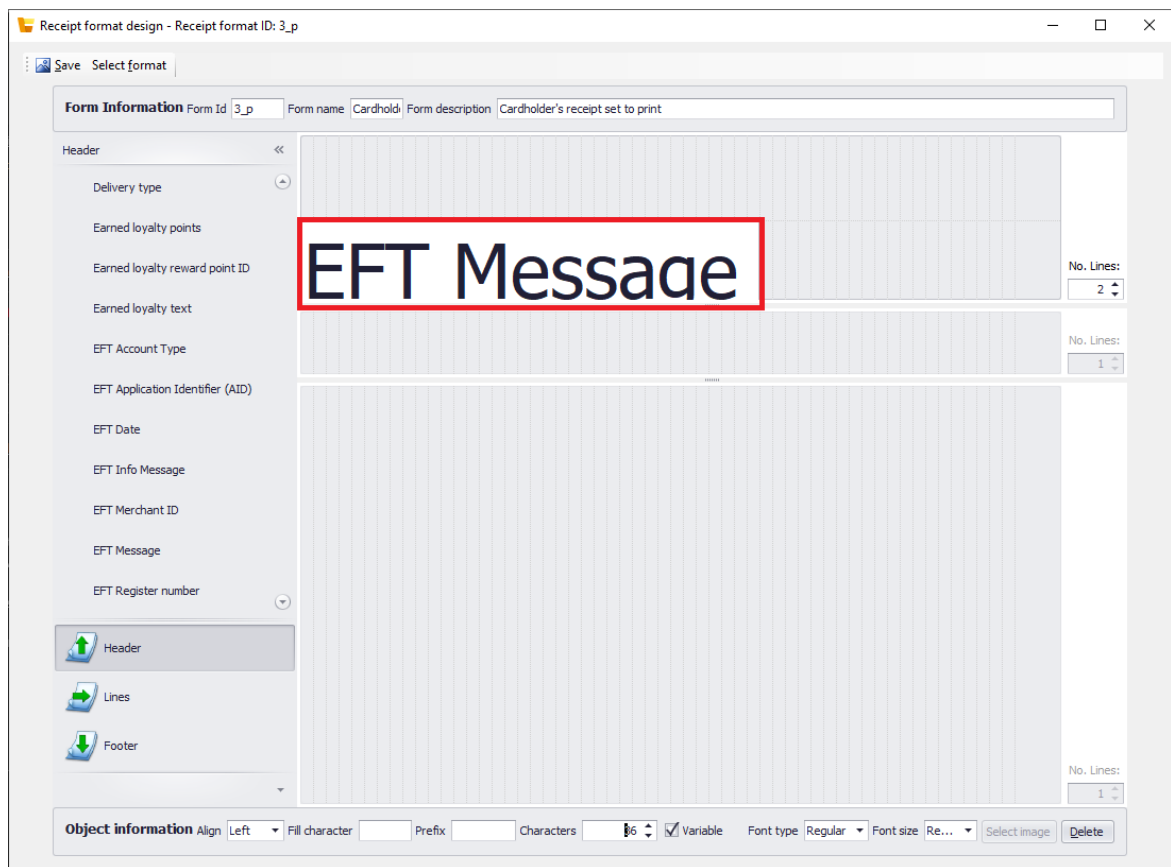
- A payment connector that is implemented in accordance with the payments software development kit (SDK)
- A POS that has a working printer

## Set up receipts

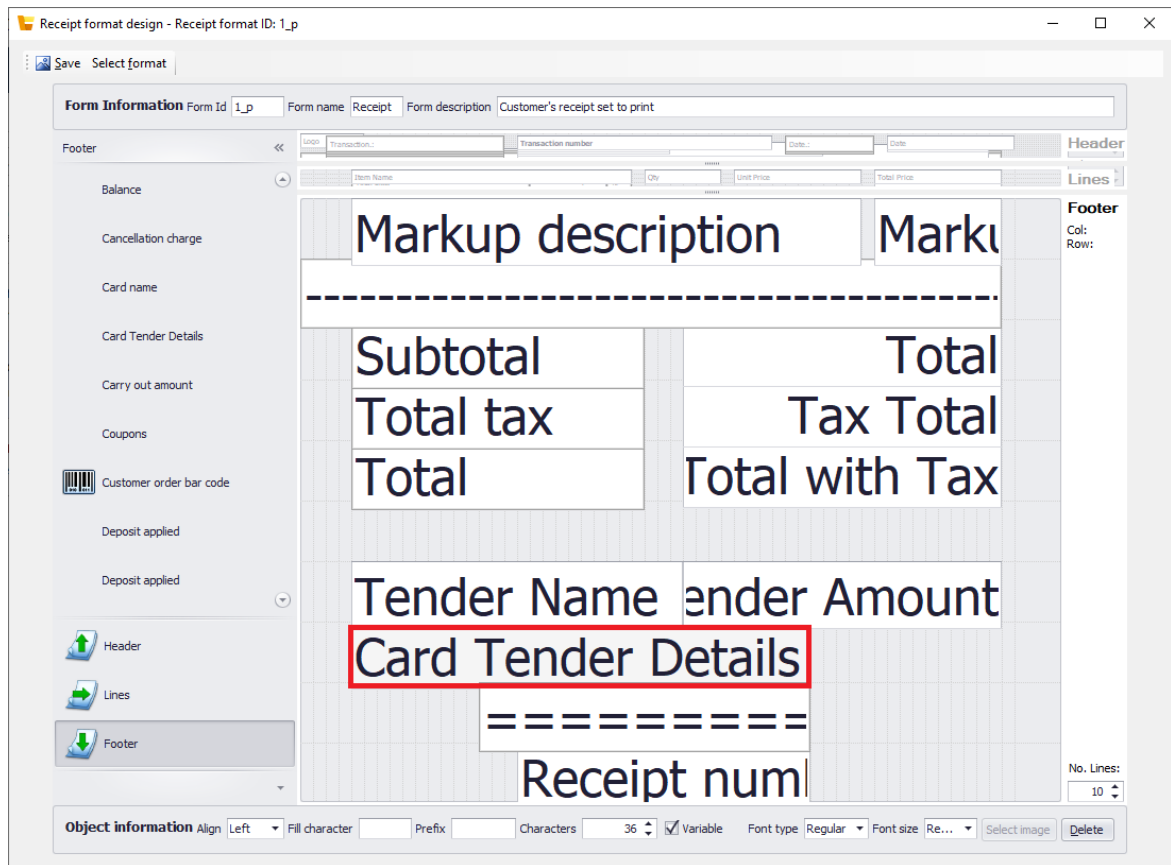
1. In the POS, search for "receipt formats" to open the **Receipt formats** page.
2. Select the receipt of the **Customer's credit card receipt** type that will be used at the POS. If you're using demo data, select receipt format 3\_P, and set the **Print Behavior** field to **Do not print**.
3. Select **Designer** to open the receipt designer.
4. Remove all the fields from the receipt format.

To edit a section of the receipt, you must first select that section at the bottom of the left pane in the receipt designer. Then select the desired receipt variable in the selected section. Finally, to delete the selected variable, you can use the **Alt+D** keyboard shortcut.

5. Select the **Header** section at the bottom of the left pane, and then drag the **EFT Message** receipt variable into the header.

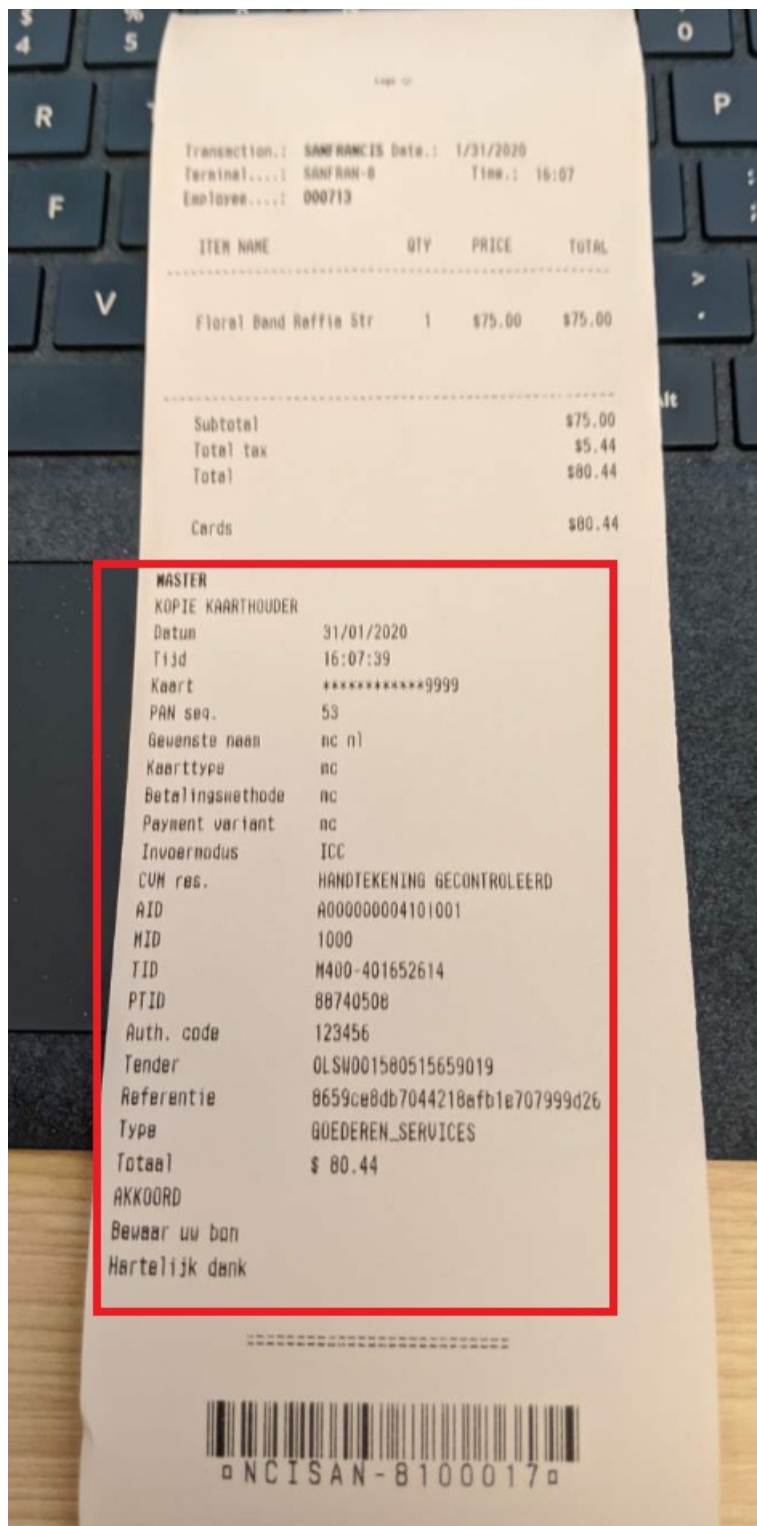


6. Select **Save**.
7. While the receipt designer is still open, select **Select format** in the upper-left corner to open the receipt selector.
8. In the receipt selector, select the receipt of the **Receipt** type that will be used at the POS. If you're using demo data, select receipt format **1\_p**.
9. In the receipt designer, select the **Footer** section at the bottom of the left pane, and then drag the **Card Tender Details** receipt variable into the footer.



10. Select **Save**.
11. Sync the changes to the POS by using the **1090** distribution schedule.
12. Close the shift in the POS, and then open a new shift.
13. Perform a credit card transaction to confirm that the processor's credit card receipt is embedded into the customer's receipt.





#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure the functionality profile for a sales representative

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates how to configure a store's functionality profile settings that apply to sales representatives using the USRT demo data company.

1. Go to Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles.
2. Click Edit.
3. Expand the Functions section.
  - You can use the functionality profile settings to configure POS to automatically add the cashier's default sales group, to prompt for sales groups, and to require sales groups.
4. In the Prompt for sales representative field, select an option.
5. Select Yes in the Require sales representative field.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create financial dimensions for retail channels and configure dimension values on stores

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through creating a commerce channel financial dimension with dimension values and steps to configure financial dimension values on stores. The topic does not include other related steps, such as creating dimension sets and account structures. This procedure uses the USRT company in demo data.

1. Go to General ledger > Chart of accounts > Dimensions > Financial dimensions.
2. Click New.
3. In the Use values from field, select 'Commerce channels'.
4. In the Dimension name field, type a value.
5. Click Activate.
6. Click Close.
7. Click Activate.
8. Click Dimension values.
9. Close the page.
10. Click Save.
11. Close the page.
12. Go to Retail and Commerce > Channels > Stores > All stores.
13. In the list, click the link in the selected row.
14. Toggle the expansion of the Financial dimensions section.
15. Click Edit.
16. In the Commerce channel field, click the drop-down button to open the lookup.
17. In the list, find and select the dimension value for the store being updated.
18. In the list, click the link in the selected row.
19. In the CostCenter field, click the drop-down button to open the lookup.
20. In the list, find and select the desired record.
21. In the list, click the link in the selected row.
22. In the Department field, click the drop-down button to open the lookup.
23. In the list, find and select the desired record.
24. In the list, click the link in the selected row.
25. Click Save.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up an online channel

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to create a new online channel in Microsoft Dynamics 365 Commerce.

## Overview

Dynamics 365 Commerce supports multiple retail channels. These retail channels include online stores, call centers, and retail stores (also known as brick-and-mortar stores). Online stores give customers the option of purchasing products from the retailer's online store in addition to its retail stores.

To create an online store in Commerce, you must first create an online channel. Before you create a new online channel, ensure that you have completed the [Channel set up prerequisites](#).

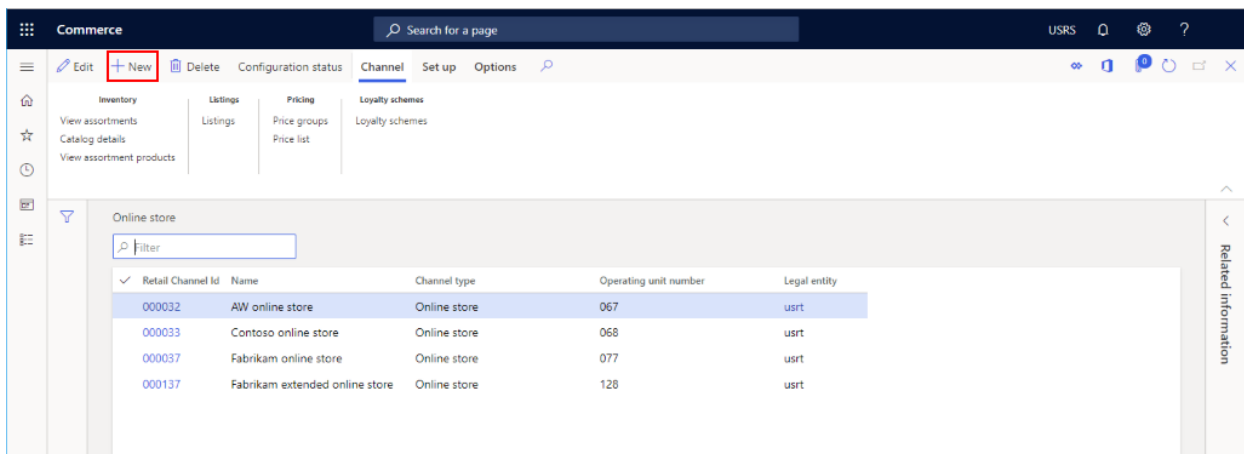
Before you can create a new site, at least one online store must be created in Commerce. For more information, see [Create an e-Commerce site](#).

## Create and configure a new online channel

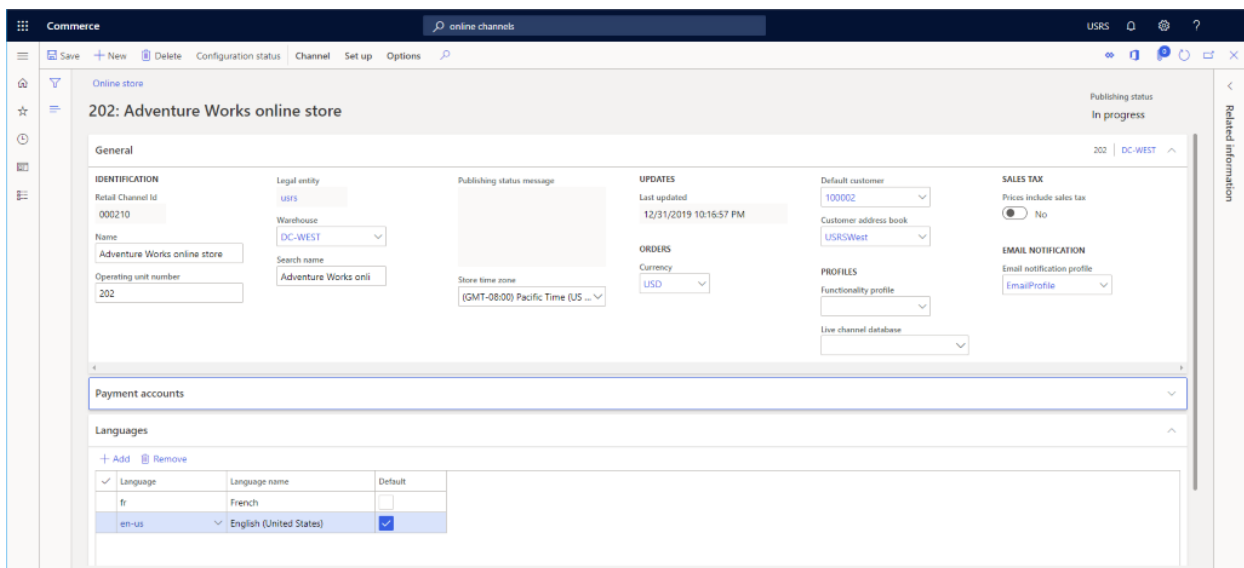
To create and configure a new online channel, follow these steps.

1. In the navigation pane, go to **Modules > Channels > Online Stores**.
2. On the action pane, select **New**.
3. In the **Name** field, provide a name for the new channel.
4. In the **Legal entity** drop-down, enter the appropriate legal entity.
5. In the **Warehouse** drop-down, enter the appropriate warehouse.
6. In the **Store time zone** field, select the appropriate time zone.
7. In the **Currency** field, select the appropriate currency.
8. In the **Default customer** field, provide a valid default customer.
9. In the **Customer address book** field, provide a valid address book.
10. In the **Functionality profile** field, select a functionality profile if applicable.
11. In the **Email notification profile** field, provide a valid email notification profile.
12. On the action pane, select **Save**.

The following image shows the creation of a new online channel.



The following image shows an example online channel.



## Set up languages

If your e-Commerce site will support multiple languages, expand the **Languages** section and add additional languages as needed.

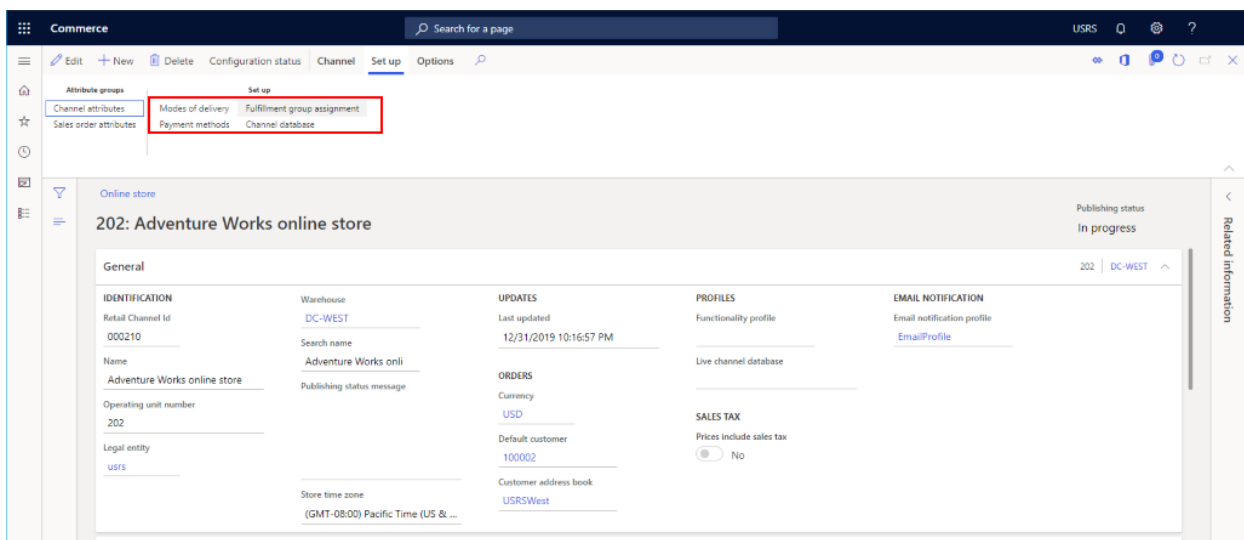
## Set up payment account

From within the **Payment account** section, you can add a third-party payment provider. For information on setting up an Adyen payment connector, see [Dynamics 365 Payment Connector for Adyen](#).

## Additional channel setup

Additional tasks that are required for online channel setup include setting up payment methods, modes of delivery, and the fulfillment group assignment.

The following image shows **Modes of delivery**, **Payment methods**, and **Fulfillment group assignment** setup options on the **Set up** tab.



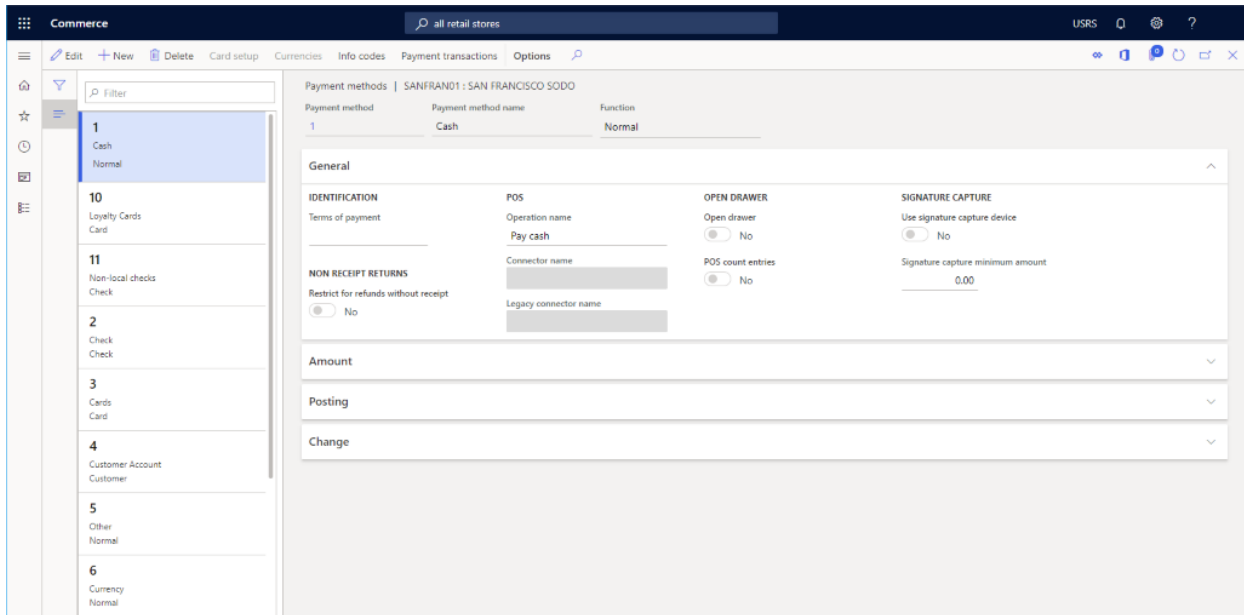
## Set up payment methods

To set up payment methods, for each payment type supported on this channel follow these steps.

1. On the action pane, select the **Set Up** tab, then select **Payment methods**.
2. On the action pane, select **New**.
3. In the navigation pane, select a desired payment method.

4. In the **General** section, provide an **Operation name** and configure any other desired settings.
5. Configure any additional settings as required for the payment type.
6. On the action pane, select **Save**.

The following image shows an example of a cash payment method.



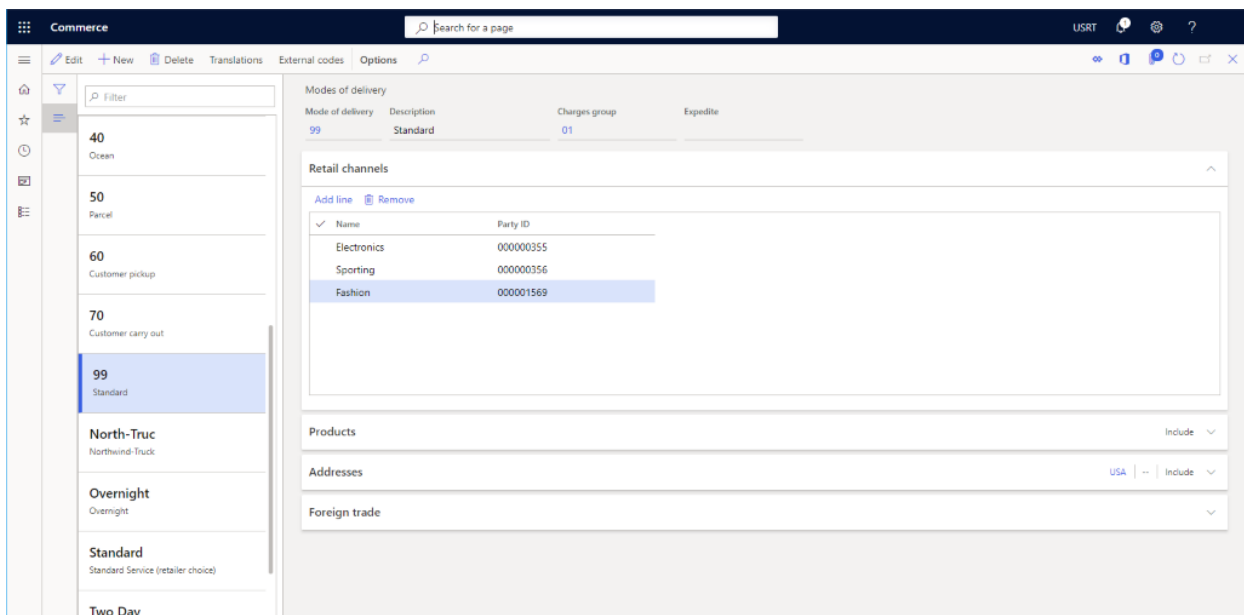
### Set up modes of delivery

You can see the configured modes of delivery by selecting **Modes of delivery** from the **Set up** tab on the **Action pane**.

To change or add a mode of delivery, follow these steps.

1. In the navigation pane, go to **Modules > Inventory management > Modes of delivery**.
2. On the action pane, select **New** to create a new mode of delivery, or select an existing mode.
3. In the **Retail channels** section, select **Add line** to add the channel. Adding channels using organization nodes instead of adding each channel individually can streamline adding channels.

The following image shows an example of a mode of delivery.

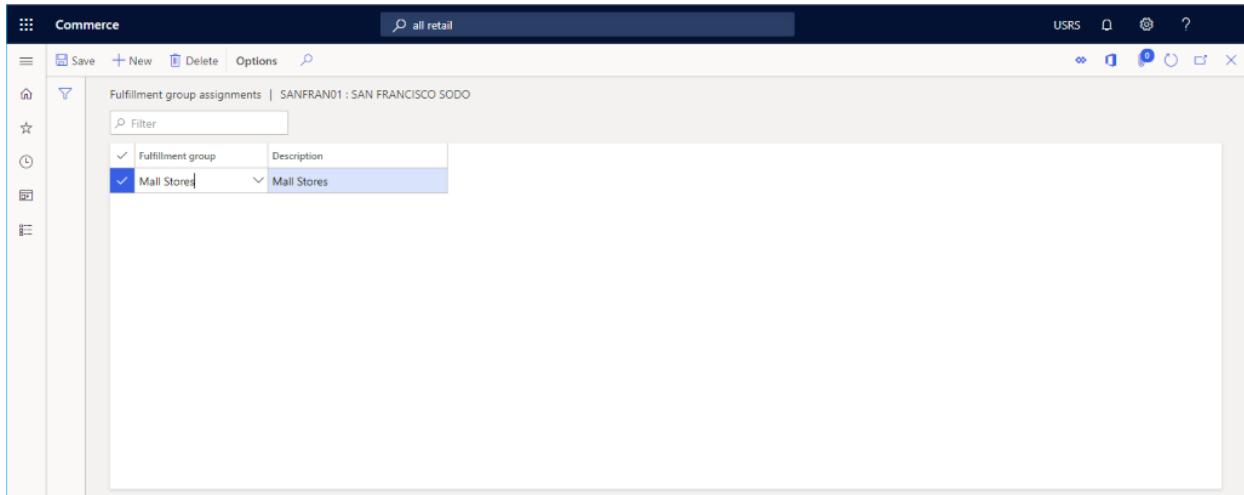


### Set up a fulfillment group assignment

To set up a fulfillment group assignment, follow these steps.

1. On the action pane, select the **Set up** tab, then select **Fulfillment group assignment**.
2. On the action pane, select **New**.
3. In the **Fulfillment group** drop-down list, select a fulfillment group.
4. In the **Description** drop-down list, enter a description.
5. On the action pane, select **Save**.

The following image shows an example of a fulfillment group assignment setup.



## Additional resources

[Channels overview](#)

[Channel setup prerequisites](#)

[Set up a retail channel](#)

[Set up a call center channel](#)

[Dynamics 365 Payment Connector for Adyen](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Call center sales functionality

2/18/2021 • 2 minutes to read • [Edit Online](#)

In Dynamics 365 Commerce, a call center is a type of channel that can be defined in the application. Defining a specific channel for your call center entities allows the system to tie specific data defaults and order processing defaults to sales orders created by a user of the call center channel.

Call center features include advanced price and promotions, catalogs, gift cards, loyalty programs, and coupons. Call center orders are also leveraged by the point of sale (POS) application to support cross-channel order fulfillment scenarios.

It's important to note that while the call center module can be utilized by other industries outside of Commerce, the current release of the call center application hasn't been optimized for use in business-to-business (B2B) order processing scenarios, or scenarios where orders have a large number of sales lines. It's recommended that users who want to utilize the call center features for order processing outside of typical direct-to-consumer transaction processing, take adequate time to test and validate that enabling call center functionality will meet functional and performance needs.

In addition to supporting order creation, the call center module also provides a user-friendly customer service application that makes it easier for users to locate customer accounts and review all of the related customer order data and attributes. The customer service screen is designed to enable a user to quickly access order-related data that will allow them to answer the most common order-related questions received from customers.

This page provides links to relevant documentation related to the setup, configuration, and functional use of the call center features.

## Configure the call center

[Set up call center channels](#)

## Configure order processing

[Set up and work with call center fraud alerts](#)

[Configure and work with call center order holds](#)

## Configure payment processing

[Payment methods in call centers](#)

## Configure delivery modes

[Configure call center delivery modes and charges](#)

## Configure direct marketing

[Call center catalogs](#)

[Set up Recency, Frequency, and Monetary \(RFM\) analysis](#)

## Configure continuity programs



## Set up continuity programs for call centers

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up a call center channel

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to create a new call center channel in Microsoft Dynamics 365 Commerce.

## Overview

In Dynamics 365 Commerce, a call center is a type of Commerce channel that can be defined in the application. Defining a channel for your call center entities allows the system to tie specific data and order processing defaults to sales orders. While a company can define multiple call center channels in Commerce, it is important to note that an individual user may only be linked to one call center channel.

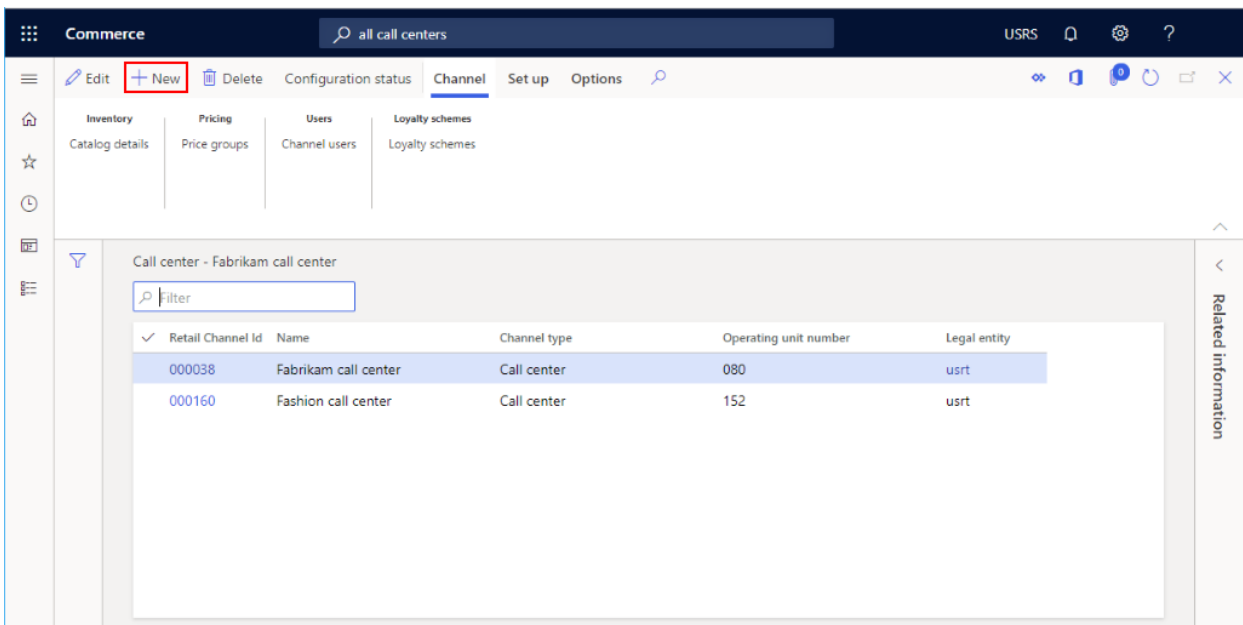
Before you create a new call center channel, ensure that you have completed the [Channel setup prerequisites](#).

## Create and configure a new call center channel

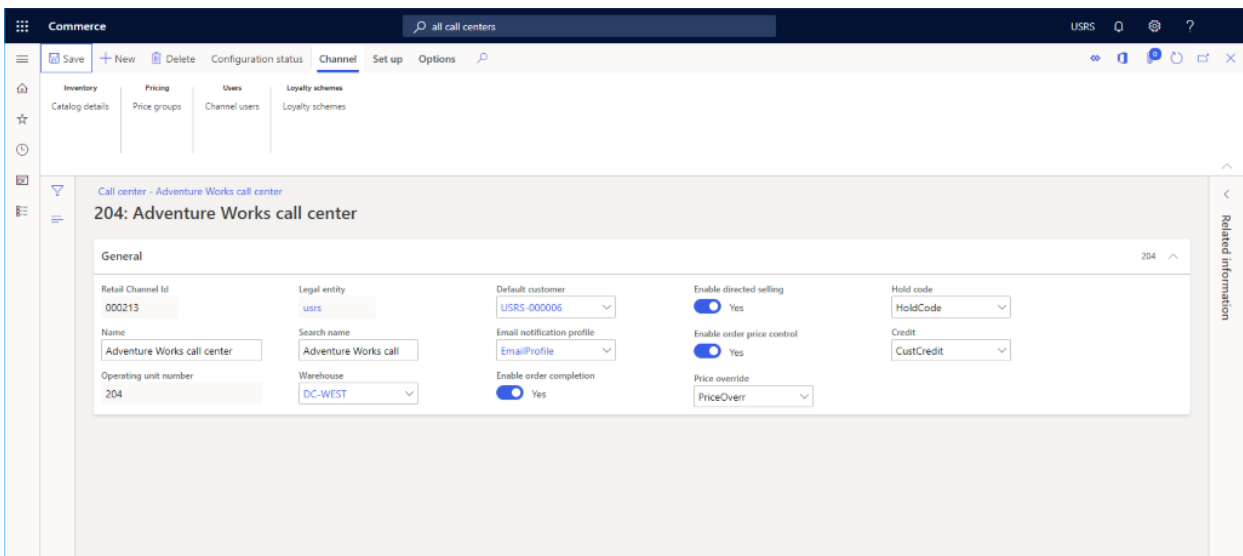
To create and configure a new call center channel, follow these steps.

1. In the navigation pane, go to **Retail and Commerce > Channels > Call centers > All call centers**.
2. On the action pane, select **New**.
3. In the **Name** field, provide a name for the new channel.
4. Select the appropriate **Legal entity** from the drop-down.
5. Select the appropriate **Warehouse** location from the drop-down. This location will be used as the default on sales orders created for this call center channel, unless other defaults have been defined at the customer or item level.
6. In the **Default customer** field, provide a valid default customer. This data is used to assist in autopopulating defaults when new customer records are created. When creating call center orders, it is not advisable to create orders for the default customer.
7. In the **Email notification profile** field, provide a valid email notification profile. As call center orders are created and processed, the email notification profile is used to trigger automated email alerts to customers with information about their order status.
8. Provide a **Price override** info code. You may need to create an info code for this first. This info code provides the set of reason codes that the user will be prompted to choose from when using the price override functionality on a call center order.
9. Provide a **Hold code** info code. You may need to create an info code for this first. This info code provides the set of optional reason codes that the user will be prompted to choose from when placing an order on hold.
10. Provide a **Credit** info code. You may need to create an info code for this first. This info code provides the set of reason codes that the user can choose from when using the order credit functionality of call center to give misc refunds to the customer for customer service reasons.
11. Optional: set up financial dimensions on the **Financial dimensions** FastTab. The dimensions entered here will default on any sales order created in this call center channel.
12. Click **Save**.

The following image shows the creation of a new call center channel.



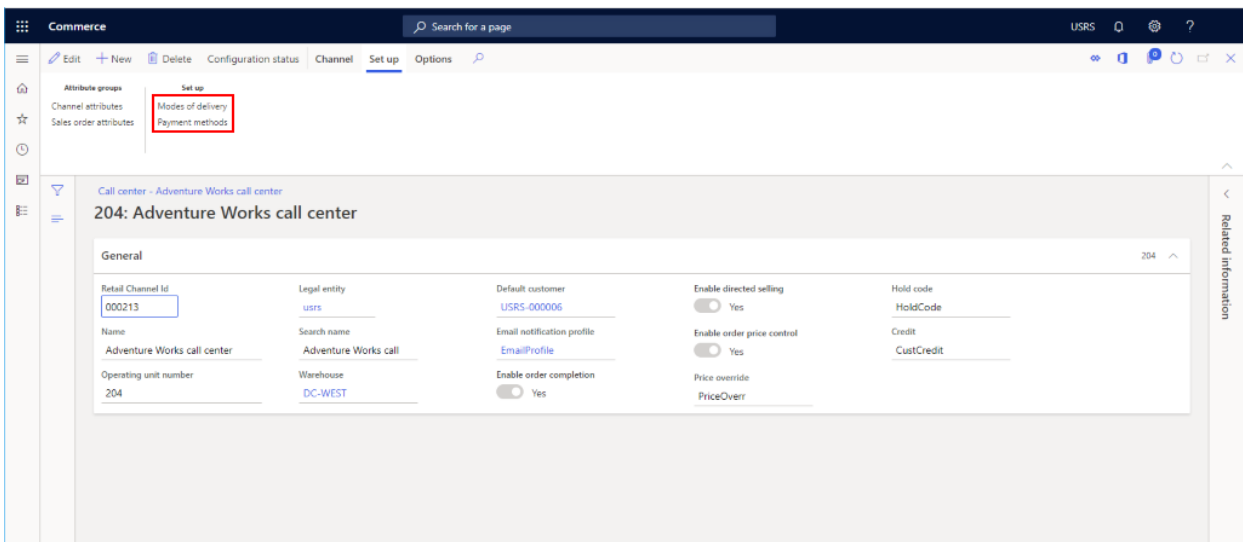
The following image shows an example call center channel.



## Additional channel setup

Additional tasks required for call center channel setup include setting up payment methods and modes of delivery.

The following image shows **Modes of delivery** and **Payment methods** setup options on the **Set up** tab.

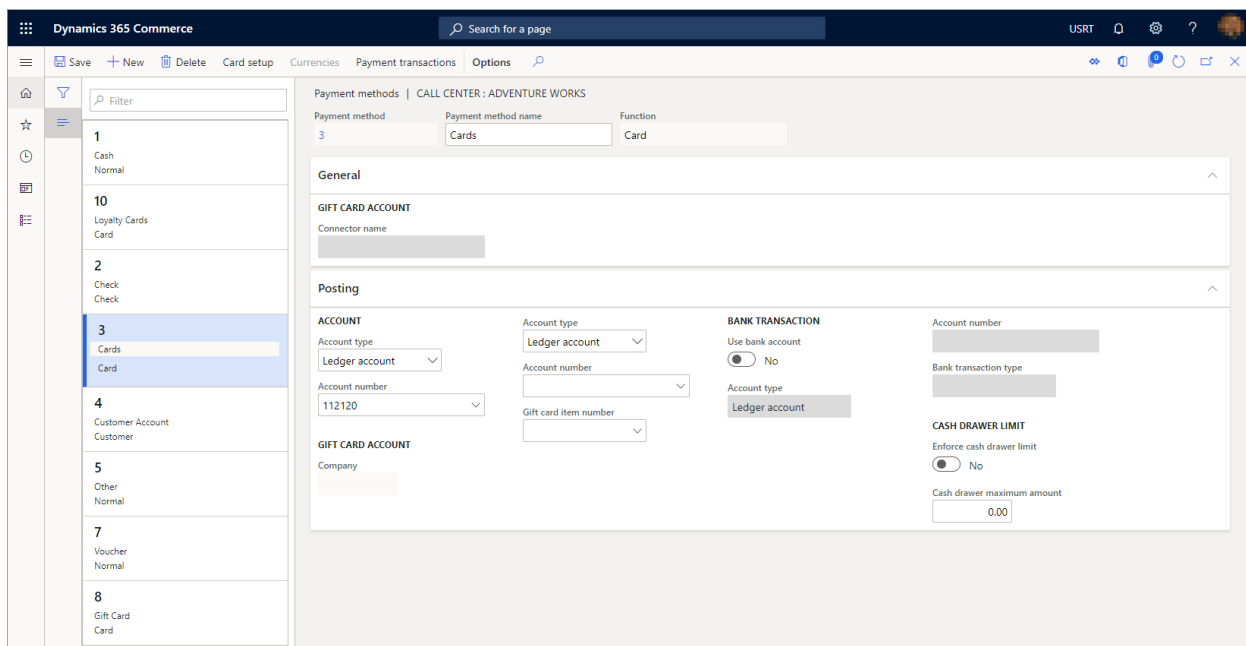


## Set up payment methods

To set up payment methods, follow these steps for each payment type supported on this channel. Users will be required to select from pre-defined payment methods to link them to the call center channel. Before setting up your call center payment methods, first set up your master methods of payment in **Retail and Commerce > Channel setup > Payment methods > Payment methods**.

1. On the action pane, select the **Set up** tab, and then select **Payment methods**.
2. On the action pane, select **New**.
3. In the navigation pane, select a payment method from the pre-defined payments available.
4. Configure any additional settings as required for the payment type. For credit cards, gift cards, or loyalty cards, additional setup is required by selecting the **Card setup** function.
5. Configure proper posting accounts for the payment type in the **Posting** section.
6. On the action pane, click **Save**.

The following image shows an example of a cash payment method.



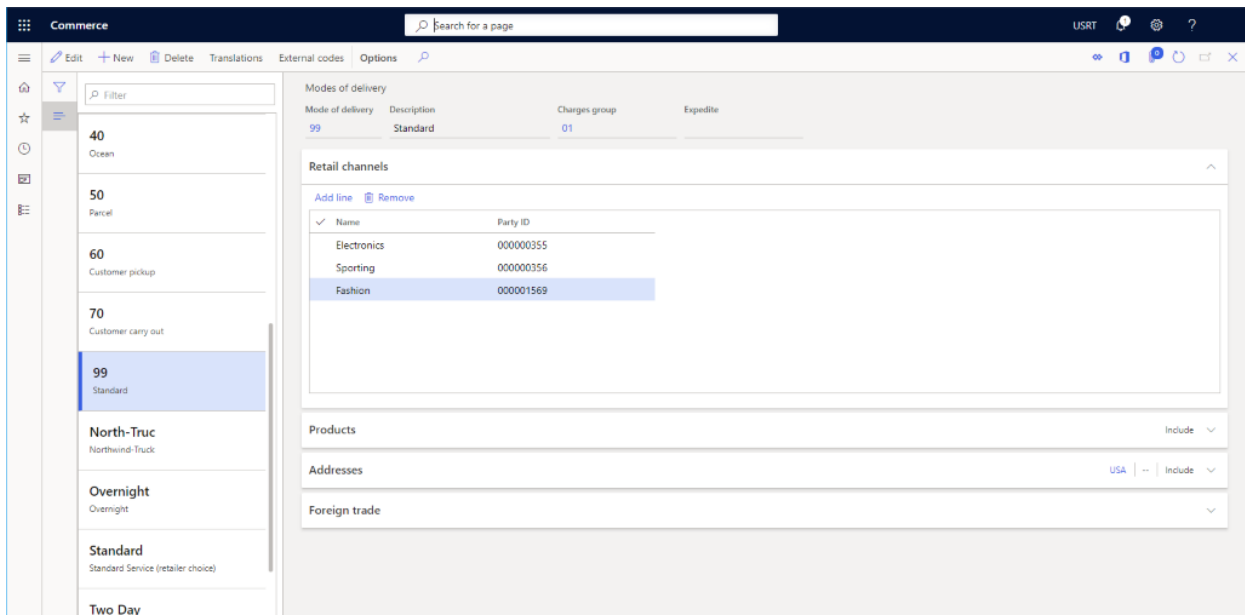
## Set up modes of delivery

You can see the configured modes of delivery by selecting **Modes of delivery** from the **Set up** tab on the **Action** pane.

To change or add a mode of delivery to be associated to the call center channel, follow these steps.

1. From the Call center modes of delivery form, select **Manage modes of delivery**
2. On the action pane, select **New** to create a new mode of delivery, or select an existing mode.
3. In the **Retail channels** section, click **Add line** to add the call center channel. Adding channels using organization nodes instead of adding each channel individually can streamline adding channels.
4. Ensure the mode of delivery has been configured with data on the **Products** FastTab and the **Addresses** FastTab. If no products or delivery addresses are valid for the mode of delivery, choosing it during order entry will result in errors.
5. After any changes have been made to the call center mode of delivery configurations, the **Process delivery modes** job must be run to explode the change matrix. This job can be found by navigating to **Retail and Commerce > Retail and Commerce IT > Process delivery modes**.

The following image shows an example of a mode of delivery.



### Set up channel users

To create a sales order that is linked to the call center channel from Commerce Headquarters, the user creating the sales order must be linked to the call center channel. The user cannot manually link a sales order created in Commerce Headquarters to the call center channel. The link is systematic, and is based on the user and the user's relationship to the call center channel. A user may only be linked to one call center channel.

1. On the action pane, select the **Channel** tab, and then select **Channel users**.
2. On the action pane, select **New**.
3. Choose an existing **User ID** from the dropdown selection list to link this user to the call center channel

After the channel user setup is done and the user creates a new sales order in Commerce Headquarters, the sales order will be linked to their associated call center channel. Any configurations for this channel will be applied systematically to the sales order. A user can confirm which call center channel the sales order is linked to by viewing the channel name reference on the sales order header.

### Set up price groups

Price groups are optional, but if used, can control which sales prices will be offered to customers placing orders in the call center channel. If a price group has not been configured for the customer, or if catalog price groups are not being applied to the sales order (using the **Source code ID** field on the call center order header), then the channel price group is used to locate item prices. If a price group is not found on the call center channel, the default item master prices are used.

To set up a price group, do the following.

1. On the action pane, click the **Channel** tab, and then select **Price groups**.

2. On the action pane, click **New**.
3. Select a **Retail price group** from the dropdown selection list.

## Additional resources

[Channel setup prerequisites](#)

[Call center sales functionality](#)

[Set up call center order processing options](#)

[Call center catalogs](#)

[Set up and work with fraud alerts](#)

[Set up continuity programs for call centers](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Call center catalogs

2/18/2021 • 10 minutes to read • [Edit Online](#)

This topic describes the call center–specific functionality linked to the catalog capabilities in Dynamics 365 Commerce.

The catalog features found in Commerce can be used for multiple purposes. Initially the catalog features were created to support third-party e-Commerce integrations. Catalog setup allowed companies to create a grouping of products and attributes that could be published externally for consumption by a third-party e-Commerce solution.

When call center channel support was added, the catalog concept was expanded to add additional capabilities for supporting and managing features related to traditional direct-to-consumer marketing catalogs. A direct-to-consumer company will often produce printed catalogs, which are then mailed to one or more segments of customers. These catalogs will typically have specific promotions or offers that will only be honored if the customer provides a catalog identification code at the time of order creation.

Direct-to-consumer marketing companies are very focused on tracking the response to these catalogs to ensure that the costs to produce and mail them are justified. To track the response, a code is traditionally printed on the back of the catalog and this code is then requested and applied when the catalog recipient calls to place an order by phone (or now more traditionally the code may be entered when the customer places an order online). While there are different industry terms that have been used to identify this catalog tracking code (including key code, promo code, catalog code, source code), we refer to the code in Commerce as the **Source code ID**.

## Basic catalog setup

Go to **Retail and Commerce > Catalogs and assortments > All catalogs** to configure your catalog.

When you create a new catalog, you must first link the catalog to one or more channels. This is done in the **Commerce channels** FastTab on the **Catalog setup** form. Click **Add** and select one or more channels. Only items linked to your selected channel **assortments** can be used when creating the catalog.

To add products to a catalog, a navigation hierarchy must be chosen. The navigation hierarchy will support the category structure for the catalog. You must pick from one of the navigation hierarchies linked to the channels selected on the **Commerce channels** FastTab of the **Catalog** page. If a navigation channel was not linked to a channel previously, go to **Retail and Commerce > Channel setup > Channel categories and product attributes** to link a navigation hierarchy default to each of your channels.

On the **Catalogs** menu tab, on the **Catalog setup** page, click **Add products** to configure the products to add to the catalog, or select a node in the navigation hierarchy (selecting a node will change the screen presentation and allow you to add products directly to a category within the catalog).

Click the top node of the catalog hierarchy to return to the main catalog header view. Configure effective and expiration dates as necessary on the **General** FastTab.

Before the catalog is available to use, it must be published. Click **Validate catalog** on the **Catalogs** menu to process a validation. This is required action and will validate that the required setup is accurate. Click **View results** to see the details of the validation. If errors are found, you must correct the data and run validation again until the validation has passed.

After validation is confirmed, click **Workflow** on the menu to start the approval workflow. Click **Submit** on the **Workflow** menu to execute the process. Configure the steps and authorized users for the workflow from **Retail and Commerce > Headquarters setup > Commerce workflows**. The workflow will define the steps

needed to get the catalog into an **Approved** status. When the catalog is in an **Approved** status, you can click the **Publish** option on the **Catalogs** menu to complete the process. After the catalog is in a **Published** status, it can be used in call center order entry and send catalog processes.

## Use catalogs to drive sales order pricing and promotions

A core reason for defining a catalog to use with a call center is to be able to configure specific prices and promotions for that catalog. Customers ordering from this catalog will receive these prices and promotions in call center order entry if the catalog's **Source code ID** is applied to the order header or lines.

To configure catalog-specific prices, select the **Price groups** option from the **Catalogs** tab to link one or more price groups to the catalog. All trade agreements, price adjustment journals, and advanced discounts (threshold, quantity, mix and match) that have been linked to the same price group will be applied when customers order from this catalog.

On the **Source codes** FastTab, click **Add** to add one or more **Source code ID** identifiers to this catalog. This is the code that will be applied during call center order entry to the sales order header (and lines). This code is used to link the sales order to the catalog and ultimately to the price groups and any special prices and promotions that have been configured.

## Use the source ID to track costs and response rates

When defining the **Source code ID**, you can optionally link this ID to a **Target market ID**. The **Target market ID** can be defined in **Retail and Commerce > Customers > Target market**. The target market is a list of customers and/or prospects that belong to a user-defined segment. Linking the customer or prospect data to the source code ID allows for better visibility into the recipients of the catalog. If a customer is linked to a target market and that target market is linked to an active source code ID/catalog, call center users will be able to see what catalogs a customer has received by selecting the **Source codes** menu option on the **Customers** menu tab on the **Customer service** page. During order entry, call center users can also see the specific catalogs a customer was sent in the **Source** drop-down list on the sales order header. Changing the filter from **All** to **Targeted** will allow the user to see the specific active catalogs the customer was sent. This is helpful in situations where the customer may have forgotten their catalog or can't locate or read the catalog code when they are calling in to create a sales order.

It's possible to link multiple source code IDs to a catalog. This is often needed when a company wants to track the response rate by different segments. The company will give a unique catalog code to different customer segments, which allows for tracking the response rate, down to the segment level, within a particular catalog event.

Selecting a particular **Source code ID** record and clicking the **Details** option on the **Source codes** FastTab will provide additional fields where sales projections, mailing costs, and mailing dates can be captured. This data is helpful for doing detailed analysis on the effectiveness of the catalog. Users can return to this page over time and use the **Source code analysis** and **Compare promotions** buttons to trigger analytical reports based on current sales data and compare costs and budget to actuals.

## Configure catalog-specific order and item scripts

When a call center user is creating a sales order, they can use on-screen scripts. These text-based scripts may provide additional information that the user should say to the customer, or it may be internal notes/reminders that the call center user should review and react to as they are creating the sales order.

It is often helpful to have different sets of scripts for different catalogs. On the **Scripts** FastTab, pre-defined scripts can be linked to a catalog. Use the **Timing** field to determine if the script will appear at the beginning of the order (as soon as the source code ID is entered on the order header), or at the end of the order (in the sales order summary form).



When selecting a node in the catalog's hierarchy and working with the data on the **Products** FastTab, users can also link scripts that are specific to catalogs or items using the **Scripts** action.

## Configure catalog-specific up-sell and cross-sell items

Linking up-sell/cross-sell suggestions to an item can be done from the products setup, but in some cases, a company may want to promote special up-sell/cross-sell items to customers ordering a specific product from a specific catalog. On the **Products** FastTab, select an item and click **Up-sell/cross-sell items** to configure products to be up or cross-sold to customers who purchase the selected item from the catalog. During call center order entry, catalog-specific up-sell/cross-sell items will appear on the screen instead of standard up-sell/cross-sell products that may have been configured for that item through the usual product configuration.

Up-sell/cross-sell items can also take advantage of the script features to show specific messages that a user will see when the up-sell/cross-sell item is displayed during order entry. Scripts tied to up-sell/cross-sell products configured specifically for a catalog product will only appear when that catalog's source code is applied in call center order entry.

## Catalog page analysis

On the **Catalogs** tab, options are available to configure **Catalog pages**. This feature allows you to define specific pages and page types for the printed catalog and their associated costs.

When configuring the products in the catalog, use the **Product page layout** action to define the specific pages, percentage of page, and position of page details for the item. Configuring this data will allow users to take advantage of the **Catalog area analysis report**. This report is found by navigating to **Retail and Commerce > Call center reports > Catalog area analysis** report. This report analyzes sales placed against the catalog (sales orders where the source ID for the catalog was tied to the order header or line) and their associated percent of page and costs to give a traditional direct marketing **Square inch analysis** report.

## Catalog requests

As catalogs are configured and published in Commerce, the **Send catalog** feature can be utilized. This feature is available on the **Customer search** and **Customer service** pages. After selecting a customer record through **Customer search** or while viewing a selected customer's account from **Customer service**, users may select the **Send catalog** option which will open a dialog box allowing the user to choose from a list of any published and active catalogs. A user can select a catalog and a quantity, and a particular source code ID to send. When they click the **Send** button, a request is stored which can then be managed by printing the **Catalog requests** report. This report is found by navigating to **Retail and Commerce > Call center reports > Catalog requests** report. It lists all the catalog requests, including the customer name and address details of the customer who requested the catalog. This report can be used internally or the data can be transmitted to a third-party supporting external processes for physically sending the catalog to the customer.

## Additional features

On the **Catalogs** tab, options for configuring a **Payment schedule** and **Free products** are also available. If the source code ID linked to the catalog is applied during call center order entry, the customer will be eligible for the free products or use of the specific catalog payment schedules as defined. If it's necessary to limit the customer to only being able to select from payment schedules linked to their catalog and not all active payment schedules in the system, the **Only allow catalog plans** check box can be selected for one or more of the source code IDs defined to enforce that limitation.

## Additional notes

Currently, when a source code ID is applied to a sales order in call center, it is used to drive prices, promotions,

scripts and up-sell/cross-sell's that are catalog specific. The system will not prohibit or prevent a product that is not in the catalog from being ordered on the sales order. If an item is ordered that is not part of the catalog, the system will first use the **Price group** that is defined on the call center channel (**Retail and Commerce > Channels > Call centers > All call centers**) for item price or promotions. If no specific channel price is found, the base selling price of the item will be used.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up call center channels

2/18/2021 • 7 minutes to read • [Edit Online](#)

A company can define multiple call center channels in Dynamics 365 Commerce. Call center channels are configured at **Retail and Commerce > Channels > Call centers > All call centers**, and they are specific to a legal entity.

When a new call center channel is created, it's systematically assigned an operating unit number. Because call centers are created as operating units, users can link the call center channel to various Commerce features, such as assortments, catalogs, and specific modes of delivery.

A default warehouse can be configured on the call center channel. Then, when sales orders are created in that channel, the default warehouse is automatically entered on the sales order header, unless another warehouse has been defined on the customer that is selected for the sales order. In that case, the customer's warehouse is entered by default.

Users must be linked to a call center channel to use the features of call center. Any sales order that a user creates is automatically linked to that user's call center channel. Currently, a single user may not be linked to multiple call center channels at the same time.

An email notification profile can also be configured on the call center channel. The profile defines the set of email templates that is used when email is sent to customers who place orders through the call center channel. The email triggers can be configured against system events, such as order submission or order shipment.

Before sales can be correctly process through a call center channel, correct [payment methods](#) and delivery modes must be defined for the channel.

At the level of the call center channel, you can define other default values that are related to the financial dimensions that will be linked to orders that are created by that channel.

## Options for order processing behavior

Three settings in the configuration of a call center have a major effect on the features and functions that are available for sales orders that are created against that call center: **Enable order completion**, **Enable direct selling**, and **Enable order price control**.

### Enable order completion

The **Enable order completion** setting on the call center channel has a major effect on the order processing flow of sales orders that are entered for that channel. When this setting is turned on, all sales orders must go through a set of validation rules before they can be confirmed. You run these rules by selecting the **Complete** button that is added on the Action Pane of the sales order page. All sales orders that are created when the **Enable order completion** setting is turned on must go through the order completion process. This process enforces the capture of payment and payment validation logic. In addition to payment enforcement, the order submission process can trigger [fraud checks](#) that you configure in the system. Orders that fail payment or fraud validations are put on hold and can't be released to further processing (such as picking or shipping) until the issue that caused the hold is resolved.

When the **Enable order completion** setting is turned on for the call center channel, if line items are entered on a sales order and the channel user tries to close or navigate away from the sales order form without first selecting **Complete**, the system enforces the order completion process by opening the sales order recap page and requiring that the user correctly submit the order. If the order can't be correctly submitted together with payment, the user can use the [order holds](#) functionality to put the order on hold. If the user is trying to cancel

the order, he or she must correctly cancel it by using either the Cancel function or the Delete function, depending on the function that the user's security allows.

If the **Enable order completion** setting is turned on for the call center channel, the **Payment status** field will be tracked on the order. The system calculates the **Payment status** when the sales order is submitted. Only orders that have an approved payment status are allowed to move through the system for additional order processing steps, such as picking and shipping. If payments are declined, the **do not process** flag will be enabled on the detailed order status, this puts the order on hold until the payment issue is resolved.

Additionally, if the **Enable order completion** setting is turned on, when users create sales orders and are in line item entry mode, the **Source** field will be available on the main sales order header. The **Source** field is used to capture a [catalog source code](#) in a direct marketing selling scenario. This code can then drive special prices and promotions.

Even if the **Enable order completion** setting is turned off, users can still apply a source code to a sales order. However, they must first open the sales order header details to access the **Source** field. In other words, some additional clicks are required. The same behavior applies to features such as ship complete and expedited orders. These features are available for all orders that are created in the call center. However, when the **Enable order completion** setting is turned on, users can see the configuration of these features on the sales header while they are in the line entry view. They don't have to drill into the sales order header details to find the appropriate settings and fields.

### **Enable direct selling**

If the **Enable direct selling** setting is turned for the call center channel, users can take advantage of the upsell and cross-sell features of Commerce. In this case, pop-up windows appear during order entry and suggest other products that the call center user can offer to the customer. The products that are suggested are based on the product that was just ordered on the sales order line. Currently, the upsell and cross-sell suggestions are configured at the item level on products or catalogs. If the **Enable direct selling** setting is turned off for the call center channel, pop-up windows don't appear during order entry, even if a valid upsell or cross-sell was defined for an item that is being ordered.

When the **Enable direct selling** setting is turned on, the scripts and images features of the sales order entry page are also turned on. In this case, an information panel is available on right side of the page during order entry. This panel can show scripts that are related to the generic order entry process, the catalog source code that was applied, or scripts that are related to the items that are being ordered. Additionally, the images panel can show a product image for the items that are being ordered, if an image has been defined for the item in the product setup.

### **Enable order price control**

When the **Enable order price control** setting is turned, only authorized users can change the sales price of an item during order entry. The changes must be within defined tolerances. Users who don't have the proper authorization must submit a request for a price change instead. The request will then be processed through system workflows for review and approval.

## **Channel users**

When you define the call center channel, you must link channel users to the call center. Otherwise, the call center can't be used in the system. When users sign in to Commerce and enter sales orders or return orders on a page that is related to order entry, their user ID is validated against the configuration of the call center channel. If a user is linked to a specific call center channel, orders that the user creates inherit the traits and default values of that channel.

By default, the **Sale** flag on the sales order header is turned on for all orders that call center users create. The orders can then take advantage of the system's commerce-specific price and promotions features.

Users who aren't linked to a call center channel use the standard order entry features of Microsoft Dynamics 365 Finance. Orders that these users enter through the sales order entry form will not be systematically identified as Commerce orders. Additionally, these orders entered by these users will not be subject to any of the order completion processing rules, pricing logic, or other order validations that can be defined in the call center channel configuration or call center system parameters.

After you've finished configuring the call center channel and defining channel users, to help ensure the desired system behavior, make sure that all required Call center parameters are defined at **Retail and Commerce > Channel setup > Call center setup > Call center parameters**. Make sure that related number sequences are also defined.

**NOTE**

To use call center functionality, the configuration key for **Multiple ship-to** must be enabled. This configuration key can be found in the **Trade configuration** keys under **System Administration > Setup > License Configuration**. This is required due to call center functionality that performs various validations based on the delivery address configured at the sales order line level.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure call center delivery modes and charges

2/18/2021 • 9 minutes to read • [Edit Online](#)

When a sales order is placed in Dynamics 365 Commerce, if the person who entered the sales order is linked to a call center channel, logic and rules are used to validate the mode of delivery (delivery mode) and calculate charges for the order.

When you create a sales order, you can select a delivery mode on the sales order header and the sales order lines. By default, the delivery mode that you select on the header is used for all sales order lines. However, you can override the default delivery mode on individual sales lines as you require. You can also define a delivery mode on a customer record. Then, when orders are created for the customer, that delivery mode is used by default on the sales order header.

Commerce has capabilities that let users limit the delivery modes that can be used by a channel, the delivery modes that can be used for a product, and the delivery modes that are valid for specific shipping destinations. Charges can also be defined so that additional fees are added to a customer's order, based on the delivery modes that are selected for the sales order and the total order value.

## Define delivery modes

Before you specify which delivery modes can be used for call center orders, and define the associated rules and charges, you must define the delivery modes. Go to **Sales and marketing > Setup > Distribution > Modes of delivery**. Select **New** to create a new delivery mode. Alternatively, select an existing delivery mode in the list, and then select **Edit** to make changes.

In the **Mode of delivery** field, you can enter any combination alphanumeric characters, based on your business requirement. You can then use the **Description** field to provide additional context. The **Charges group** and **Expedite** fields are optional and will be explained in more detail later in this topic.

On the **Commerce channels** FastTab, add any channel that should be allowed to use the delivery mode when sales transactions are created in that channel.

On the **Products** FastTab, you can specify which products and/or product categories the delivery mode can and can't be used for. For example, if a product can't be shipped by air because of hazardous material (hazmat) restrictions, make sure that the product or product category is excluded from all delivery modes that involve air transportation.

On the **Addresses** FastTab, you can specify which countries or regions, or states, the delivery mode can and can't be used for. For example, orders that are shipped to Hawaii or Alaska aren't eligible for ground delivery. Therefore, these states should be excluded from any delivery mode that is associated with a ground delivery service but included in any delivery mode that is associated with an air delivery service.

## Validate delivery modes for a call center order

After the delivery modes are defined, you must run the **Process delivery modes** batch job. This job makes the delivery modes available so that they can be used in sales order processes for channels. To run the **Process delivery modes** job, go to **Retail and Commerce > Retail and Commerce IT > Process delivery modes**. This job should be run any time that new delivery modes are added to a channel or changes are made to existing delivery mode/channel relationships.

After you run the **Process delivery modes** batch job, you can go to **Retail and Commerce > Channels > Call centers > All call centers**. On the **All call centers** page, on the Action Pane, on the **Set up** tab, select

**Modes of delivery.** The **Modes of delivery** page lists all the valid delivery modes for the selected call center channel. To edit existing delivery modes or add new delivery modes, select **Manage modes of delivery**. Note that the **Process delivery modes** job must be run whenever changes are made.

## Define charges for delivery services

When sales orders are created for customers, a company might want to add charges that are automatically calculated based on the delivery modes that are selected for the order. These charges can be configured so that they are the same for all customers and delivery modes. Alternatively, the charges can vary, depending on the customer and/or the delivery modes that are selected for the sales order.

To define the charges, go to **Retail and Commerce > Channel setup > Charges > Auto charges**. Select **New** to add new charges. Alternatively, select an existing entry, and then select **Edit**.

Charges can be defined so that they are calculated at the level of either the order header or the order lines. Use the **Level** field to select the level desired.

Charges can be defined for a specific customer, a group of customers, or all customers. In the **Account code** field, select **Table** to define charges that are applied only to a specific customer. Select **Group** to define charges for a specific customer group. Select **All** to apply the charges to every customer who places a sales order that uses the related delivery mode. If you selected **Table** or **Group** in the **Account code** field, select the customer or customer group in the **Account relation** field.

Charges can be configured so that they are applied for a specific delivery mode, a delivery mode group, or all delivery modes. If you select **Table** in the **Mode of delivery code** field, you must select a specific delivery mode in the **Mode of delivery relation** field. If you select **Group**, you must select a delivery mode group in the **Mode of delivery relation** field. Delivery mode groups are defined at **Retail and Commerce > Channel setup > Charges > Delivery charges group**. They can then be linked to one or more delivery modes on the **Modes of delivery** page. If you select a group when you define charges, any delivery mode that is linked to the selected delivery group uses those charges. Finally, if you select **All** in the **Mode of delivery code** field, all delivery modes use the charges. Therefore, you don't select a value in the **Mode of delivery relation** field.

In the **Lines** section, you can define one or more charges by currency, as you require. Charges must be linked to a charges code that defines the financial posting rules for the charge. The **Category** field is used to define how charges are calculated. For example, if customers should be charged a flat rate of \$9.95 to have an order shipped by a specific delivery mode, use the **Fixed** category. If the business decides to charge customers a percentage of the order total to cover the delivery charges, use the **Percent** category. The actual charge to the customers is defined in the **Charges value** field.

Companies often configure tiered charges. In this case, the amount that customers pay for delivery is based on the order value. To configure tiered charges, enter values in the **From amount** and **To amount** fields in addition to defining the charge itself in the **Charges value** field. For example, for orders that have a value that is less than \$50, a retailer charges \$5.95 for ground shipping. For orders that have a value that is equal to or more than \$50, but less than \$100, the retailer charges \$7.95. Finally, for orders that have a value that is equal to or more than \$100, the retailer provides free shipping. The following illustration shows the configuration of these charges.

Auto charges

Account code: All | Account relation: | Mode of delivery code: Table | Mode of delivery relation: 20

Lines

+ Add | Remove

✓	Currency	Charges code	Category	Charges value	Charges currenc...	From amount	To amount	Sales tax group	Keep
	USD	FREIGHT	Fixed	5.95	USD	0.01	49.99		<input type="checkbox"/>
	USD	FREIGHT	Fixed	7.95	USD	50.00	99.99		<input type="checkbox"/>
	USD	FREIGHT	Fixed	0.00	USD	100.00	9,999,999.00		<input type="checkbox"/>

You can use a mixture of categories for charges, depending on your business requirements. For example, for all orders that have a value that is less than \$100, there is a fixed charge of \$9.95 for shipping. Then, for orders that have a value that is equal to or more than \$100, delivery charges are calculated at a rate of 5 percent of the order value. The following illustration shows the configuration of these charges.

Auto charges

Account code: All | Account relation: | Mode of delivery code: Table | Mode of delivery relation: 20

Lines

+ Add | Remove

✓	Currency	Charges code	Category	Charges value	Charges currenc...	From amount	To amount	Sales tax group	Keep
	USD	FREIGHT	Fixed	9.95	USD	0.01	99.99		<input type="checkbox"/>
	USD	FREIGHT	Percent	5.00	USD	100.00	999,999.00		<input type="checkbox"/>

## Apply delivery modes during order entry in a call center

When a new sales order is created, a value must be specified in the **Mode of delivery** field on the **Delivery** FastTab of the sales order header. This field might be filled in automatically, based on default values from the customer record.

The delivery mode that is defined on the order header is automatically copied to the sales order lines as they are created. However, you can change the delivery mode setup for a specific line item on the **Delivery** tab in the **Line details** section of the sales order entry page.

If the selected delivery mode isn't valid for the product or the delivery address that is defined for the order or order line, you receive an error message. You must then select a delivery mode that has been defined to support that product or address configuration.

## Calculation of delivery charges during entry of order

If the **Enable order completion** setting is turned on for your call center channel, shipping charges are automatically calculated for sales orders when users select **Complete**. The following message appears at the top of the **Sales order summary** page: "Tiered charges calculated." The charges that are calculated are added to the value of the **Sales total** field. On the **Amount** FastTab, the **Charges** field shows the total amount of all charges that have been calculated for the order and lines. To see a more detailed breakdown of the charges, select **Order** on the **Sales order summary** page, and then select the **Charges** option to view, add, or edit the charges. Note that the calculation of delivery charges on the order header is based on the delivery mode that is linked to the header. Line-level delivery charges are calculated based on the delivery mode that is configured for the sales line. If multiple delivery modes are used on different lines, multiple charges might be applied and added together. The total amount is then shown in the **Charges** field on the **Sales order summary** page.

If the **Enable order completion** setting is turned off, users must manually trigger the calculation of charges. On the **Sales order** page, on the Action Pane, on the **Sell** tab, in the **Calculate** group, select **Tiered charges**.



The "Tiered charges calculated" message appears. You can then select the **Charges** option on the **Sell** tab to view, edit, or delete the calculated charges.

## Use expedited delivery modes on call center orders

You can optionally link an expedite code to any delivery mode that you configure. This code is used as a prioritization sorting and reporting tool. It doesn't currently cause additional fees to be applied to the order. To set up expedite codes, go to **Sales and marketing > Setup > Distribution > Expedite codes**.

For example, for orders that will be shipped by next-day air, picking must be done in the warehouse by 1 PM every day. In this case, an expedite code can be created, and that code can be linked to any next-day delivery mode that is configured in the system. When the warehouse creates its pick wave, the appropriate expedite code in the **Expedite** field can be used as a filter, so that picking is run only for orders that have delivery modes that are linked to that code.

Additionally, when a call center order is entered, an expedite code can be manually applied either to the sales order header or to an individual sales order line. Again, the code can be used for sorting or reporting purposes. Sometimes, an order must be handled carefully because of a customer service issue. In this case, a specific expedite code can be applied to the order header or lines to help identify and prioritize the order during the fulfillment process.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up and work with call center fraud alerts

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic explains how to set up criteria and rules to put potentially fraudulent sales orders on hold for further review. The fraud check feature is used to determine the validity of the information in a sales order. If the information in the sales order appears to be questionable, based on an organization's fraud criteria and rules, the order can be put on hold for further review. In this case, the order can't be released to the warehouse for further processing until the hold has been cleared.

## NOTE

This feature can be used only with sales order processing for the Commerce call center channel.

## Turning on the fraud check feature

To use the fraud check feature, you must set the **Enable order completion** option on the channel to **Yes** when the call center channel is [defined](#). When order completion is turned on, call center users must select **Complete** on the sales order page for all sales orders that are created. The Complete action causes the **Sales order summary** page to open. After users enter the required payment data on the **Sales order summary** page, they select **Submit** to finalize the order. When the order is submitted, the fraud check feature is triggered, and any rules that are active in the system are automatically validated.

Call center users can also manually put sales orders on hold for fraud review before they select **Submit**. To manually put a sales order on hold, on the **Sales order summary** page, select **Hold > Manual fraud hold**. You're then prompted to enter a comment to explain your reason for putting the order on hold. This comment will appear in the [order holds](#) workbench to provide context to the user who reviews orders that are on hold to determine whether the order should be released.

In addition to configuring the **Enable order completion** option on the channel, you must configure the fraud check feature in the Call center parameters. Go to **Retail and Commerce > Channel setup > Call center setup > Call center parameters**. On the **Call center parameters** page, on the **Holds** tab, set the **Fraud check** option to **Yes**.

On the **Holds** tab, you should also define the [hold codes](#) that will be applied to an order that is either manually or automatically put on hold for fraud review. Set the hold codes in the **Manual fraud hold code** and **Fraud hold code** fields. You might find it helpful to create two unique hold codes, so that users who work in the holds workbench can easily filter and distinguish automatic holds from manual holds.

For the fraud check feature to work effectively, you must also set the **Minimum score** field. Every fraud criterion and rule that is defined in the system has a score. When a sales order is checked for fraud matches, if one or more matches are found, the scores are added together to give the order a total fraud score. If the total fraud score for an order exceeds the value of the **Minimum score** field, the order is automatically put on hold. You can optionally use the other score-related fields on the **Holds** tab to define the email score, phone score, ZIP/postal code score, and extended ZIP/postal code score. If you don't specify a score for any of these static fraud criteria when you define them on the **Static fraud data** page, the system will score them by using the default scores that you specify on the **Holds** tab of the **Call center parameters** page.

Finally, use the **Fraud comment type** field to specify the document type that should be used when users enter comments when they manually put an order on hold for fraud review. Most often, this field is set to **Note**.

# Defining fraud criteria and rules

The system references two types of fraud criteria to determine whether an order should be put on hold for fraud review:

- **Static fraud data** uses a specific value, such as a phone number that has been put on a list of blocked numbers or an email address that has been flagged because it's known to have been used for previous fraudulent transactions. To set up static fraud data, go to **Retail and Commerce > Channel setup > Call center setup > Fraud > Static fraud data**. On the **Static fraud data** page, you can add fraud criteria manually or through data import. Scores are attached to the fraudulent information. If the fraud check feature is turned on, every sales order that is entered is compared to the static data. If the data is found in either the customer's billing address or the delivery address that is linked to the order header, or if the data is found in the delivery addresses that are linked to any of the lines on that sales order, the scores of all unique matches are added together and compared to the **Minimum score** value to determine whether the order should be put on hold.
- **Fraud rules** consist of user-defined variables and the conditions that are defined for those variables. To create rules, go to **Retail and Commerce > Channel setup > Call center setup > Fraud > Rules**. Fraud rules let a company configure a more complex rule set that can include **AND** or **OR** statements to evaluate multiple conditions. For example, a user wants all orders for customers who belong to a specific customer group and who ordered a specific product to be put on hold for fraud review. In this case, conditions to validate the customer and products are defined on the **Rules** page, and an **AND** condition is used. An order is then put on hold only if both conditions are true, and if the score value that is assigned to this rule, plus the score value of any other rules that the order matches, causes the order's total fraud score to exceed the **Minimum score** value that is defined on the **Call center parameters** page.

## NOTE

Multiple rules or overly complex rules will affect system performance when sales orders are submitted. The fraud check feature hasn't been optimized to handle a large volume of static fraud data entries and many active rules. Remember that every rule is evaluated when call center users select **Submit** during sales order entry. The rules are evaluated against the sales order header and all order lines. The more rules there are and the more complex the rule statements are, the more time will be required for processing. If there are many line items on an order, and many active rules and static data entries, the automatic process of reviewing and validating all the data and calculating a fraud score can have a severe impact on performance. Organizations that use this feature should always test and confirm that the processing time for order submission is acceptable before they apply any changes to rules or static fraud criteria to the production environment.

# Identifying orders that are on hold for fraud review

When call center users submit a sales order, if the order matches the fraud criteria or rules, and if the score exceeds the minimum, the users receive a warning message that states that the order has been put on hold. Users can close this message, because it's for informational purposes only. Users can optionally communicate this information to the customer. The business should determine the protocol that users follow in this situation.

The order is saved, but the **Do not process** flag is set on it. This flag helps guarantee that the order can't be released to the warehouse. At any time, users can view the setting of the **Do not process** flag for any sales order on the **Detailed status** page. This page can be opened from the **All sales order** and **Customer service** pages. The system also updates the value of the **Detailed status** field for the order to **Fraud hold**.

To view and manage the orders that are on hold for fraud review, go to **Retail and Commerce > Customers > Order holds**. On the **Order holds** page, select an entry in the list, and then click **Order hold** to see a more detailed view that includes information about the reason for the hold. On the **Fraud details** FastTab, you can view the systematic fraud criteria that were found to be a match for the order and the scores that were applied. If the order was put on manual hold, you can review any comments that were entered by the user who put the

order on hold by looking at the **Fraud notes** section on the **Notes** FastTab.

For more information about how to work with hold orders, see [Order holds](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add a channel to an organizational hierarchy

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add a channel to an organizational hierarchy in Microsoft Dynamics 365 Commerce.

## Overview

Channels need to be associated with one or more organizational hierarchies. Before creating channels, you need to confirm that your organizational hierarchies have been set up.

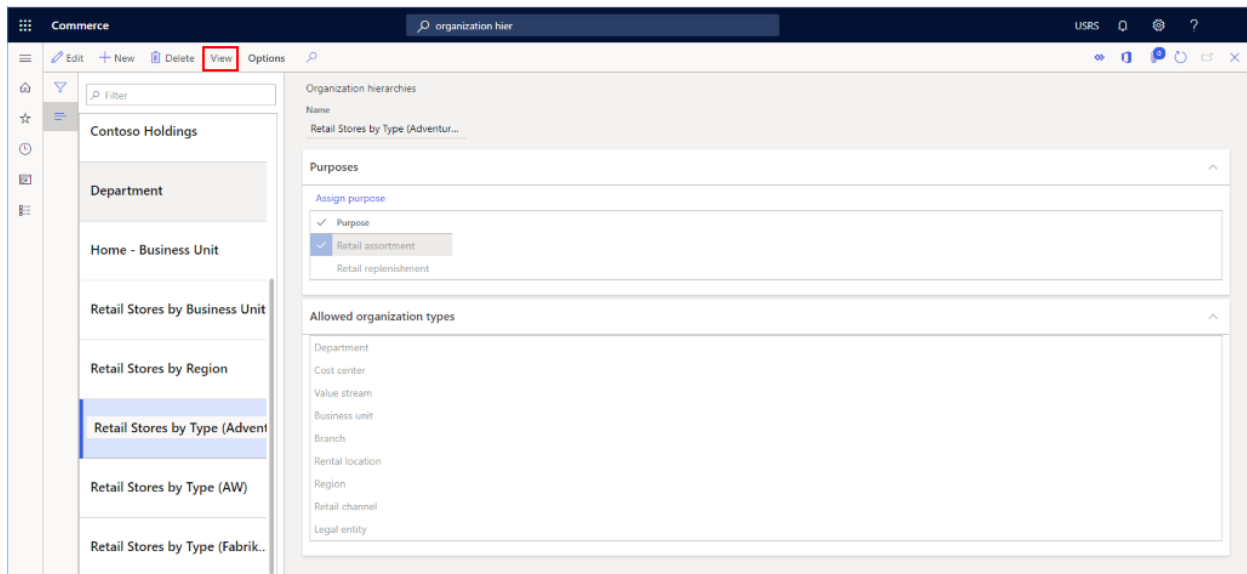
See [Organizational hierarchies](#) for more details on how to create organizational hierarchies.

## Select a hierarchy

To select a hierarchy, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Channel Setup > Organization hierarchies**.
2. From the list, select the organization hierarchy that you'll be adding the channel to.
3. On the action pane, select **View** to view hierarchy details.

The following image shows organizational hierarchy details for the selected hierarchy.

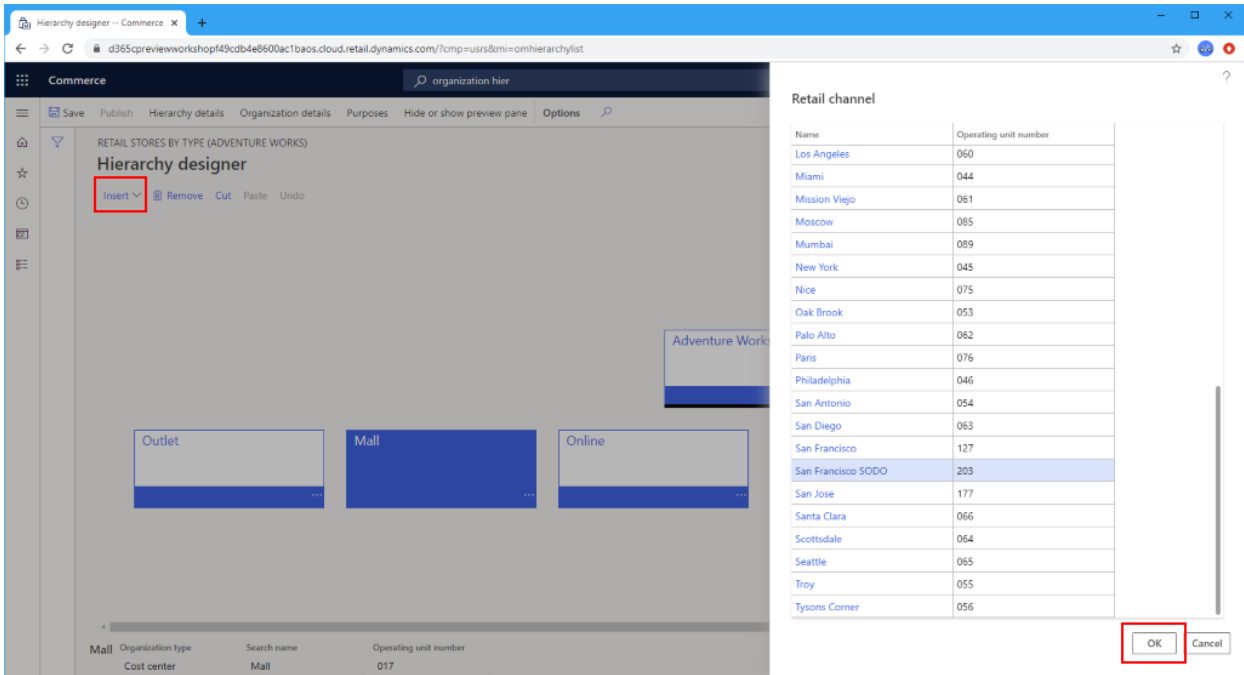


## Add a channel to a hierarchy node

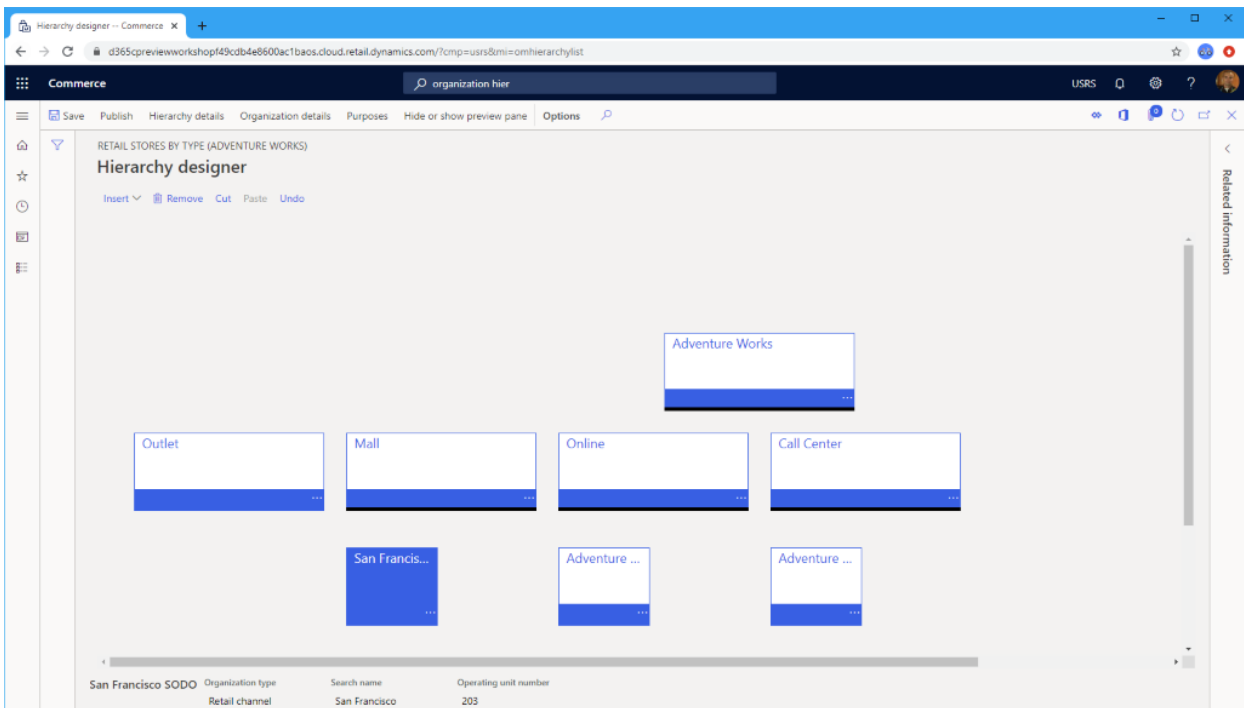
To add a channel to a hierarchy node, follow these steps.

1. On the action pane, select **Edit**.
2. Select the hierarchy node you want the channel added to, then from the **Insert** drop-down list, select **Retail Channel**.
3. Select the channel to add, then select the **OK** button.
4. On the action pane, select **Save**.
5. On the action pane, select **Publish** and provide an **Effective date** in the past to have this action go into effect immediately.

The following image shows how to select a channel to add to a hierarchy node.



The following image shows a hierarchy with various channels added.



## Additional resources

[Channels overview](#)

[Channel setup prerequisites](#)

[Organizations and organizational hierarchies overview](#)

[Plan your organizational hierarchy](#)

[Organization hierarchies](#)

[Set up a retail channel](#)

[Set up an online channel](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Omni-channel advanced auto charges

2/18/2021 • 17 minutes to read • [Edit Online](#)

This topic provides information on configuration and deployment of the advanced auto charges features that are available in Dynamics 365 for Retail version 10.0.

When the advanced auto charges features are enabled, orders created in any supported Commerce channel (point of sale (POS), call center, and online), can take advantage of the [auto charges](#) configurations defined in the ERP application for both header and line-level related charges.

In releases prior to Retail version 10.0, [auto charge](#) configurations are only accessible by orders created in e-Commerce and call center channels. In versions 10.0 and later, POS-created orders can leverage the auto charges configurations. That way, additional miscellaneous charges can systematically be added to sales transactions.

When using releases prior to version 10.0, a POS user is prompted to manually enter a shipping fee during the creation of a "ship all" or "ship selected" POS transaction. While the miscellaneous charges capabilities of the application are utilized in respect to how the charges are written to the order, no systematic calculation is provided – the calculation relies on the user's input to determine the value of the charges. The charges can only be added as a single "shipping" related charges code and cannot easily be edited or changed in the POS after they are created.

The use of manual prompts to add shipping charges is still available in versions 10.0 and later. If an organization does not enable the **Advanced Auto-charges** parameter, the POS prompts for manual entry of charges will remain the same.

With the advanced auto charges feature, POS users can have systematic calculations for any defined miscellaneous charges based on auto charges setup tables. In addition, users will have the ability to add or edit an unlimited number of additional charges and fees to any POS sales transaction at the header or line-level (for a cash and carry or customer order).

## Enabling advanced auto charges

On the **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters** page, go to the **Customer orders** tab. On the **Charges** FastTab, set **Use advanced auto-charges** to **Yes**.

The screenshot shows the Dynamics 365 Commerce parameters page. The breadcrumb navigation is: Dynamics 365 > Finance and Operations > Commerce > Headquarters setup > Parameters > Commerce parameters. The page title is "Commerce parameters". The left sidebar contains a navigation menu with categories: General, Posting, Customer orders (selected), Replenishment, Discounts, Number sequences, POS authentication, Channel deployment, Configuration parameters, Workforce management, Electronic documents, and POS search criteria. The main content area is titled "Set up the parameters for customer orders". It is divided into several sections: "Order" (Default order type: Sales order, Default deposit percentage: 100.00, Disposition code: 71, Days quotation expires: 30, Process return orders as sales o...: No, Enable returns for multiple orders: No, MODE OF DELIVERY: Pickup mode of delivery: 40, Carry Out mode of delivery: 30, Electronic mode of delivery: 12), "Charges" (Cancellation charge percentage: 5.00, Shipping charge code: FREIGHT, Cancellation charge code: Cancel, Use advanced auto-charges: Yes), "Payments" (Terms of payment: CreditCard, Method of payment: CRED), and "Refund shipping charges" (Refund shipping charges: Yes, Maximum amount without approval: 0.00).

When advanced auto charges are enabled, users are no longer prompted to manually enter a shipping charge at the POS terminal when creating a ship-all or ship-selected customer order. POS order charges are systematically calculated and added to the POS transaction (if a corresponding auto charges table that matches the criterion of



the order being created are found). Users can also add or maintain header or line-level charges manually through newly added POS operations that can be added to the POS screen layouts.

When advanced auto charges are enabled, the existing **Commerce parameters** for **Shipping charges code** and **Refund shipping charges** are no longer utilized. These parameters are only applicable if the **Use advanced auto charges** parameter is set to **No**.

Before you enable this feature, ensure that you have tested and trained your employees, as the enabled feature will change the business process flow of how shipping or other charges are calculated and added to POS sales orders. Make sure that you understand the impact of the process flow to the creation of transactions from POS. For call center and e-Commerce orders, the impact of enabling advanced auto charges is minimal. Call center and e-Commerce applications will continue to have the same behavior they have had historically related to the auto charges tables to calculate additional order fees. Call center channel users will continue to have the ability to manually edit any system calculated auto charges at the header or line level, or manually add additional miscellaneous charges at the header or line level.

## Additional POS operations

For advanced auto charges to work properly in your POS application environment, new POS operations have been added. These operations must be added to your [POS screen layouts](#) and deployed to the POS devices as you deploy advanced auto charges. If these operations are not added, users will not be able to manage or maintain miscellaneous charges on the POS transactions and will have no way of adjusting or changing the charges values that are systematically calculated based on auto charges configurations. At minimum, it is suggested that you deploy the **Manage charges** operation to your POS layout.

The new operations are as follows.

- **142 - Manage charges** – Use this operation to allow POS users to view and edit miscellaneous charges for the POS transaction that were either added manually or systematically through auto charges calculations.
- **141 - Add header charges** – Use this operation to give the user the ability to manually add a header-level miscellaneous charge to any POS sales transaction (and select the charges code to be used).
- **140 - Add line charges** – Use this operation to give the user the ability to manually add a line level miscellaneous charge to any POS sales transaction line (and select the charges code to be used).
- **143 - Recalculate charges** – Use this operation to perform a full recalculation of the charges for the sales transaction. Any previously user-overwritten auto charges will be recalculated based on the current cart configuration.

As with all POS operations, security configurations can be made to require manager approval in order to execute the operation.

It is important to note that the above listed POS operations can also be added to the POS layout even if the **Use advanced auto-charges** parameter is disabled. In this scenario, organizations will still get added benefits of being able to view manually added charges and edit them using the **Manage charges** operation. Users may also use the **Add header charges** and **Add line charges** operations for POS transactions even when **Use advanced auto-charges** parameter is disabled. The **Recalculate charges** operation has less functionality if used when **Use advanced auto-charges** is disabled. In this scenario, nothing would be recalculated and any charges manually added to the transaction would just reset to \$0.00.

## Use case examples

In this section, sample use cases are presented to help you understand the configuration and usage of auto charges and miscellaneous charges within the context of channel orders. These examples illustrate the behavior of the application when the **Use advanced auto-charges** parameter has been enabled.

### **Auto charges header charges example**

## Use case scenario

A retailer wants to automatically add charges for freight when transactions are created in any Commerce channel that require a shipment of products to the customer. The retailer offers two methods of delivery: Ground and Air. If a customer chooses Ground delivery and the order value is less than \$100, the retailer wants to charge the customer a freight charge of \$10.00. If the order is over \$100 in value and the customer chooses ground shipping, the customer will not be charged any additional freight fees. If the customer chooses the Air method of delivery for all orders, regardless of their total value, will be charged a freight fee of \$20.00.

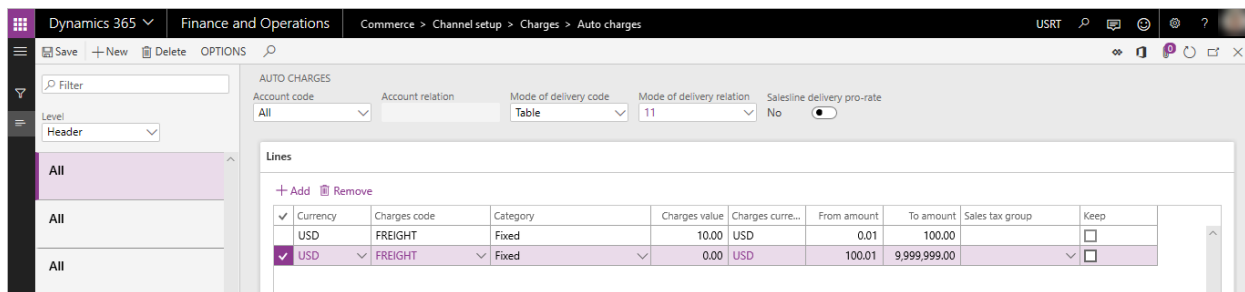
## Setup and configuration

This scenario requires the configuration of two auto charges tables.

Go to **Accounts receivable > Charges setup > Auto charges**.

Configure two different header-level auto charges. Configure one for the "Ground mode" of delivery and one for the "Air mode" of delivery. For this scenario, configure them to be used for "All customers".

For the ground delivery charges, in the lines section of the **Auto-charges** page, define a charge that will be applied for orders between \$.01 and \$100 as \$10.00. Create another charges line to indicate orders over \$100.01 will have no charges.



For the air delivery charges, in the lines section of the auto charges form, define a charge of \$20.00 that will be applied to all orders (between a value of \$.01 to \$9,999,999).

Send the changes to the Commerce Scale Unit/Channel DB so that the POS can utilize them by running the **1040 distribution schedule job**.

## Sales processing for this scenario

After the configuration steps above are complete and the changes have been applied to the channel database, any customer order or sales transaction created in the POS, call center, or e-Commerce channels that have the ground or air delivery methods set at the header level will utilize these charges and automatically apply them to the sale.

At this time, the charges will apply to all sales transactions created within the legal entity that utilize these delivery modes, as there is no functionality to designate that an auto charge configuration will only apply to a specific selling channel.

For POS and e-Commerce scenarios, because there is no clearly defined "header" on these orders, header-level charges will only apply if all sales lines on the transaction are set to ship with the exact same mode of delivery. If there are "mixed-modes" of fulfillment on the transactions created by POS or e-Commerce, only line-level auto charges will be considered and applied.

In call center scenarios, the user has control over the setting of the delivery mode at the order header, therefore header-level charges will apply for these orders even if some of the sales lines have been configured to use a different mode of delivery. Header-level charges for call center orders will always be based on the mode of delivery that is defined at the order header level of the sales order.

## Auto charges line charges example

### Use case scenario

A retailer wants to add an additional charge to the customer for setup fees when the customer purchases a

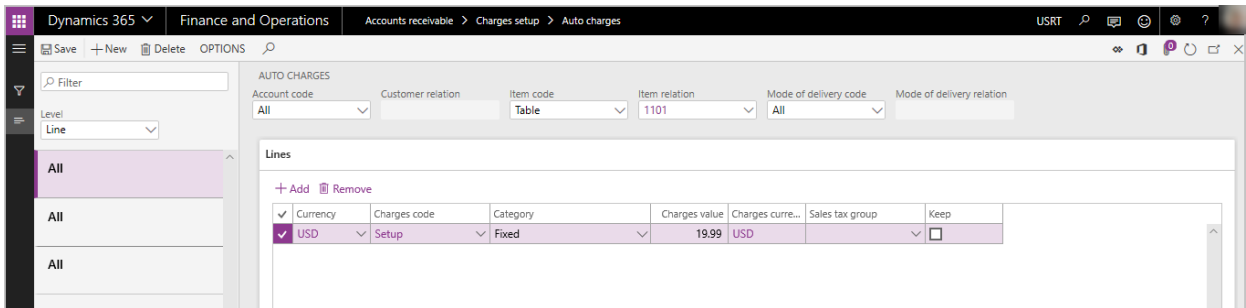
particular model of computer. This computer requires additional non-optional setup actions that the retailer will perform for the customer. The retailer has informed customers that there will be an additional fee for this setup. The retailer prefers to manage the charges related to this fee separately from the product sales price for financial reporting purposes. A setup fee of \$19.99 will be charged to the customer when this specific computer is purchased in any channel.

### Setup and configuration

This scenario requires the configuration of one line-level auto charges table.

Go to **Accounts Receivable > Charges setup > Auto charges**.

Set the **Level** drop-down menu to **Line**, and create a new auto charges record for all customers and for the specific product or product group where the setup fees will be charged.



Send the charges to the Commerce Scale Unit/Channel DB so that the POS can utilize them by running the **1040 distribution schedule job**.

### Sales processing for this scenario

After the configurations steps above are complete and the changes have been applied to the channel database, any customer order or sales transaction created in the POS, call center, or e-Commerce channels that have this item on the order will trigger a line-level charge to be systematically added to the order total.

At this time the charges will apply to any sales line that matches the configuration of the line-level auto charges within the legal entity, as there is no functionality to configure a line-level auto charge to apply only to a specific selling channel.

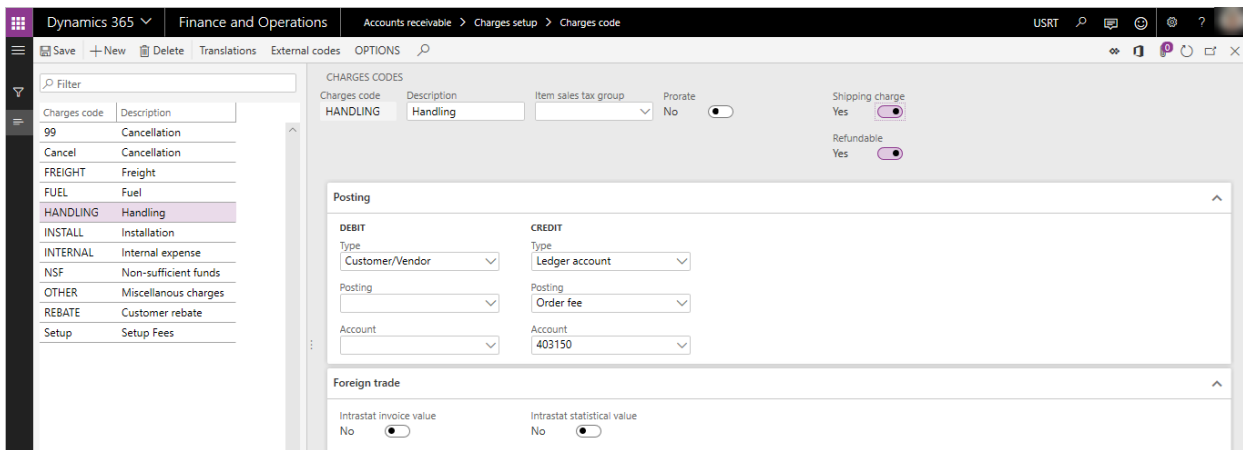
### Manual header charges example

#### Use case scenario description

A retailer is making an exception to typical processes by offering to provide a special home delivery of products to customers who order products in the store. The retailer and the customer have agreed that the customer will pay an additional \$25 handling fee for this service. The order-taker needs to add this additional fee to the transaction. Because the fee is a blanket fee and not related to any single product on the order, a header charge will be utilized.

### Setup and configuration

Ensure the charges code that will be used in this scenario has been properly configured by going to **Accounts Receivable > Charges setup > Charges** to define an appropriate charges code for the scenario.



If the charge should be considered a "shipping" related charge for the purpose of shipping related discounts or promotions, set **Shipping charge** on the charges code to **Yes**. If this charge is also allowed to be systematically refunded during the processing of a return transaction in the POS application, set **Refundable** to **Yes**. The **Refundable** flag is only applicable when the **Use advanced auto-charges** parameter is set to **Yes**.

Send the charges to the Commerce Scale Unit/Channel DB so that the POS can utilize them by running the **1040 distribution schedule job**.

The **Add header charge** operation must be configured in your [POS screen layout](#) so that a button that is accessible to the user from POS can call this operation (operation 141). The screen layout changes must be distributed to the channel as well through the distribution schedule function.

#### Sales processing of manual header charges

To execute the scenario in the POS application, the POS user will create the sales transaction as usual, adding the products and any other configurations to the sale. Prior to collecting payment, the user should execute the **Add header charge** operation, which will prompt the user to select a charges code and enter the charges value. Once the user completes the process, the charge will be added to the sales order as a header-level charge.

This process can be applied in the call center by using the existing **Charges** feature found on the **Sell** tab on the toolbar. On the **Maintain charges** page, the user can add a new charges line to the order header.

#### Manual line charges example

##### Use case scenario

A customer has requested that two of the five items on their sales order be gift-wrapped. The retailer offers this optional service for a fee of \$2.00 per item. The order-taker will need to add these fees to the specific items that need to be gift-wrapped.

##### Setup and configuration

Ensure the charges code that will be used in this scenario has been properly configured by going to **Accounts Receivable > Charges setup > Charges** to define an appropriate charges code for the scenario.

If the charge should be considered a "shipping" related charge for the purpose of shipping related discounts or promotions, set the **Shipping charge** on the charges code to **Yes**. If the charge is also allowed to be systematically refunded during the processing of a return transaction in the POS application, set **Refundable** to **Yes**. The **Refundable** flag is only applicable when the **Use advanced auto-charges** parameter is set to **Yes**.

Send the charges to the Commerce Scale Unit/Channel DB so that the POS can utilize them by running the **1040 distribution schedule job**.

The **Add line charge** operation must be configured in your [POS screen layout](#) so that a button that is accessible to the user from POS can call this operation (operation 140). The screen layout changes must be distributed to the channel as well through the distribution schedule function.

##### Sales processing of the manual line charge

To execute the scenario in the POS application, the POS user will create the sales transaction as usual, adding the

products and any other configurations to the sale. Prior to collecting payment, the user should select the specific line where the charge will apply from the POS item list display and execute the **Add line charge** operation. The user will be prompted to select a charges code and enter the charges value. Once the user completes the process, the charge will be linked to the line and added to the order total as a line level charge. The user can repeat the process to add additional line charges to other items lines on the transaction if needed.

The same process can be applied in the call center by using the "maintain charges" feature found under the **Financials** drop-down menu in the **Sales order lines** section on the **Sales order** page. Selecting this option will open the **Maintain charges** page where the user can add a new line-specific charge to the transaction.

## Additional features

### Editing charges on a POS sales transaction

The **Manage charges** operation (142) should be added to the **POS screen layout** so that a user can view and edit or override any system-calculated or manually created header or line-level charges. If the operation is not added, users will not be able to adjust the value of the charges on the POS transaction, nor will they be able to view the details of the charges such as the type of charges code tied to the charge.

On the **Manage charges** page in POS, the user can view both header and line-level charges details. The user can use the **Edit** function available on this page to make changes to the amount charged to a specific charges line. Once a charges line is overwritten manually, it will not be systematically recalculated unless the user initiates the **Recalculate charges** operation.

If the **Charge override reason code** has been configured on the **Commerce parameters** setup page, the user will be prompted to provide a reason code when charges have been modified in the POS application.

If reason codes have been captured for overwritten charges, a new report is also available to review and audit these overrides. The report can be found in **Retail and Commerce > Inquiries and reports > Charge override history**.

### Refunding charges on a POS return transaction

If the **Use advanced auto-charges** parameter is set to **Yes**, the existing Commerce parameter for **Refund shipping charges** is no longer applicable. To indicate which charges should be systematically refunded to a customer when using advanced auto charges, ensure the related charges code has been configured as **Refundable** on the **Charges code** setup page. Make sure that the settings have been synchronized to your Commerce channel databases through distribution schedule processing.

### Refunding charges on a return order transaction

Charges are not systematically refunded to **Return orders** created in Commerce. Users are required to select the **Copy charges** option when creating the **Return order**. If **Copy charges** is not selected, charges from the original sales transaction will not be automatically refunded. If **Copy charges** is selected, all charges will be copied to the return order and the user can manually edit or remove any charges they do not want to have refunded. The call center return order process currently does not acknowledge the **Refundable** flag on the **Charges code** setup.

### Configuring POS receipts to show charges

The following receipt elements have been added to the receipt line and footer to support the advanced auto charges functionality.

- **Line Shipping Charges** – This line-level element can be used to recap specific charges codes that have been applied to the sales line. Only charges codes that have been flagged as **Shipping** charges on the **Charges code** page will be displayed here.
- **Line Other Charges** – This line-level element can be used to recap any non-shipping specific charge codes that have been applied to the sales line. These are charges codes where the **Shipping** flag on the **Charges code** page has not been enabled.

- **Order Shipping Charges Details** – This footer-level element displays the descriptions of the charge codes applied to the order that have been flagged as **Shipping** charges on the **Charges code** setup page.
- **Order Shipping Charges** – This footer-level element shows the dollar value of the shipping-related charges.
- **Order Other Charges Details** – This footer-level element displays the description of the charges codes applied to the order that have not been flagged as shipping-related charges.
- **Order Other Charges** – This footer-level element displays the dollar value of the other charges that are not shipping-related.

It is recommended that the organization also add free text fields to the receipt footer, in order to define the areas where charges will be recapped.

### **Preventing charges from being calculated until the POS order is completed**

Some organizations may prefer to wait until the user has finished adding all of the sales lines to the POS transaction before calculating charges. To prevent calculation of charges as items are added to the POS transaction, turn on the **Manual charge calculation** parameter in the **Functionality profile** used by the store. Enabling this parameter will require the POS user to use the **Calculate totals** operation when they have completed adding the products to the POS transaction. The **Calculate totals** operation will then trigger the calculation of any auto charges for the order header or lines as applicable.

### **Charges override reports**

If users manually override the calculated charges or add a manual charge to the transaction, this data will be available for auditing in the **Charge Override History** report. The report can be accessed from **Retail and Commerce > Inquiries and reports > Charge Override History**. It is important to note that the data needed for this report is imported from the channel database into HQ through the "P" distribution schedule jobs. Therefore, information about overrides just performed in the POS may not be immediately available on this report until this job has uploaded the store transaction data into HQ.

## **Additional resources**

[Enable and configure auto charges by channel](#)

[Prorate header charges to matching sales lines](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable and configure auto charges by channel

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how to enable and configure automatic charges (auto charges) by channel in Microsoft Dynamics 365 Commerce.

## Overview

You might have scenarios where recycling fees or other fees must be applied to a group of products that are sold in all or some stores in a specific state (for example, California). The **Enable filter auto charges by channel** feature in Commerce lets you specify auto charges by channel (for example, a specific brick-and-mortar channel). This feature is available in Dynamics 365 Commerce version 10.0.10 and later.

To enable and configure auto charges by channel, you must complete the following tasks:

- Turn on the **Enable filter auto charges by channel** feature.
- Configure the organization hierarchy purpose.
- Define auto charges by channel.

### NOTE

The **Enable filter auto charges by channel** feature works only if the advanced auto charges feature is also turned on. For information about how to turn on the advanced auto charges feature, see [Omni-channel advanced auto charges](#).

## Turn on the Enable filter auto charges by channel feature

To enable auto charges by channel in Commerce, follow these steps.

1. Go to **System administrator > Workspaces > Feature management**.
2. On the **Not enabled** tab, in the **Feature name** list, find and select **Enable filter auto charges by channel**.
3. In the lower-right corner, select **Enable now**. After the feature has been turned on, it will appear in the list on the **All** tab.
4. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
5. In the left pane, find and select the **1110 (Global configuration)** job.
6. On the Action Pane, select **Run now** to propagate the configuration changes.

### WARNING

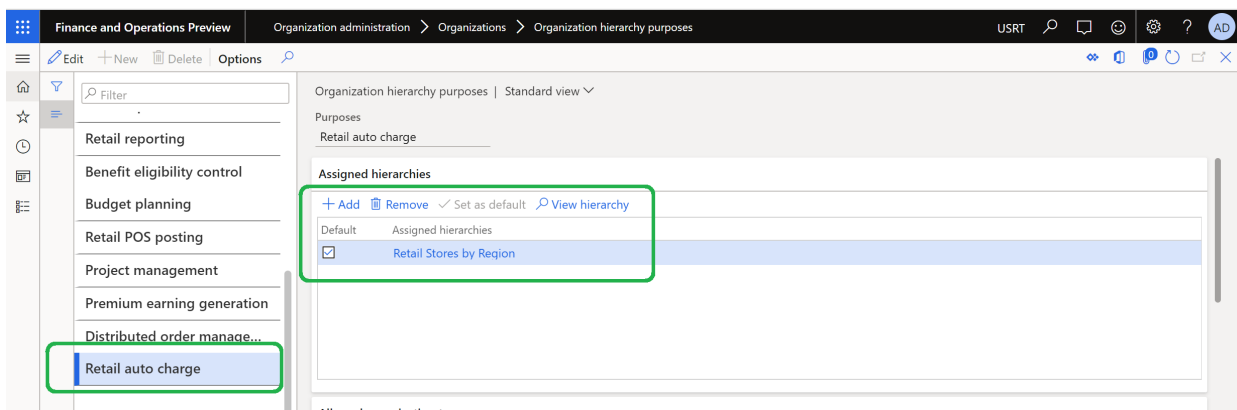
If you turn off the **Enable filter auto charges by channel** feature after you've already used it, the **Retail channel relation** field under **Auto charges** will no longer appear, and you will lose all existing configurations. If removal of the **Retail channel relation** configurations will cause auto charges rules to be duplicated, an attempt to turn off the feature will fail. Before you turn off the feature, be sure to review all auto charges rules and make any required changes.

## Configure the organization hierarchy purpose

A new organization hierarchy purpose that is named **Retail auto charge** has been created to manage the hierarchy for auto charges by channel.

To assign a default hierarchy to an organization hierarchy purpose in Commerce, follow these steps.

1. Go to **Organization administration > Organizations > Organization hierarchy purposes**.
2. In the left pane, select **Retail auto charge**.
3. Under **Assigned hierarchies**, select **Add**.
4. In the **Organization hierarchies** dialog box, select an organization hierarchy (for example, **Retail Stores by Region**), and then select **OK**.
5. Under **Assigned hierarchies**, select **Set as default**.
6. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
7. In the left pane, find and select the **1040 (Products)** job.
8. On the Action Pane, select **Run now**.
9. Repeat the previous two steps to run the **1070 (Channel configuration)** and **1110 (Global configuration)** jobs.



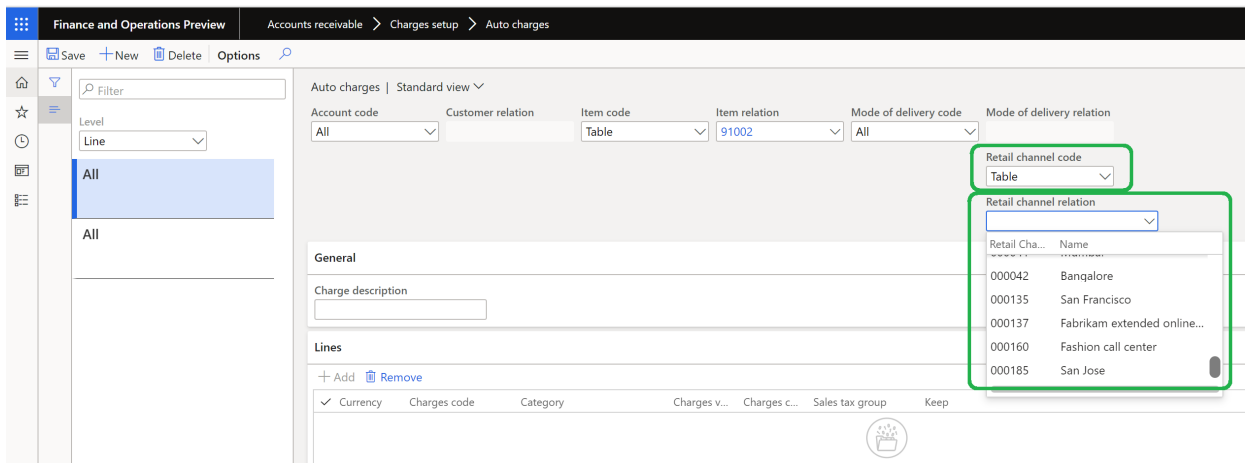
## Define auto charges by channel

After you've turned on the **Enable filter auto charges by channel** feature and configured the **Retail auto charge** organization hierarchy purpose, auto charges by channel can be defined at either the order header level or the order line level.

To define auto charges by channel in Commerce, follow these steps.

1. Go to **Accounts receivable > Charges setup > Auto charges**.
2. In the left pane, in the **Level** field, select either **Header** or **Line**, depending on your business requirements.
3. In the **Retail channel code** field, select the appropriate channel code (for example, **Table** or **Group**). If the default setting, **All**, is used, charge rules are applied to all channels.
  - If you select **Group**, make sure that a retail channel charges group is created at **Retail and Commerce > Channel setup > Charges > Retail channel charge groups**.
  - If you select **Table**, you can select a specific channel (for example, **San Francisco**) in the **Retail channel relation** field.
4. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
5. In the left pane, find and select the **1040 (Products)** job.
6. On the Action Pane, select **Run now**.
7. Repeat the previous two steps to run the **1070 (Channel configuration)** and **1110 (Global configuration)** jobs.

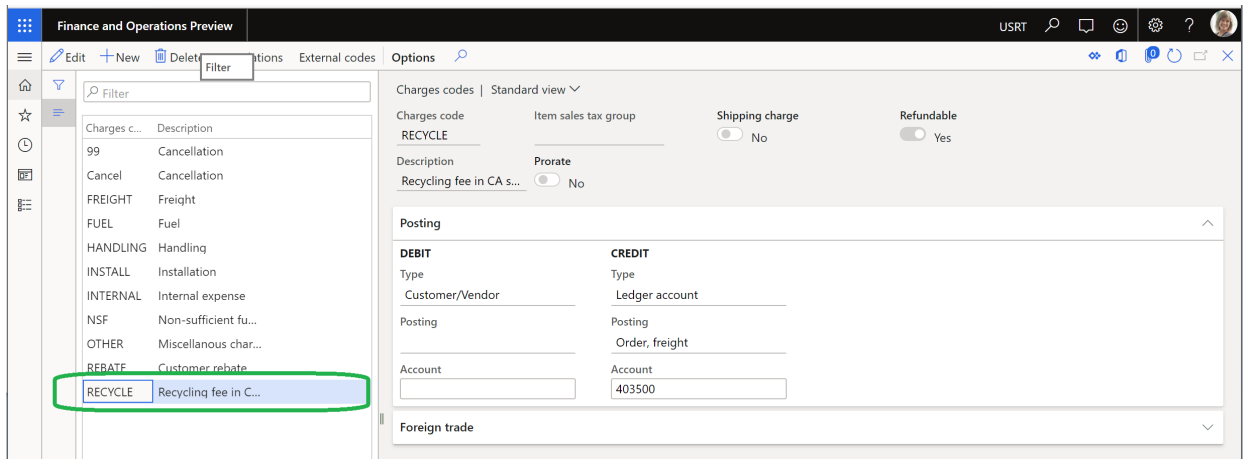




## Example scenario

The following example outlines the steps that are required to configure a product so that recycling fees are charged when the product is sold through a San Francisco brick-and-mortar channel. The example also shows how the auto charges appear in the Commerce point of sale (POS) application.

The organization defines a charges code that is named **RECYCLE**, as shown in the following illustration.

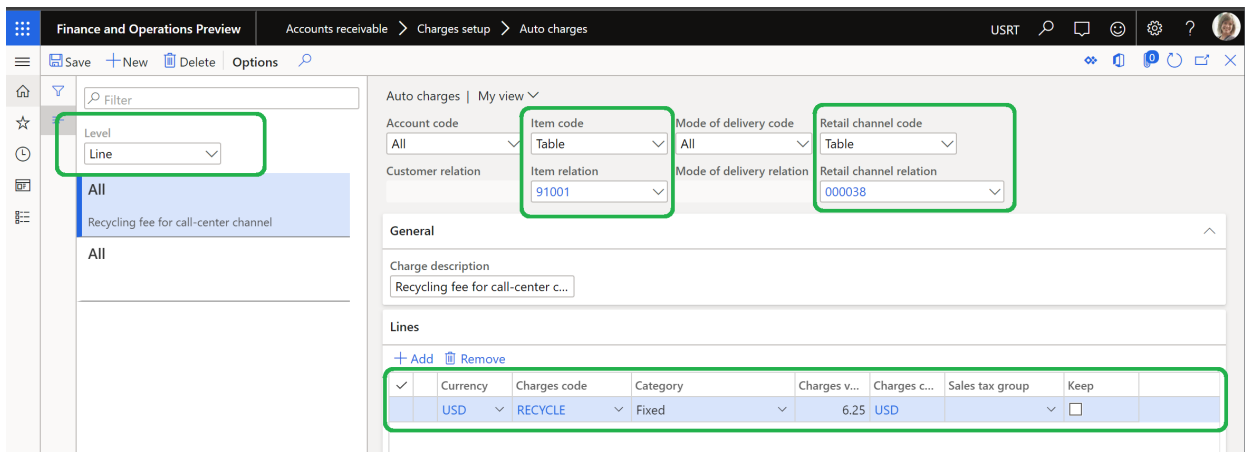


An auto charge is created at the line level. It has the following configuration:

- The **Account code** field is set to **All**.
- The **Item code** field is set to **Table**.
- The **Item relation** field is set to product ID **91001**.
- The **Mode of delivery code** field is set to **All**.
- The **Retail channel code** field is set to **Table**.
- The **Retail channel relation** field is set to the **San Francisco** store.

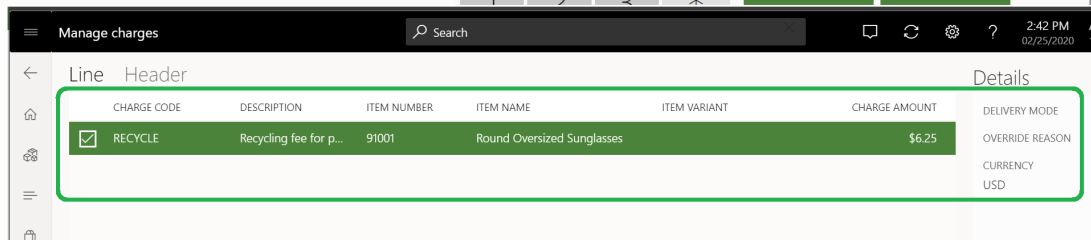
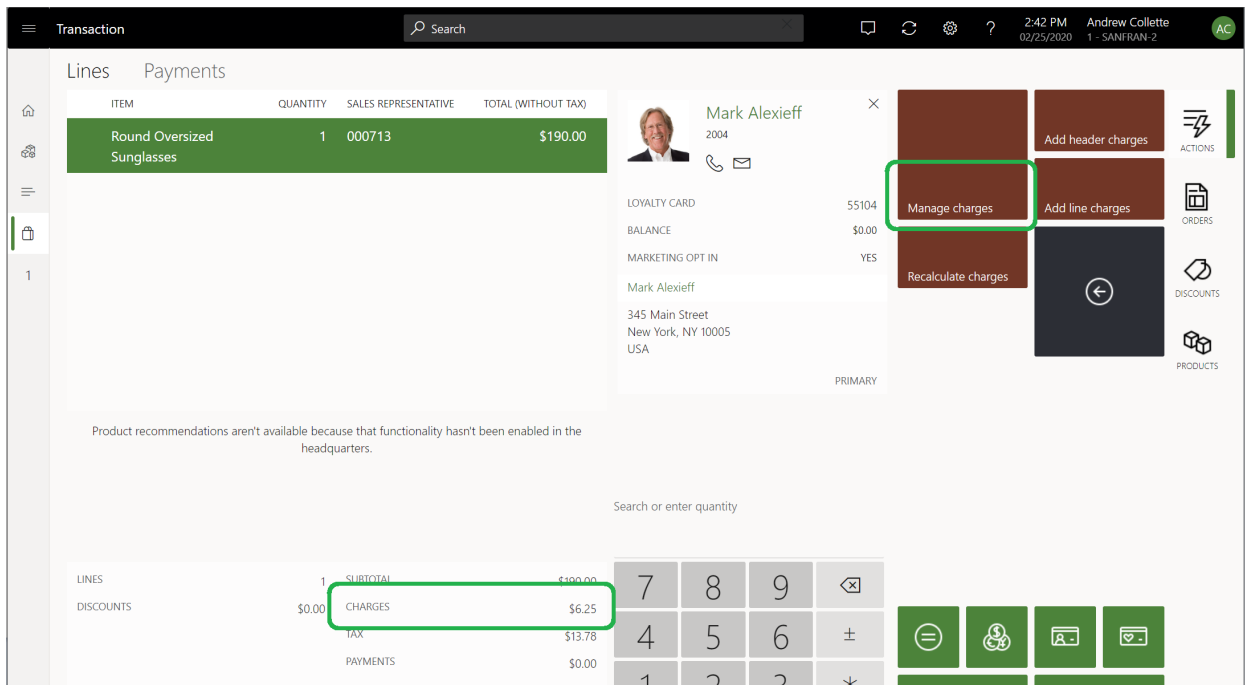
An auto charges line is created. It has the following configuration:

- The **Currency** field is set to **USD**.
- The **Charges code** field is set to **RECYCLE**.
- The **Category** field is set to **Fixed**.
- The **Charges** field is set to **\$6.25**.



In the POS application, a sales order is created in the **San Francisco** store channel. The **Charges** line shows the recycling fee of **\$6.25**.

By selecting **Transaction options > Charges > Manage charges** in the POS application, you can view the charges code and description for the recycling fee.



## Additional resources

[Omni-channel advanced auto charges](#)

[Prorate header charges to matching sales lines](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Prorate header charges to matching sales lines

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes the functionality for grouping header-level auto-charges and prorating them to commerce sales lines. This functionality is available for transactions that are created at the point of sale (POS) in Retail version 10.0.1 and sales that are created in a call center in Retail version 10.0.2.

This functionality is available only if the [advanced auto-charges](#) feature is turned on by using the option on the **Commerce parameters** page. Additionally, the enhanced calculation method for auto-charges can be applied only to sales orders that are created through commerce channels (the POS, a call center, and the Dynamics e-Commerce platform).

This new functionality gives organizations more flexibility in the way that header-level auto-charges are calculated and applied to sales transactions.

In versions of the app earlier than version 10.0.1, header-level auto-charges that have a specific mode of delivery relation are calculated only when there is a match with the mode of delivery that is defined on the sales order header.

For example, header-level auto-charges are defined for mode of delivery **99** and mode of delivery **11**. A sales order is created, and mode of delivery **99** is defined on the order header. However, some of the sale lines are set up so that they're shipped by using mode of delivery **11**. In this case, only the header-level charges that are linked to mode of delivery **99** are considered and applied to the sales order.

In Commerce, the header-level charges have an additional feature that lets you define a [tiered charge configuration](#) that is based on the order value. For example, if the order value is between \$50.00 and \$200.00, an organization might want to charge a freight charge of \$5.00. However, if the order value is between \$200.01 and \$500.00, the freight charge might be \$4.00.

Some organizations want the benefits of the tiered charge calculation that is provided with header-level charges. However, in scenarios that involve mixed modes of delivery, they also want to make sure that the charges that are calculated are based on a match with the mode of delivery that is defined on each sales line.

You can now configure header-level auto-charges so that all modes of delivery on the order are considered when charges are calculated. This functionality requires a more complex calculation logic to calculate the header-level charges. The logic groups together all the items that are shipped using the same mode of delivery, and it treats that group as the calculation group for the items when it calculates the header-level auto-charges. For items that have the same mode of delivery, auto-charges are calculated based on the combined sales value of the items. In this way, the appropriate auto-charge tier is determined.

After the appropriate header-level charges are obtained for the sales lines that are shipped using the same mode of delivery, the calculated charges are prorated down to the sales line level. Because these charges are at the line level and not kept at the header level, a more specific link is made between the item and the charge value that calculated for it. This behavior can be useful in partial return scenarios, where an organization wants to refund only part of the charge instead of the whole charge when only some items are returned.

## Scenarios

The following two sample scenarios outline how these charges are calculated both when the new functionality is used and when it isn't used.

### Scenario 1

This scenario outlines the behavior when the **Pro-rate to matching sales lines** option is set to **No** in the auto-charge setup. (The behavior is equivalent to the behavior of header-level charges in app versions that are earlier than version 10.0.1.)

In this scenario, the organization has defined header-level charges for mode of delivery relation 99 and mode of delivery relation 11. No auto-charges are configured for mode of delivery 21.

AUTO CHARGES

Account code: All | Customer relation: | Item code: All | Item relation: | Mode of delivery code: Table | Mode of delivery relation: 99 | Pro-rate to matching sales lines: No

Lines

+ Add Remove

✓	Currency	Charges code	Category	Charges value	Charges currenc...	From amount ↑	To amount	Sales tax group	Keep
	USD	FREIGHT	Fixed	10.00	USD	0.01	50.00		<input type="checkbox"/>
	USD	FREIGHT	Fixed	15.00	USD	50.01	250.00		<input type="checkbox"/>
✓	USD	FREIGHT	Fixed	20.00	USD	250.01	99,999.99		<input type="checkbox"/>

AUTO CHARGES

Account code: All | Customer relation: | Item code: All | Item relation: | Mode of delivery code: Table | Mode of delivery relation: 11 | Pro-rate to matching sales lines: No

Lines

+ Add Remove

✓	Currency	Charges code	Category	Charges value	Charges currenc...	From amount	To amount	Sales tax group	Keep
	USD	FREIGHT	Fixed	5.00	USD	0.01	50.00		<input type="checkbox"/>
	USD	FREIGHT	Fixed	7.00	USD	50.01	100.00		<input type="checkbox"/>
	USD	FREIGHT	Fixed	8.00	USD	100.01	250.00		<input type="checkbox"/>
✓	USD	FREIGHT	Fixed	0.00	USD	250.01	9,999,999.00		<input type="checkbox"/>

A sales order is created in the call center, and the mode of delivery is set to 99. This order contains five items. Two order lines have been configured to use mode of delivery 99, two lines have been configured to use mode of delivery 11, and one line has been configured to use mode of delivery 21, as shown in the following table.

ITEM	LINE QUANTITY	DELIVERY MODE	PRICE PER UNIT
81331	1	11	\$10
81332	1	99	\$50
81333	2	11	\$30
81334	3	99	\$10
81334	3	21	\$5

In this scenario, the whole order is evaluated against the auto-charge table for mode of delivery 99. The full total of all sales lines is used to determine a matching tier in the auto-charge configuration, and this charge is applied at the order header level. In this example, the order total is \$165.00, and the \$15.00 freight charge is applied to the order header. Auto-charges that are configured for mode of delivery 11 are never referenced or applied.

In this scenario, if a customer returns some of the items on the order, and if the [charge code has been configured so that it will be refunded](#), the total header-level charge is systematically applied to the refund, even if only some of the items are returned.

## Scenario 2

In this scenario, header-level charges are defined for mode of delivery relation 99 and mode of delivery relation 11. However, the **Pro-rate to matching sales lines** option is set to **Yes** for these auto-charge tables.

AUTO CHARGES									
Account code	Customer relation	Item code	Item relation	Mode of delivery code	Mode of delivery relation	Pro-rate to matching sales lines			
All		All		Table	99	Yes <input checked="" type="checkbox"/>			

Lines									
✓	Currency	Charges code	Category	Charges value	Charges currenc...	From amount ↑	To amount	Sales tax group	Keep
	USD	FREIGHT	Fixed	10.00	USD	0.01	50.00		<input type="checkbox"/>
	USD	FREIGHT	Fixed	15.00	USD	50.01	250.00		<input type="checkbox"/>
✓	USD	FREIGHT	Fixed	20.00	USD	250.01	99,999.99		<input type="checkbox"/>

AUTO CHARGES									
Account code	Customer relation	Item code	Item relation	Mode of delivery code	Mode of delivery relation	Pro-rate to matching sales lines			
All		All		Table	11	Yes <input checked="" type="checkbox"/>			

Lines									
✓	Currency	Charges code	Category	Charges value	Charges currenc...	From amount	To amount	Sales tax group	Keep
	USD	FREIGHT	Fixed	5.00	USD	0.01	50.00		<input type="checkbox"/>
	USD	FREIGHT	Fixed	7.00	USD	50.01	100.00		<input type="checkbox"/>
	USD	FREIGHT	Fixed	8.00	USD	100.01	250.00		<input type="checkbox"/>
✓	USD	FREIGHT	Fixed	0.00	USD	250.01	9,999,999.00		<input type="checkbox"/>

This scenario uses the same sales order that contains five lines. The mode of delivery on the order header is set to 99, but the mode of delivery for each item on the sales order is configured as shown in the following table.

ITEM	LINE QUANTITY	DELIVERY MODE	PRICE PER UNIT
81331	1	11	\$10
81332	1	99	\$50
81333	2	11	\$30
81334	3	99	\$10
81334	3	21	\$5

Because the auto-charge configuration is set to prorate to matching sales lines, the system performs the following calculation steps.

1. All items that have the same mode of delivery are grouped together, and the system calculates the total product value of the items in the group.

#### Delivery mode 11

- Item 81331, quantity 1 = \$10
- Item 81333, quantity 2 = \$60 net (\$30 per unit)
- **Total product value for delivery mode 11 = \$70**

#### Delivery mode 99

- Item 81332, quantity 1 = \$50
- Item 81334, quantity 3 = \$30 net
- **Total product value for delivery mode 99 = \$80**

## Delivery mode 21

- Item 81334, quantity 3 = \$15 net
  - **Total product value for delivery mode 21 = \$15**
2. The system looks for the configuration for header-level auto-charges that matches the customer and mode of delivery settings for each group of items. If the configuration is found, the system looks in the tiered configuration to find the charge that should be applied, based on the total product value of items in the mode of delivery group.

## Delivery mode 11

- Total product value = \$70
- **Charge value = \$7**

## Delivery mode 99

- Total product value = \$80
- **Charge value = \$15**

## Delivery mode 21

- Total product value = \$15
- **Charge value = \$0** (No auto-charges have been configured for this combination of a customer and a mode of delivery.)

AUTO CHARGES									
Account code	Customer relation	Item code	Item relation	Mode of delivery code	Mode of delivery relation	Pro-rate to matching sales lines			
All		All		Table	11	Yes			
Lines									
+ Add - Remove									
✓ Currency	Charges code	Category	Charges value	Charges curre...	From amount	To amount	Sales tax group	Keep	
USD	FREIGHT	Fixed	5.00	USD	0.01	50.00			
USD	FREIGHT	Fixed	7.00	USD	50.01	100.00			
USD	FREIGHT	Fixed	8.00	USD	100.01	250.00			
USD	FREIGHT	Fixed	0.00	USD	250.01	9,999,999.00			

AUTO CHARGES									
Account code	Customer relation	Item code	Item relation	Mode of delivery code	Mode of delivery relation	Pro-rate to matching sales lines			
All		All		Table	99	Yes			
Lines									
+ Add - Remove									
✓ Currency	Charges code	Category	Charges value	Charges curre...	From amou... 1	To amount	Sales tax group	Keep	
USD	FREIGHT	Fixed	10.00	USD	0.01	50.00			
USD	FREIGHT	Fixed	15.00	USD	50.01	250.00			
USD	FREIGHT	Fixed	20.00	USD	250.01	99,999.99			

3. The system calculates the charge value that should be applied to each line, based on proration logic that considers the proportional value of the line in relation to the group's total product value.

## Delivery mode 11

- Charge value = \$7
- Group product value = \$70
- Line 1 value = \$10 (= 14.2857 percent of the group value)
- Line 3 value = \$60 (= 85.7143 percent of the group value)
- **Line charge for line 1 = \$1**
- **Line charge for line 3 = \$6**

## Delivery mode 99

- Charge value = \$15
- Group product value = \$80
- Line 2 value = \$50 (= 62.5 percent of the group value)
- Line 4 value = \$30 (= 37.5 percent of the group value)
- **Line charge for line 2 = \$9.38**

- Line charge for line 4 = \$5.62

#### Delivery mode 21

- Charge value = \$0
- Group product value = \$15
- Line 5 value = \$15 (= 100 percent of the group value)
- Line charge for line 5 = \$0

Therefore, for this example, item 81334 will be assigned a freight charge of \$5.62. You can view these charges on the **Maintain charges** page for the sales line. The following illustration shows what this page looks like for item 81334.

Charges code	Description	Category	Keep	Charges value	Currency	Broker contract...	Sales tax group	Item sales tax group
FREIGHT	Freight	Fixed	<input type="checkbox"/>	5.62	USD	<input type="checkbox"/>	OH	

**LEDGER**

Voucher:

Date:

Amount in transaction currency:

Calculated value:

**INTERCOMPANY**

Company:

Intercompany invoice number:

When this method of calculation is used in a partial return scenario, if the charge code is refundable, only the part of the charge that is allocated to that line will be refunded when the item is returned.

## Additional resources

[Omni-channel advanced auto charges](#)

[Enable and configure auto charges by channel](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up continuity programs for call centers

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes how to set up a continuity program for a call center.

In a continuity program, which is also known as a recurring order program, customers receive regular product shipments according to a predefined schedule. Each shipment can contain a different product, as in the case of a book-of-the-month club, or the same product can be sent repeatedly. To set up a continuity program, you must complete the following tasks.

1. Set the continuity parameters on the **Call center parameters** page.
2. Create a continuity program that specifies details such as the payment schedule, the timing of the shipments, and whether billing is up front. You must also add a list of products that are included in the continuity program. Each product receives an event ID number that is assigned sequentially, beginning with 1. The event IDs determine the order that products are sent in.
  - If customers receive a different product in each shipment, the products are sent in sequential order, based on their event IDs and beginning with the current event.
  - If customers receive the same product in each shipment, the list contains only one product that has one event ID. The same event occurs repeatedly. You can specify how many times each event is repeated.
3. Create a parent product that represents the continuity program that you created in task 2. If you add this product to a sales order, the **Continuity** page opens. You can then use that page to create the actual continuity order. The parent product doesn't specify the individual products that the customer receives in each shipment.

After you've set up a continuity program as described above, you can create a continuity order for a customer. You might also have to perform the following additional maintenance tasks.

- **Update the current continuity event period** – Set up a batch job that tells the system what the current event period is.
- **Create continuity child orders** – Create child orders from the parent continuity order.
- **Process continuity payments** – Process billing and notifications for payments that are associated with continuity sales orders.
- **Extend continuity lines** (if required) – Extend the number of times that a continuity event can be repeated. The repetition of shipments can then extend beyond the limit that was set in the **Continuity repeat threshold** field in the call center parameters.
- **Perform a continuity update** (if required) – Synchronize changes between the continuity program and the continuity parent sales orders.
- **Close continuity parent lines and orders** – Close continuity orders.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Define continuity schedules

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic walks through setting up a continuity program (otherwise known as reoccurring orders). This topic uses company USRT in the demo data.

## Create continuity program

1. Go to Retail and Commerce > Continuity > Continuity programs.
2. Click New.
3. In the Schedule ID field, type the continuity schedule ID.
4. In the Order start field, select 'First event'.
  - If a customer places a new order for the continuity program, there are two options for which product will be shipped: 1. First event: the first product in the continuity program will be shipped. 2. Current event: the current product will be shipped. For example. three months into the program, the customer will receive the third in the program.
5. Select 'Yes' to prompt for the order start date.
6. Click Add line.
7. In the Item number field, type the item number for the first product ('0013').
8. Type 'CP'.
9. Enter the date when the first product will be available for order.
10. Click Add line.
11. In the Item number field, type '0014'.
12. In the Date interval code field, clear the value so the field is empty.
  - For this procedure, clear the date interval. You'll set the date as incremental from the start date of the first item.
13. Here you'll enter the interval at which the products are shipped. Type '30'.
  - For a monthly program, you'll enter 30 days for the interval.
14. Click Add line.
15. In the Item number field, type '0015'.
16. Type 'End'.
17. Click Save.

## Assign to continuity item

1. Go to Product information management > Products > Released products.
2. Select item '0016'.
  - For this procedure, you'll select item number 0016. Normally, you'll have created a released product that has additional continuity business logic applied when it's placed on a sales order in call center.
3. In the list, click the link in the selected row.
4. Click Edit.
5. Expand or collapse the Sell section.
6. Here you'll enter the continuity program that this item represents. Type the Schedule ID you created earlier.
  - When this item is sold in a call center, additional business logic is applied from the selected continuity program.
7. Click Save.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create and update store hours

2/18/2021 • 2 minutes to read • [Edit Online](#)

## Overview

From a single place, retailers can create, maintain, and manage the store hours for different stores across geographic regions. The store hours can then be shown on point of sale (POS) terminals. In this way, cashiers can share store hours with customers and better help shoppers who are interested in inventory in other stores. The store hours can also be printed on receipts, in case customers want to return to the store later.

Multiple store hours can be configured across different channels. These channels include brick-and-mortar stores, call centers, mobile devices, and e-Commerce sites.

If a customer has a pickup order for a different store, the cashier can select dates when the pickup will be available in that store. The store lookup will provide a reference to the dates and store times. The cashier can select a date and location, and can also print a pickup receipt that includes the store hours.

This functionality is available in Microsoft Dynamics 365 Retail versions 8.1.2 and later.

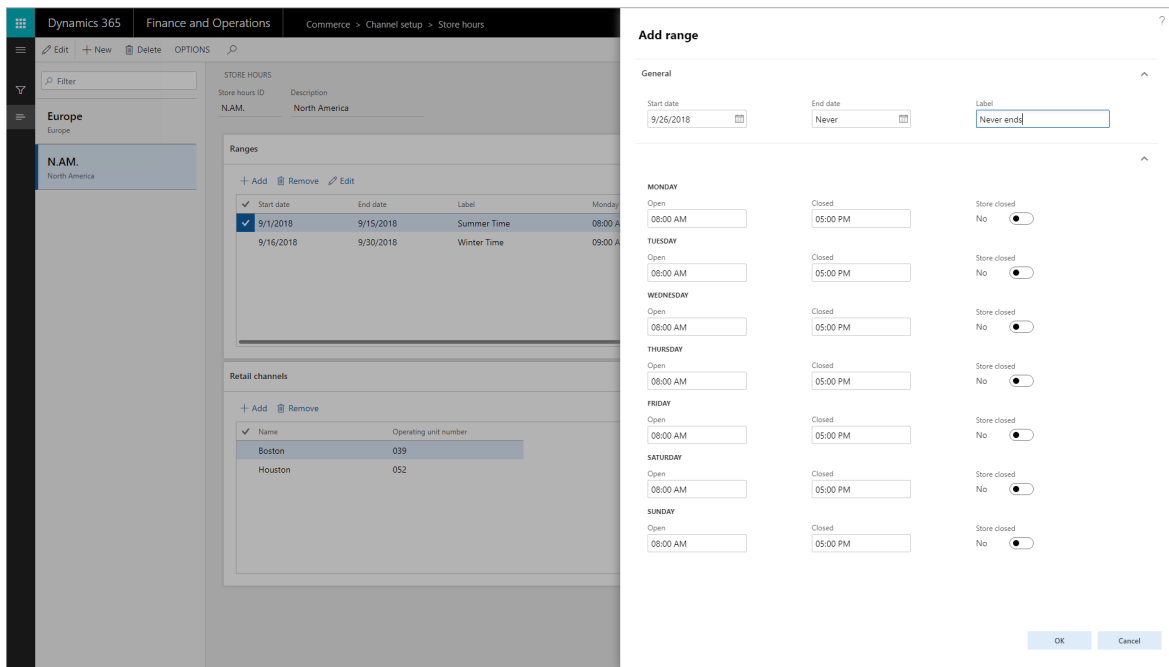
## Configure store hours

Follow these steps to configure store hours.

1. Go to **Retail and Commerce > Channel Setup > Store hours**.
2. Select **New** to create a new store hours template. To use an existing template, select the template in the left pane.
3. In the **Add range** dialog box, define the date range, the store hours, and any holidays that are required.
  - If store hours don't change, select **Never ends** in the **End date** field.
  - If the store hours are for a specific month, week, or day, set the appropriate dates in the **Start Date** and **End date** fields.

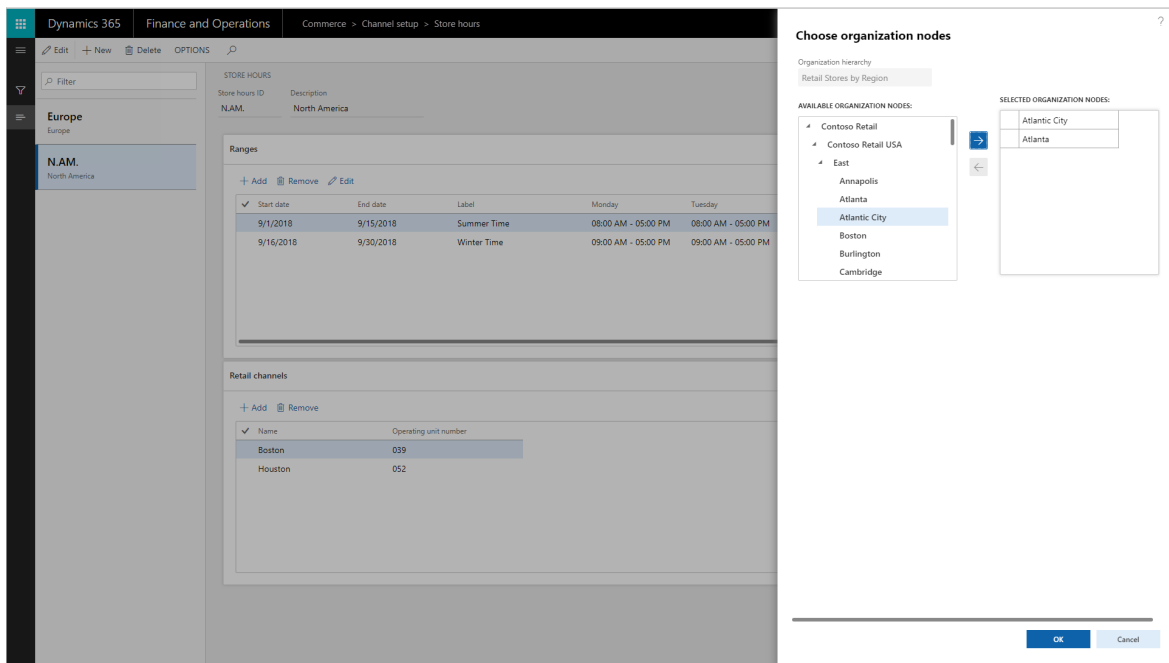
### NOTE

You can create multiple templates that have overlapping start and end dates. Therefore, you can, for example, define store hours for stores in different time zones.



4. Associate the store hours template with the stores where it will be used. In the **Choose organization nodes** dialog box, select the stores, regions, and organizations that the template should be associated with.

- Only one store hours template can be associated with each store.
- Use the arrow buttons to select stores, regions, or organizations. The calendar will be available to the stores or store groups, and it will be visible at the POS for reference.



5. On the **Distribution schedule** page, run the 1070 and 1090 jobs to make the store hours available to the POS.

## Add store hours to printed receipts

Follow these steps to add store hours to the printed POS receipts.

1. Open the receipt designer.
2. Select **Footer** in the lower-left corner.
3. Drag the **Store hours** element from the left pane to the footer at the bottom of the receipt template.
4. You can edit the default label on the **Store hours** element as you require.

5. Save the receipt, and close the receipt designer.
6. On the **Distribution schedule** page, run the **1070** and **1090** jobs.
7. Sign in to the POS.
8. Complete a sale, and select to print a receipt.

POS receipts now include the store hours. If any holidays were included in the template, they are shown on the receipt.

<b>Random Receipt</b>		
Selling Item 1	X 2 Qty	\$3.67
(Random Product Description)		
Selling Item 2	X 2 Qty	\$3.67
(Random Product Description)		
Selling Item 3	X 2 Qty	\$3.67
(Random Product Description)		
Selling Item 4	X 2 Qty	\$3.67
(Random Product Description)		
<b>Total</b>		<b>\$14.68</b>
<b>Discount</b>		<b>\$00.00</b>
<b>Taxes</b>		<b>\$00.00</b>
<b>Store Hours</b>		
Monday - Friday: 8:00 am - 7:00 pm		
Saturday: 10:00 am - 4:00 pm		
Sunday: Closed		

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce Scale Unit (self-hosted)

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes Commerce Scale Unit (self-hosted) and when to use it.

## Overview

Commerce Scale Unit (self-hosted) is a set of features that supports selling products in a store that has inconsistent internet connectivity to a back office or headquarters (HQ). The Store Scale Unit is designed specifically for in-store operation, and enables cross-terminal transactions and shift operations despite poor internet service. By automatically connecting to the back office, when you do have internet connectivity, your store can seamlessly process credit card transactions, issue gift cards, and sync data with HQ. The Store Scale Unit is available for download in the standard HQ deployment.

## Is Commerce Scale Unit (self-hosted) right for you?

Before you begin setting up Commerce Scale Unit (self-hosted), take a moment determine whether this option is the right fit for your store. Commerce Scale Unit (self-hosted) is a deployment choice intended for retailers with store locations that have slow or intermittent internet connectivity, and who need the flexibility of the Commerce Scale Unit deployed on premises in each store. In scenarios where a stable internet connection is available and there is low latency to the cloud environment, then it is recommended to consider operating the store as Cloud only, without setting up a Commerce Scale Unit. Consider the following before you begin:

- Carefully choose the store topology configuration for each store to either operate with a self-hosted or cloud-hosted Commerce Scale Unit topology. Reconfiguring a live store from a self-hosted to cloud-hosted Commerce Scale Unit or vice versa may cause a service disruption.
- Commerce Scale Unit will support both MPOS and Cloud POS within the store.
- Commerce Scale Unit (self-hosted) can be set up in a one-box deployment topology on a single computer (recommended) or in a multi-box topology on different computers.
- If you choose the one-box option, most of the settings are pre-configured. For a multi-box topology, you will have to manually configure connections between components.
- With Commerce Scale Unit (self-hosted), users can perform cross-terminal scenarios across multiple POS devices, like suspend/recall transactions and shift operations, even with temporary network disruption to HQ.
- With Commerce Scale Unit (self-hosted), users cannot perform any real-time operations such as issuing gift cards, looking up products, or performing credit card transactions, unless there is internet connectivity to HQ or a payment provider. If most of your transactions involve real-time transactions, then you will always need internet connectivity to enable the connection to HQ or payment provider.
- Direct database connectivity from POS to the channel database is not supported in the Commerce Scale Unit. The POS devices always use the Commerce Scale Unit for performing operations.

### NOTE

It is critical to note that Commerce Scale Unit (self-hosted) does not replace offline. Currently, Retail Modern POS with an offline database is the only way to have offline capabilities.

## Get started with Commerce Scale Unit (self-hosted)

To get started, review the following topic on configuring the Commerce Scale Unit (self-hosted), [Configure and](#)

[install Commerce Scale Unit \(self-hosted\)](#).

## Additional resources

[Configure and install Commerce Scale Unit \(self-hosted\)](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and work with call center order holds

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic describes the order hold features that Dynamics 365 Commerce has for call center orders.

## Configuring call center order holds

To use the call center order hold features, you must first define hold codes. To create a set of user-defined hold codes, based on your business requirements, go to **Sales and marketing > Setup > Sales orders > Order hold codes**. You can optionally flag one of the hold codes as the default hold code by setting the **Default for sales order** option to **Yes** for it. This hold code will be used any time that a sales order is put on hold. If a sales order has reserved inventory, and the reservations must be automatically removed if the order is put on hold for a particular reason, set the **Remove inventory reservations** option to **Yes** for the reason codes.

To specify the type of note that will be captured when users who put a sales order on hold enter optional notes, go to **Accounts receivable > Setup > Accounts receivable parameters**, and then, on the **Sales setup** FastTab, on the **General** tab, set the **Note type** field. Use the **On Hold sales order status** field to define the color that will be used to highlight sales orders that are on hold when they are viewed on the **Customer service** page.

To create an optional set of hold reason codes, go to **Retail and Commerce > Channel setup > Info codes**. These info codes can be used as a secondary reason code to further define the main hold code. Select **New** to create a reason code set, and then select **Subcodes** to define the list of additional reasons. To link any info codes that you define to the call center channel, go to **Retail and Commerce > Channels > Call centers > All call centers**. On the **General** FastTab, set the **Hold code** field.

## Putting orders on hold

Orders that call center users create in the back-office Commerce program can be manually put on hold manually or automatically in specific situations.

During order entry, but before order submission and confirmation, call center users might want to manually put an order on hold to prevent it from being released to the warehouse for further processing. For example, the customer who is placing the order might not be ready to commit to it, or critical data that is required in order to process the order might be missing.

On the order entry page, the call center user can put an order on hold by using the **Order holds** option on the **Sales order** tab of the order entry menu. Alternatively, the user can select the **Hold** menu item on the **Sales order summary** page that appears when he or she selects **Complete** on a call center sales order.

In both cases, the **Order holds** page appears. The user can then select **New** to create a hold for the order. In the **Hold code** field, the user should select the code that best describes the reason for the hold. In the **Reason code** field, the user can optionally select an additional code to provide a second level of description of the hold.

On the **Notes** FastTab, in the **Hold Notes** field, the user can enter additional, free-form notes to provide additional context or information about the hold. These notes can help other users who review or work with the hold order later.

After the hold information is entered and saved, the user can close the **Order holds** page. The user is then returned to the sales order entry page. If no further actions are required on the sales order, the user can close the sales order entry page.



If the **Enable order completion** flag is turned on in the call center channel, payment doesn't have to be applied to an order that is put on hold. By contrast, for a sales order that isn't put on hold, users can't leave the sales order entry page until payment is applied. Of course, payment will be required before the order hold is released.

Additionally, call center users can put a manual fraud hold on orders that are suspicious for some reason. Orders can also be put on hold automatically when they match active fraud criteria and rules. For more information on about this type of order hold, see [Set up fraud alerts](#).

## Viewing and managing orders that are on hold

### Viewing hold information for a single sales order

On the **Customer service** page, users can visually identify orders that are on hold, because the order lines are highlighted in a specific color. This color is defined by the **On Hold sales order status** field on the **Accounts receivable parameters** page.

#### NOTE

If the line is selected on the page, the highlight color isn't visible.

Users can also view detailed status information for a sales order to learn whether the order is on hold. The detailed status information can be accessed from the **All sales orders** or **Customer service** page. If an order is on hold, the **Do not process** flag is set for it, and the **Detailed status** field shows a status of either **On Hold** or **Fraud Hold**, depending on the scenario.

To view the details of an individual order hold, users can open a detailed view of the **Order hold** page from the **Customer service** page, by using the **Options** menu for the selected order. Users can also access this view from the **All sales order** page, by selecting the **Order holds** menu item on the **Sales order** tab.

### Viewing all orders that are on hold

To view all orders that have been put on manual or automatic hold, go to **Retail and Commerce > Customers > Order holds**.

The **Order holds** workbench provides a list view of all orders that are on hold because of manual or fraud-related hold actions. By taking advantage of the standard filtering and sorting options on the page, users can create views that let them work with or manage specific hold codes that they are responsible for reviewing. The **Order holds** workbench also indicates the number of days that an order has been on hold. This information can help users prioritize the queue.

To get a more detailed view of the orders that are on hold, users can click the **Order hold** option on the menu. This view provides information about the customer, any notes that have been applied to the order, customer, or hold action. The view also provides details about the reason for an automatic hold, if the order was put on hold because it matched a fraud rule.

From both the list view and the detailed view of the **Order holds** page, users can view or edit additional order-related information, such as payments, totals, and notes.

The options on the **Hold checkout** tab might be useful if multiple users in your company work on the hold queue at the same time. By selecting the **Check out** option, users can indicate that they are working to review and investigate the order hold. In this way, other users don't waste time by trying to do the same work. From the detailed view of the **Order holds** page, users can view information about the checkout date and time, and the user who checked out the hold record.

After a hold record is checked out, only the user who checked it out can clear the checkout. This restriction is intended to prevent users from taking records that other users are already working on. To release an order back to the queue so that other users can work on it, the user who checked out the record selects the **Clear**

checkout option.

#### NOTE

The hold isn't released when the checkout is cleared.

In some situations, such as when a user is out sick or has left the company, records that are checked out to that user might have to be reassigned to another user. A manager can reassign records by using the **Override checkout** option.

## Releasing orders that are on hold

In both the list view and the detailed view of the **Order holds** page, the **Clear hold** tab contains the options that are used to release an order hold. Use the **Clear holds** option to release an order from the selected hold code.

Call center orders require payment. Therefore, a hold can't be fully cleared if payment hasn't been applied to the order. On the **Call center parameters** page, on the **Holds** tab, make sure that the **Submit when cleared** parameter is turned on. This setting helps guarantee that a cleared hold order goes through the correct order submission logic to validate and authorize payments. If payments are missing, the user receives an error, and the hold code isn't cleared.

If the **Submit when cleared** parameter hasn't been set, users should select the **Clear and submit** option on the **Clear hold** menu to help guarantee that the order goes through all the required payment validations. If order submission fails when the **Enable order completion** flag is turned on in the call center channel, the order is released from its hold status, but the **Do not process** flag remains set. Therefore, the order isn't released to the warehouse until correct payments have been applied and validated.

If users want to clear a hold but make additional changes to the order before it's released for further processing, they can select the **Clear and modify** option. This option removes the hold code and opens the sales order details so that users can make additional changes to the order. Users can also apply payment and submit the sales order through payment validation logic when the **Enable order completion** flag is turned on in the call center channel.

## Reporting options

Go to **Retail and Commerce > Inquiries and reports > Call center reports > Order holds report** to run a report about order holds by date range, hold code, or other related criteria.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Payment methods

2/18/2021 • 3 minutes to read • [Edit Online](#)

Each payment type that a retailer accepts must be configured when the system is set up. This article describes the payment types that you can set up and describes the process for setting them up.

Retailers can accept various types of payment in exchange for the products and services that they sell. Although cash is the most common form of payment, retailers can also receive payment in the form of checks, cards, vouchers, and so on. Each payment type that the retailer accepts must be configured in Dynamics 365 Commerce when the system is set up. The following list describes each payment type that can be set up:

- **Cash** – Money in the physical form of currency, such as banknotes and coins. This currency can be either the company currency or the store's local currency.
- **Check** – A negotiable instrument that instructs payment of a specific amount of a specific currency, and that is drawn on a specific bank. A check is typically valid either indefinitely or for six months after the date of issue, unless another period of validity is specified. This period varies, depending on the bank that the check is drawn on. There are various kinds of checks, such as order checks, counter checks, bearer checks, and account payee checks. You can set up checks as a payment method for each store. Checks can be accepted in the currency that is defined at either the company level or the store level. You must set up checks as a payment method before you can accept a check as payment in a store.
- **Currency** – The primary form of payment other than the company's default currency. Coins and paper money are both forms of currency. The currency payment method represents all currency that is used. Before you can use this payment method, you must set up currencies and specify exchange information for the currencies.
- **Card** – All the kinds of cards that are used, such as debit cards and credit cards. It's a good idea to set up one card payment method at the organization level, to represent every kind of card. Then, at the store level, set up a payment method for each card or set of cards that is processed by using the same settings. You must set up the manufacturer cards that are available in the market, such as debit cards and credit cards, before you can accept the cards as payment in a store.
- **Credit memo** – Credit memos that are issued or redeemed at the point of sale. The credit memo can be a credit or a return credit memo that is issued against a return sale. If credit memos are only partially redeemed, the program issues a new credit memo for the new balance. The new credit memo has a new number. A credit memo can be used only one time, and the system keeps a record of all the numbers that are used. The record can be viewed on the **Credit memo table** page. A customer can't redeem more than the value of the credit memo.
- **Gift Card** – Gift cards that are issued and redeemed at the point of sale. Overpayment isn't allowed on gift cards. All gift cards should have card number mappings.
- **Customer account** – Payments that can be charged to a customer account at the register at the time of the sale. You can also use this payment method to collect sales information or customer-specific discounts when the customer makes a payment by using another payment method. In this case, you must set up customer-specific information.
- **Loyalty points** – The points that customers accumulate through loyalty programs. If you create loyalty programs, customers can earn points and then redeem them in various ways. For example, in some loyalty programs, customers can redeem loyalty points in the form of a discount or even use them as a form of payment.

To set up payment methods, you must complete the following tasks.

1. Set up payment methods for an organization. Create the payment methods that are accepted by the whole

organization.

2. Create organization-wide card types and card numbers. If credit cards or debit cards are accepted, you must create one payment method for cards, and then create the organization-wide card types and card numbers.
3. Set up store payment method. Associate payment methods with each store, and then enter the store-specific settings for each payment method.
4. Set up card payment methods for stores. For any card payment methods that the store accepts, complete the card setup.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Payment methods in call centers

2/18/2021 • 5 minutes to read • [Edit Online](#)

In Dynamics 365 Commerce, the configuration of the call center channel includes a setting that is named **Enable order completion**. This setting helps guarantee that all orders that users of the channel create are released to order processing only if they have a prepaid or pre-authorized payment that is within approved tolerances. If the **Enable order completion** setting is turned on, call center users can enter payments against sales orders for customers by using the payment processing features of Call center. If the setting is turned off, call center users can't use the Call center payment processing features, but they can still apply prepayments to sales orders by using standard Accounts receivable functionality.

As part of the channel configuration, a company can define the methods of payment that are allowed for a call center channel. The call center channel uses the same payment methods that are defined for the store channels.

To configure the payment methods for a call center channel, go to **Retail and Commerce > Channels > Call centers > All call centers**, and then, on the **Set up** menu, select the **Payment methods** option.

When you create a payment method, there are five payment method functions that you can assign.

FUNCTION	DESCRIPTION
Normal	Use the <b>Normal</b> function on your payment method when you define payment methods such as cash or vouchers. When these types of payments are applied to a sales order in the call center, the <b>Prepay</b> flag will default to <b>Yes</b> . This will immediately post a prepayment voucher to the customer account when this order is submitted. Users may change the <b>Prepay</b> flag to <b>No</b> if desired so that the payment voucher is not created until invoice posting. The prepayment voucher is posted in the customer transaction history, where it will be systematically settled against the invoice for the sales order.
Check	Use the <b>Check</b> function when you define a bank check instrument as a form of payment. When this type of payment is applied to a sales order, the user must enter the customer's check number as part of the payment application processing. Check payments are always treated as prepayments when they are applied and payment vouchers are created immediately upon order submission. These prepayment vouchers will be systematically settled against the invoices that are created for the order.

FUNCTION	DESCRIPTION
Cards	Card payment types represent any type of payment that requires entry of a card number that has been defined on the customer's payment card. Examples include credit cards and gift cards. When you configure these types of payments, you must use the <b>Card setup</b> menu to define the card IDs that are associated with this payment method. At the time of order entry, users can indicate whether the card payment will be prepaid, by using the <b>Prepay</b> option that appears on the payment entry page. Unless the business requires prepayments, the typical flow of a true credit card payment is a two-step process, where authorization is obtained at the time of order entry, and then payment is settled and collected from the customer's card at the time of invoicing. For gift card payments, prepayment is recommended, because the gift card balance should be reduced immediately so that the customer can't apply that same value somewhere else.
Customer	The <b>Customer</b> function on a payment method implies that the payment will be applied to the customer's credit limit or put "on account." In Commerce, a customer can be assigned a credit limit that can be validated at the time of order entry. Payments that are made by using a payment method that is linked to the <b>Customer</b> function create a liability against the customer's account. Then, when the sales order is invoiced, a balance due is shown. In these situations, customers typically send a payment, according to terms that have been given. Alternatively, a previous open credit voucher on the customer's account can be applied to settle the balance that is due. Note that even if you define this payment method, it doesn't appear among the payment selection options in call center order entry unless the <b>Allow on account</b> flag is set on the customer record for the customer that you're working with. This flag is found on the <b>Payment Defaults</b> tab of the customer record.
Tender Remove/Float	The <b>Tender Remove/Float</b> function isn't used by the call center. It's applicable only when you define the methods of payment that the point of sale (POS) application uses in a store channel.

As methods of payment are defined, they should be linked to a ledger or bank account. If you omit this step, users will receive errors when they try to save the payment type.

## Refund payment methods

For refund processing scenarios, Call center also uses some of the payment methods that are defined in Accounts receivable. To configure these payment methods, go to **Retail and Commerce > Channel setup > Call center setup > Call center refund methods**. You must complete this configuration to process refund checks to customers. For example, if a customer originally paid for an order by using cash or a check, the user might want to send the customer a refund check through Accounts receivable. In this case, the cash and check payment types in the call center must be mapped to the correct payment method in Accounts receivable to help guarantee that the refund is correctly processed.

Additionally, if a user is processing a return order as a call center user in Commerce, but he or she can't link the return to an original sale, the **Return** payment method must be defined in the Call center parameters. Go to **Retail and Commerce > Channel setup > Call center setup > Call center parameters**, and then, on the

**RMA/Return** tab, in the **Payment method** field, make sure that a payment method is defined. The payment method will be the payment method that is used for refunds. Typically, it will be defined as either a check method or a customer account method.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Refund payment processing in call centers

2/18/2021 • 11 minutes to read • [Edit Online](#)

This topic explains how payment refunds are generated through call centers when returns are created, or when orders or order lines are canceled.

A user who creates a return order for a customer as a call center user in Microsoft Dynamics 365 Commerce headquarters uses the **Return order** page to create the initial return materials authorization (RMA). The RMA defines the products that the customer wants to return or exchange, and it creates a linked return sales order that has an order type of **Returned order**. This linked returned order is used to track the posting of the returned inventory and any credit notes or payment refunds that are posted.

If the **Enable order completion** option is set to **Yes** for the call center channel, the call center user who creates the RMA must run the order completion processing flow by selecting **Complete** on the **Return order** page. The **Complete** function provides a calculated return summary that outlines the refund amount that is due. Additionally, when it's correctly configured, it systematically creates a refund payment line against the returned order.

Call center logic determines the payment method for the refund payment line, based on the payment method that was used for the original order. If the return order that is created isn't linked to an original order, a default payment method that is taken from a system parameter is applied.

## How a call center determines which payment method to apply to a return order

The call center uses the payment method of the original order to determine the payment method that should be applied to a return order. Here is how this process works for the following original payment methods:

- **Normal** (cash) or **Check** – When a return order that is created references an original order that was paid for by using the normal (cash) or check payment type, the call center application references configurations on the **Call center refund methods** page. This page enables organizations to define, by order currency, how refunds are issued to customers for orders that were originally paid for by using the normal or check payment type. The **Call center refund methods** page also enables organizations to select whether a system-generated refund check is sent to the customer, or whether a customer account credit is created against the internal customer account balance. In these scenarios, call center logic references the currency of the return order and then uses the **Retail payment method** value for that currency to create a refund payment line on the return sales order. Later, an accounts receivable (AR) customer payment journal that uses the mapped AR payment method is linked to the currency.

The following illustration shows the configuration for a scenario where a customer returns products from a sales order that is linked to the USD currency, and that was originally paid for by using the normal or check payment type. In this scenario, a refund will be issued to the customer through a system-generated refund check. The REF-CHK AR payment method has been configured as a refund check payment type.



Currency ↑	Retail payment method	AR payment method
EUR	4	REFUND
USD	2	REF-CHK

- **Credit card** – When a return order that is created references an original order that was paid for by using a credit card, call center logic for refund payments applies the same original credit card to the return order.
- **Loyalty card** – When a return order that is created references an original order that was paid for by using a customer loyalty card, call center logic for refund payments applies the refund to the same loyalty card.
- **Gift card (internal)** – When a return order that is created references an original order that was paid for by using a gift card that was issued from Dynamics 365 Commerce (internal gift card functionality), call center logic for refund payments applies the refund to the same original gift card number.
- **Gift card (External)** – When a return order that is created references an original order that was paid for by using an external third-party gift card, call center logic for refund payments applies the default return payment method that is defined on the **RMA/Return** tab of the **Call center parameters** page.

If the original order payment type is unknown for any reason, or if multiple payment methods were used to pay for the original order, call center logic applies the default return payment method that is defined on the **RMA/Return** tab of the **Call center parameters** page.

The following illustration shows the **Payment method** field on the **RMA/Return** tab of the **Call center parameters** page.

#### NOTE

The refund processing rules that are described earlier also apply to orders or order lines that a call center user cancels in Commerce headquarters. If the cancellation of an order or specific order lines causes any overpayments, the same rules will be used to generate refund payment lines.

Typically, a return order goes through a standard process, where inventory is received (or scrapped), a packing slip is posted against the return order, and then an invoice posting process is run for the return sales order. The return sales order is linked and systematically generated as part of the process of creating the return order. In

typical scenarios, payment refunds aren't issued to customers until the invoice for the return sales order is posted.

### What happens when an invoice is posted on a return sales order

The following scenarios explain what happens when an invoice is posted on a return sales order:

- If the refund payment on the return order is for a credit card, additional logic is invoked when the invoice is posted. This logic calls the payment processor to refund payment to the customer's credit card. A refund customer payment voucher is also created and systematically posted against the customer's account. This payment journal will be settled against the return order credit note voucher.
- If the refund payment that must be issued is for the check payment type, a customer payment voucher that uses the AR payment method is created, and must be manually posted or printed before the payment voucher can be posted against the customer account. To process the refund check, users can use either the **Customer payment journal** page in Accounts receivable or the specialized **Refund check processing** page in Retail and Commerce.
- If the refund payment that must be issued is for the internal gift card or loyalty card payment type, when the return order is invoiced, the refund payment voucher is created and posted against the customer account. This invoicing step also adds the refund amount back to the customer's internally tracked gift card balance or loyalty points balance.
- If a payment method that uses the **Customer** function (for example, a customer account) is linked to the return sales order, credit limit validations are ignored when the payment is processed. No payment voucher is created or posted in this context. When a customer payment type is used on a return order, the credit note voucher that the invoice posting process creates serves as the customer credit voucher and indicates a refund to the customer's AR balance.

## Advance credit

When a user processes return orders as a call center user in a call center where the **Enable order completion** option is set to **Yes**, an exception to the previously described process for refund payment posting can occur if the call center user who is creating the return order sets the **Advance credit** option to **Yes** on the **RMA/Return** tab of the **Call center parameters** page. In this case, the payment refund occurs immediately after the return order is successfully submitted by using the **Submit** function on the **Return summary** page. The system immediately creates a prepayment customer payment voucher for the return value, even though the return sales order itself hasn't yet been invoiced. This approach can be used in situations where an organization must issue refunds to customers in advance because of customer service issues, and it doesn't want to require that returned inventory be received before the refunds are issued.

## Replacement orders

When a return order is issued, the **Replacement order** function can be used to generate a new sales order for the customer. This approach can be used in exchange scenarios. The **Replacement order** function creates another sales order for the new items that must be sent, but a cross-reference link on the **RMA/Return** tab of the **Call center parameters** page links the replacement order, the RMA, and the returned sales order.

When payments on a replacement order are processed, organizations have two options:

- Refund the customer for the return order, based on the original payment method, and then collect a separate payment for the replacement order. No additional configuration is required to use this option.
- Set the **Apply credit** option to **Yes** on the **RMA/Return** tab of the **Call center parameters** page. In this case, a customer payment method is systematically applied to both the return order and the replacement order. This option can help prevent any external refund payment from being issued. It also helps prevent any payment processing on the transaction. It can be useful in situations where an even exchange is being processed, and the organization prefers to use the credit voucher that is generated when the return order is

invoiced to pay for the invoice that is generated by the replacement order. When the **Apply credit** option is set to **Yes**, the organization must manually settle the credit note against the replacement order's invoice after both those financial documents have been generated.

A setting of **Yes** for the **Apply credit** option is applicable only when the return order will be linked to a replacement order. In this case, the customer payment method that will be used to systematically pay for the return and the exchange order is defined by the **Apply credits payment method** field on the **RMA/Return** tab of the **Call center parameters** page. Only a payment of the **Customer** function payment type can be selected in this field.

#### NOTE

For a return order that has no linked replacement order, a setting of **Yes** for the **Apply credit** option will have no effect on the return order payment logic, because this setting applies only to replacement orders.

**Call center parameters**

Set up parameters for order returns and exchanges

**RETURN PAYMENT METHODS**

Payment method: 4

Apply credits payment method: 4

Allow payment override: No

**EXCHANGES**

Source code for exchange: Original

Default exchange source code: Mail

Allow expired source code: Yes

**RETURN ORDER DEFAULT VALUES**

Pending return hold: Pending re

Exchange balance hold: Pending ex

Apply credit: No

Advance exchange: Release exchange

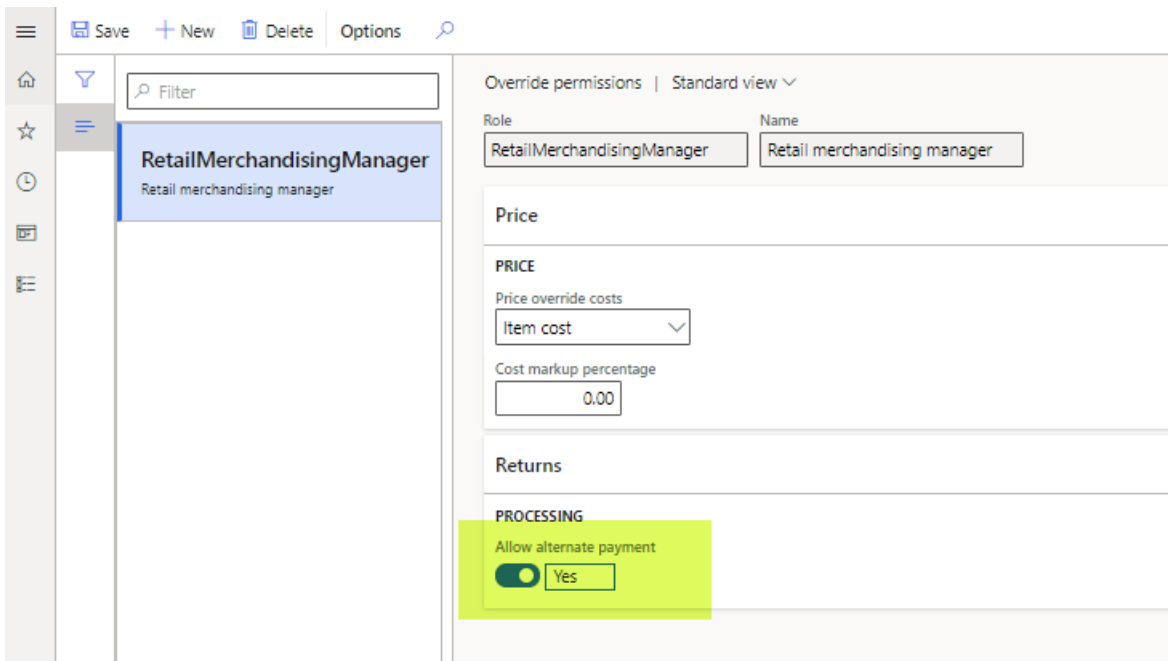
Advance credit: No

#### IMPORTANT

If users who create replacement orders plan to use the **Apply credit** option, they should not run the **Complete** function on the return order before they set the **Apply credit** option to **Yes**. After the **Complete** function is run, the refund payment is calculated and applied to the return sales order. Any attempt to set the **Apply credit** option to **Yes** after a refund payment has already been calculated and applied won't trigger a recalculation of the refund payment, and the payment method that is selected in the **Apply credits payment method** field won't be applied. If the **Apply credit** option must be used in this context, the user must delete the replacement order and the RMA, and then start over and create a new RMA. This time, the user must ensure that the **Apply credit** option is set to **Yes** before the **Complete** function is run.

## Payment overrides for call center returns

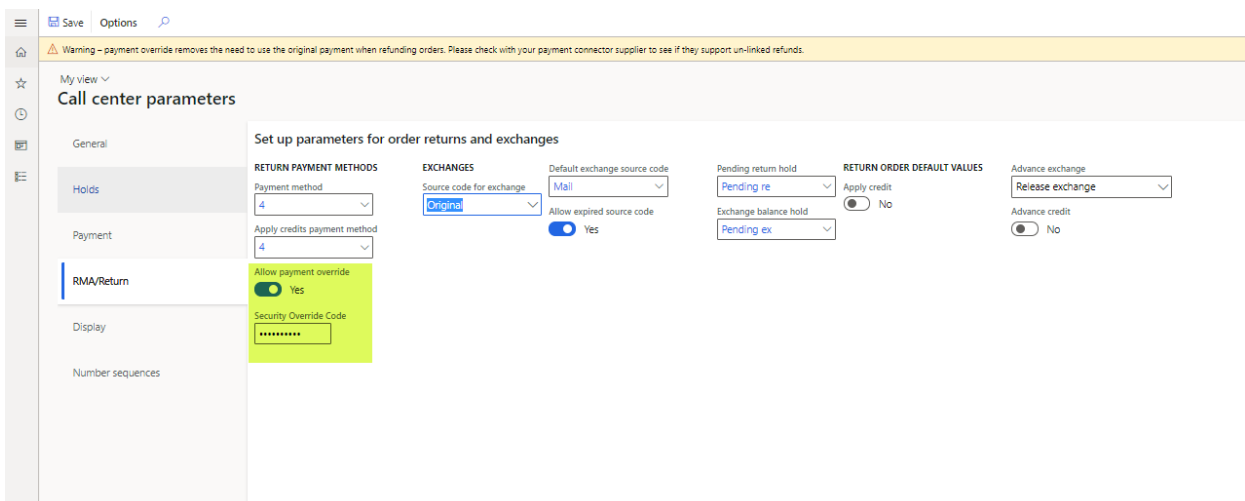
Although call center logic systematically determines the refund payment method in the manner that is described earlier in this topic, users might sometimes want to override those payments. For example, a user might edit or remove existing refund payment lines, and apply new payment lines. System-calculated refund payments can be changed only by users who have the correct override permissions. These permissions can be configured on the **Override permissions** page in Retail and Commerce. To do a refund payment override, the user must be linked to a security role where the **Allow alternate payment** option is set to **Yes** on the **Override permissions** page.



Alternatively, an organization can set the **Allow payment override** option to **Yes** on the **RMA/Return** tab of the **Call center parameters** page. In this case, a security override code must be selected in the **Security Override Code** field. The security override code is an alphanumeric code that must be externally managed, because users can't view it in Commerce headquarters after it's set. The security override code should be known by only a few key, trusted people in an organization. When the **Allow payment override** option is set to **Yes**, if any users who don't have the correct role permissions try to change the method of payment on a return order, they will have the option to enter the security override code. If they don't know it, or if a manager or supervisor can't enter it on the page for them, they won't be able to override the return payment method.

#### NOTE

If the security override code is lost or forgotten, the organization will have to reset it by defining a new security override code in the **Security Override Code** field on the **RMA/Return** tab of the **Call center parameters** page.



#### IMPORTANT

Before organizations try to override refund payments that use credit card payment types, they should verify that their credit card processor allows for unlinked returns. Many processors require that refunds be posted back to the original card. Any attempt to issue a refund to a card that has no previous captures might cause posting failures with the processor.

# Additional resources

## Payment methods in call centers

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Credit card setup, authorization, and capture

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article provides an overview of credit card authorization in Microsoft Dynamics 365 Finance. It includes information about how to set up a payment service, add a credit card to a sales order, and void an authorization.

## Setting up the credit card payment service

To use credit cards, you must set up and activate a payment service on the Payment services page. A payment service acts as a bridge between your legal entity and the bank that processes a customer's credit card charges. You must work with a credit card provider that is listed in the Payment connector field and set up an account with that provider. You must then set up the other options on the Payment services page, set up credit card types for American Express, Discover, MasterCard, and Discover on the Credit card types page, and activate the provider as the default provider. You must also follow these steps to complete your setup:

- On the Accounts receivable parameters page, specify parameters for using credit card authorizations.
- On the Terms of payment page, set up payment terms for credit cards. In the Payment type field, select Credit card.
- On the Customer credit cards page, enter credit card information for customers.

## Adding a new credit card

You can create new credit card records on the Customers page by using Customer, Set up, Credit card. You can also create credit card records when you enter sales orders on the Sales order page, by using Manage, Customer, Credit card, Register.

## Adding a credit card to a sales order

You can add a credit card to a sales order by selecting a credit card in the credit card lookup on the Price and discounts FastTab on the Sales order page. To start the authorization process, on the Action Pane, on the Manage tab, select Credit card and Authorize.

## Authorizing a credit card

When a credit card is authorized, the card number and cardholder's name are verified, and the available credit balance is confirmed. Optionally, the card verification value and the cardholder's address are verified. The customer's available credit balance is then reduced by the amount of the invoice. The payment service sends information that the credit card has been approved or declined. When the sales order is invoiced, the credit card is charged (captured) for the invoice amount.

### Card verification value

You can require the card verification value, which is sometimes referred to as the card's security code. For American Express, this is a four-digit value. For Discover, MasterCard, and Visa, it is a three-digit value.

### Address verification

Address verification information is always sent to the payment provider. You can decide how much information is required for a transaction to be accepted. Be sure to check with your provider to determine whether it accepts this information. Here are the options for address verification:

- **Always accept transaction** – Accept the transaction, regardless of address verification results.

- **Account holder** – Compare the cardholder's name from the transaction with the credit card company's information.
- **Billing address** – Compare the cardholder's name and billing address from the transaction with the credit card company's information.
- **Billing postal code** – Compare the cardholder's name, billing address, and postal code from the transaction with the credit card company's information.

## Data support

For each credit card type that is supported, you can specify the level of data support. The level controls how much information about a transaction is transferred to the payment service. Be sure to check with your provider to determine whether it can provide this information. Here are the options for the level of data support:

- **Level 1** – Transfer the transaction date, transaction amount, and description.
- **Level 2** – Transfer all Level 1 information, plus the shipping and merchant addresses, and tax information.
- **Level 3** – Transfer all Level 2 information, plus order line information.

## Partial payments

If you ship part of an order, the amount of the partial order is captured, and the authorization, which was for the amount of the whole order, is closed. A new authorization is then submitted for the remaining amount of the order that hasn't been shipped.

## Voiding an authorization

To void a credit card authorization, you can change the method of payment to another method that doesn't have a type of Credit card.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

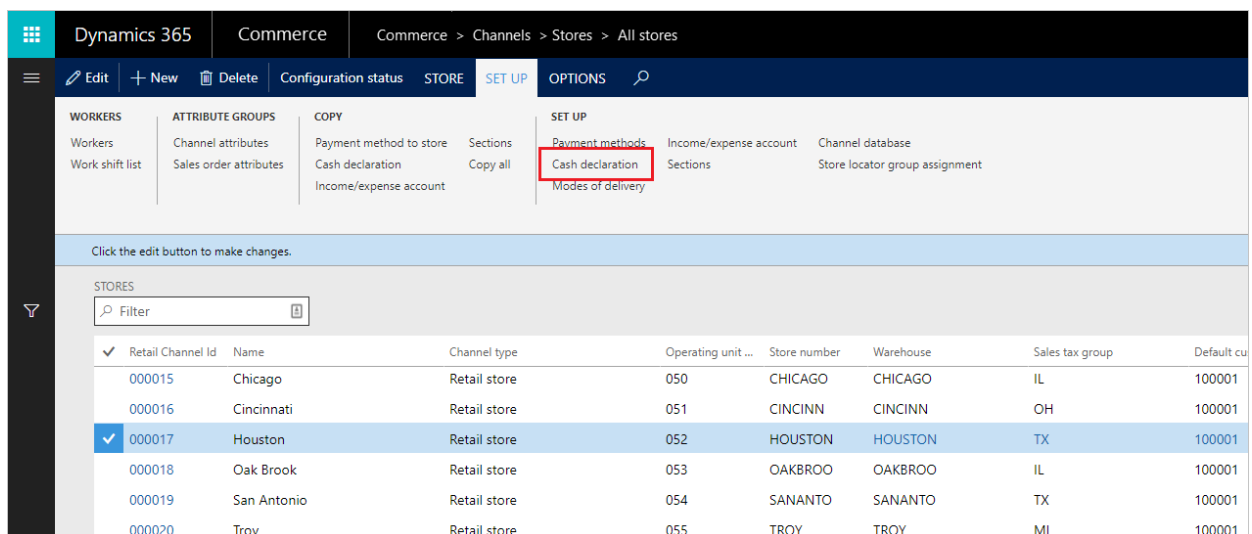
# Configure cash denominations for the point of sale (POS)

2/18/2021 • 2 minutes to read • [Edit Online](#)

Cash denominations for notes and coins can be defined in the back office to be used by cashiers, sales associates, and managers at the store from within the POS. These denominations can be used to aid in counting cash for end of day tender declarations or for quickly tendering a sale.

## Define denominations

The denominations are set up per store on the **Set up > Cash declaration** option from the store property page.



The screenshot shows the Dynamics 365 Commerce interface. The breadcrumb navigation is 'Commerce > Channels > Stores > All stores'. The 'SET UP' menu is open, and 'Cash declaration' is highlighted with a red box. Below the menu, there is a 'Click the edit button to make changes.' message and a table of stores.

✓	Retail Channel Id	Name	Channel type	Operating unit ...	Store number	Warehouse	Sales tax group	Default cu
	000015	Chicago	Retail store	050	CHICAGO	CHICAGO	IL	100001
	000016	Cincinnati	Retail store	051	CINCINN	CINCINN	OH	100001
✓	000017	Houston	Retail store	052	HOUSTON	HOUSTON	TX	100001
	000018	Oak Brook	Retail store	053	OAKBROO	OAKBROO	IL	100001
	000019	San Antonio	Retail store	054	SANANTO	SANANTO	TX	100001
	000020	Troy	Retail store	055	TROY	TROY	MI	100001

To define a denomination:

1. Click **New**.
2. Specify the type (coin or note).
3. Specify the amount (value).



Type	Amount in trans...	Currency
Coin	0.01	USD
Coin	0.05	USD
Coin	0.10	USD
Coin	0.25	USD
Coin	0.50	USD
Note	1.00	USD
Note	2.00	USD
Note	5.00	USD
Note	10.00	USD
Note	20.00	USD
Note	50.00	USD
Note	100.00	USD

## Configure the functionality profile

When paying by cash in POS, the user can use the note denominations to quickly enter the amount paid by the customer. In the functionality profile, you can configure the two options for showing the denomination in POS.

- **Greater or equal to amount due** – By default, POS will only show the note denominations that are greater than the amount due, which allows for one-touch tendering. For example, if the amount due is \$7.50, POS would show the following denominations: \$10, \$20, \$50, and \$100. Touching any of these amounts will automatically tender the sale for that amount. The \$1 and \$5 notes are not shown since these amounts are less than the amount due.
- **All denominations** – Select this option to always show all note denominations in POS, regardless of the amount due. This means that the user can use a combination of notes to reach the amount due. For example, if the amount due is \$25.00, the user can choose \$20 and \$5 to complete the sale.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure credit card processing

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through how to view the list of payment providers and how to configure a payment account for accounts receivable. This procedure uses the USRT company in demo data and is intended for Administrators and IT Professionals.

## View a list of payment providers

1. Go to Accounts receivable > Payments setup > Payment services.
2. Click View available providers.

## Configure payment account

1. Click New.
2. In the Payment service field, type a value.
3. In the Payment connector field, select an option.
4. Toggle the expansion of the Payment service account section.
5. In the Environment: field, type 'PROD'.
6. Click Credit card types.
7. In the Payment journal field, click the drop-down button to open the lookup.
8. In the list, click the link in the selected row.
9. Click Add.
10. In the Currency field, type a value.
11. In the list, find and select the desired record.
12. In the Payment journal field, click the drop-down button to open the lookup.
13. In the list, click the link in the selected row.
14. Click Add.
15. In the Currency field, type a value.
16. In the list, find and select the desired record.
  - You can repeat these steps for as many card types as you need.
17. In the Payment journal field, click the drop-down button to open the lookup.
18. In the list, click the link in the selected row.
19. Click Add.
20. In the Currency field, type a value.
21. Click Save.
22. Close the page.
23. Click Validate.
24. Click the Default processor for new credit cards checkbox.
25. Click Save.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Skip "change due" dialog box in POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to skip the **Change due** dialog box in the point of sale (POS) when a transaction is paid in full and there's no change due.

For payments at the POS, there is now a greater prevalence of credit usage, as most transactions don't require change to be provided to the customer. In Dynamics 365 Commerce, you can configure POS so that the **Change due** dialog is skipped unless there's actually change due back to the customer. The **Change due** dialog box will also be shown if the receipt format for gift receipts is configured to print **As required**. When gift receipts are configured to print **As required**, the option to print a gift receipt is included in the **Change due** dialog box, overriding the **Skip change due** configuration.

## Configure property to skip Change due dialog box

The property you need to configure to skip the **Change due** dialog box is found in the **Functionality profile** level, which is a store-level setting.

To configure the property, follow these steps.

1. Search for the **Functionality profile** for the store where the dialog box should be skipped.
2. Open the functionality profile for the target store and select **Edit**.
3. Expand the **Functions** FastTab. Under the **Terminal** subheading, in the **Change due** field, select **Skip when zero**.
4. Select **Save**, and then run the **1070** channel configuration distribution schedule.
5. After the changes are synchronized, sign out of the POS and then sign back in. Make a transaction that has no change due to the customer to verify that the **Change due** dialog box isn't displayed.

## Additional resources

[Payment methods](#)

[Credit card setup, authorization, and capture](#)

[Configure cash denominations for the point of sale \(POS\)](#)

[Configure credit card processing](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Payment Connector for Adyen

2/18/2021 • 29 minutes to read • [Edit Online](#)

This topic provides an overview of the Microsoft Dynamics 365 Payment Connector for Adyen. It includes a comprehensive list of supported features and functionality, a guide to setting up and configuring the connector, troubleshooting information, and descriptions of some common issues.

## Key terms

TERM	DESCRIPTION
Payment connector	An extension that facilitates communication between Microsoft Dynamics 365 Commerce (and associated components) and a payment service. The connector that is described in this topic was implemented by using the standard payments software development kit (SDK).
Card present	Refers to payment transactions where a physical card is presented and used on a payment terminal connector to the Dynamics 365 Point of Sale.
Card not present	Refers to payment transactions where a physical card is not present, such as e-Commerce or Call Center scenarios. In these scenarios, the payment-related information is entered manually either on an e-Commerce website, a Call Center flow, or on the point-of-sale or payment terminal.

## Overview

This topic includes the following main sections to help you evaluate and set up the Dynamics 365 Payment Connector for Adyen.

- Supported features, functionality, versions, and terminals – This section describes the set of features and functionalities that the Dynamics 365 Payment Connector for Adyen supports.
- Sign up with Adyen – This section explains how to sign up for a merchant account with Adyen.
- Setup and configuration – This section explains, in detail, how to set up and configure the Dynamics 365 Payment Connector for Adyen across the point of sale (POS), call center, and e-Commerce channels.

## Supported features, functionality, versions, and terminals

The out-of-box Dynamics 365 Payment Connector for Adyen uses the standard payments SDK. Therefore, it doesn't have special capabilities that aren't also available to other payment connectors.

### Supported versions

#### Microsoft Dynamics 365 supported versions

The first-party out-of-box Dynamics 365 Payment Connector for Adyen is supported in Microsoft Dynamics 365 for Finance and Operations version 8.1.3 (January 2019) or later, and in Microsoft Dynamics 365 Retail version 8.1.3 or later. However, third parties can still develop other payment connectors for Adyen for earlier versions of Microsoft Dynamics 365.

#### Supported Adyen Firmware versions

The list below describes the minimum and maximum Adyen firmware versions that are supported for each version of the Microsoft Dynamics 365 Retail POS.

- [8.1.3](#)
- [10.0.7](#)
- [10.0.8](#)
- [10.0.9](#)
- [10.0.10](#)
- [10.0.11](#)
- [10.0.12](#)
- [10.0.13](#)
- [10.0.14](#)
- [10.0.15](#)
- [10.0.16](#)

### Dynamics 365 Retail POS version 8.1.3

MINIMUM ADYEN FIRMWARE VERSION	MAXIMUM ADYEN FIRMWARE VERSION
adyen_v1_35p15	adyen_v1_35p15

#### NOTE

Adyen may release minor version updates after Microsoft has tested the major version. As long as a major version is supported, it's okay to have minor version updates within the same major version. These updates are normally very targeted fixes and don't meet the bar for full retesting, as long as the same major firmware version was previously tested. Updates shouldn't exceed the maximum Adyen firmware version listed in documentation.

Migrating from a Adyen firmware version earlier than version 53 to version 53 requires POS KB [4577957](#) for monthly updates of Commerce, versions 10.0.11 through 10.0.14. If one of those versions is in use and doesn't include the hotfix, post-upgrade of the payment terminal will only allow payments via NFC. Applying the hotfix to the POS resolves this issue. If the POS version is older than version 10.0.11, file a support request noting that a fix for KB [4577957](#) is required for an out of service MPOS.

### Supported payment terminals

The Dynamics 365 Payment Connector for Adyen takes advantage of the device-agnostic [Adyen Payment Terminal API](#). It supports all payment terminals that this application programming interface (API) supports. For a complete list of supported payment terminals, visit the [Adyen POS terminals](#) page.

### Supported payment instruments

#### Supported debit and credit cards

BRAND	VARIANT	CARD PRESENT	E-COMMERCE	CALL CENTER
MasterCard	Credit	✓	✓	✓
MasterCard	Debit	✓	✓	✓
MasterCard	Alpha Bank Bonus	✓	✓	✓
MasterCard	Apple Pay	✓		
MasterCard	Samsung Pay	✓		

BRAND	VARIANT	CARD PRESENT	E-COMMERCE	CALL CENTER
MasterCard	Maestro	✓	✓	✓
MasterCard	Maestro Samsung Pay	✓		
MasterCard	Maestro UK	✓	✓	✓
VISA	Credit	✓	✓	✓
VISA	Debit	✓	✓	✓
VISA	Alpha Bank Bonus	✓	✓	✓
VISA	Android Pay	✓		
VISA	Apple Pay	✓		
VISA	Samsung Pay	✓		
VISA	VISA Checkout	✓	✓	✓
VISA	VISA Dankort	✓	✓	✓
VISA	VISA Hipotecario	✓	✓	✓
VISA	VISA Aravia Card	✓	✓	✓
AMEX	Credit	✓	✓	✓
AMEX	Debit	✓	✓	✓
AMEX	Android Pay	✓		
AMEX	Apple Pay	✓		
AMEX	Samsung Pay	✓		
AMEX	AMEX Commercial	✓	✓	✓
AMEX	AMEX Consumer	✓	✓	✓
AMEX	AMEX Corporate	✓	✓	✓
AMEX	AMEX Small Business	✓	✓	✓
Discover	Standard	✓	✓	✓
Discover	Android Pay	✓		
Discover	Apple Pay	✓		

BRAND	VARIANT	CARD PRESENT	E-COMMERCE	CALL CENTER
Discover	Samsung Pay	✓		
Diners	Standard	✓	✓	✓
Dineromail	Standard	✓	✓	✓
JCB	Standard	✓	✓	✓
Union Pay*	Standard	✓		
Interac Debit*	Standard	✓		

\*Interac and Union Pay recurring card tokens aren't provided by Adyen, so they can't be supported for card not present transactions.

#### Supported gift cards

SCHEME	CARD PRESENT	CARD NOT PRESENT
Givex	✓	✓
SVS	✓	✓

To support these external gift card schemes through the Dynamics 365 Payment Connector for Adyen, you must complete additional steps. For more information, see [Support for external gift cards](#).

#### Supported wallets

SCHEME	CARD PRESENT	CARD NOT PRESENT
Alipay	Support will be added in a future release.	No
WeChat	Support will be added in a future release.	No

#### Supported card present input methods

INPUT METHOD	SUPPORTED	NOTES
Dip	✓	
Swipe	✓	
Tap	✓	
Manual Entry through POS UI.		Not supported at this time
Manual Entry through Payment Terminal.	✓	Supports manual entry of credit, debit, and gift cards with pin entry.

#### Supported card present countries

The following countries have Commerce components available and card present support from Adyen. For current international availability of Commerce, visit the [International availability page](#).

COUNTRY	SUPPORTED
Australia	✓
Austria	✓
Belgium	✓
Canada	✓
Czech Republic	✓
Denmark	✓
Estonia	✓
Finland	✓
France	✓
Germany	✓
Hong Kong SAR	✓
Hungary	✓
Iceland	✓
Ireland	✓
Italy	✓
Latvia	✓
Lithuania	✓
Malaysia	✓
Netherlands	✓
Norway	✓
Poland	✓
Singapore	✓
Switzerland	✓
Spain	✓
Sweden	✓



COUNTRY	SUPPORTED
Switzerland	✓
United Kingdom	✓
United States	✓
Brazil	Future release

**Supported card not present countries**

The following countries are supported by Adyen for card not present transactions. [Contact Adyen](#) for details about support for a specific country. For current international availability of Commerce, visit the [International availability page](#).

COUNTRY
Argentina
Armenia
Australia
Austria
Bahrain
Belgium
Brazil
Bulgaria
Canada
Chile
China
Colombia
Croatia
Cyprus
Czech Republic
Denmark
Egypt
Estonia

**COUNTRY**

Finland

France

Georgia

Germany

Gibraltar

Greece

Guernsey

Hong Kong SAR

Hungary

Iceland

India

Indonesia

Ireland

Isle of Man

Israel

Italy

Japan

Jersey

Korea

Kuwait

Latvia

Lithuania

Luxembourg

Malaysia

Malta

COUNTRY
Mexico
Morocco
Netherlands
New Zealand
Norway
Peru
Poland
Portugal
Qatar
Romania
Saudi Arabia
Serbia
Singapore
Slovakia
Slovenia
South Africa
Spain
Sweden
Switzerland
Taiwan
Tanzania
Thailand
Turkey
United Arab Emirates (UAE)
United Kingdom

## COUNTRY

United States of America including Puerto Rico

### Supported Dynamics 365 payment features

The following table shows the set of features that the Dynamics 365 Payment Connector for Adyen supports. These features use enhancements that were introduced in the payments SDK and some components in December 2018. They aren't exclusive to the Dynamics 365 Payment Connector for Adyen. For more information about how to uptake these enhancements for a different payment connector, see [Create an end-to-end payment integration for a payment terminal](#).

SCHEME	CARD PRESENT	CARD NOT PRESENT
<a href="#">Cash Out Gift Card Balance</a>	✓	
<a href="#">Duplicate Payment Protection</a>	✓	
Omni Channel Tokenization	✓	✓
Linked Refunds	✓ (Starting with 10.0.1)	✓ (Starting with 10.0.1)
<a href="#">Save online payments</a>		✓ (Starting with 10.0.2)
<a href="#">External gift cards for call center and e-commerce</a>	✓ (Starting with 10.0.10)	
<a href="#">SCA payment redirect</a>		✓ (Starting with 10.0.12)
<a href="#">Dedicated payment terminals and prompts for a printer and cash drawer</a>	✓ (Starting with 10.0.12)	
<a href="#">SDK-level tipping support through the Adyen connector</a>	✓ (Starting with 10.0.14)	

## Sign up with Adyen

To use the Dynamics 365 Payment Connector for Adyen, you must have a separate agreement with Adyen. To learn more about Adyen's services, or to create a test merchant account, visit the [Adyen website](#).

## Setup and configuration

### NOTE

These instructions assume that you've already signed up for a merchant account with Adyen, and that you have access to the Adyen merchant dashboard.

### Prerequisites

The following prerequisites must be completed before payments can be configured in any channel.

#### Set up a processor for new credit cards

To process payments across point of sale (POS) terminals, a call center, or e-Commerce, you must configure a new default payment processor for new credit cards. Follow these steps to configure a default payment processor.

1. Sign in to Headquarters, and go to **Accounts receivable > Payments setup > Payment services**.
2. On the Action Pane, select **New**, and then, on the **Setup** tab, enter the following information.

FIELD	DESCRIPTION	SAMPLE VALUE
Payment service	Enter the name of the payment service to configure.	Adyen Payment Service
Payment connector	Select the payment connector to use for new credit card payments.	Dynamics 365 Payment Connector for Adyen
Test mode	For the Adyen connector, in production and test environments you should set this field to <b>false</b> .	false
Default processor for credit cards	Specify whether this payment processor should be the default processor that's used for new credit cards.	Yes
Bypass payment processor for zero transactions	Specify whether this payment processor should be skipped for transactions that have a 0 (zero) amount.	Yes

3. On the **Payment service account** tab, enter the following information.

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Assembly Name	Auto populated name of the assembly for the Dynamics 365 Payment Connector for Adyen.	Yes	Yes	<i>Binary name</i>
Service account ID	Auto populated unique identifier for the setup of the merchant properties. This identifier is stamped on payment transactions and identifies the merchant properties that downstream processes (such as invoicing) should use.	Yes	Yes	<i>Guid</i>

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Version	Enter the version of the Dynamics 365 Payment Connector for Adyen to use. "V002" should be used for all new implementations, as it leverages a newer Adyen API for card not present payments and is required for <a href="#">SCA support</a> .	Yes	Yes	"V001"/"V002"
Gateway environment	Enter the Adyen gateway environment to map to. The possible values are <b>Test</b> and <b>Live</b> . You should set this field to <b>Live</b> only for production devices and transactions.	Yes	Yes	Live
Optional Domain	The optional domain is required for Live environments and should be obtained by contacting Adyen. This is the unique identifier for your Live environment in the form <b>[random]-[company name]</b> . This is present as the prefix inside the API URLs under <b>Account &gt; API URLs</b> in your company's Live account on the Adyen Customer Area portal. For additional details, see <a href="#">Live endpoints</a> .	Live only	No	Contact Adyen
Merchant account ID	Enter the unique Adyen merchant identifier. This value is provided when you sign up with Adyen as described in the <a href="#">Sign up with Adyen</a> section.	Yes	No	MerchantIdentifier

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Terminal architecture	This field must be set to <b>Cloud</b> for the <span style="border: 1px solid gray; padding: 2px;">Payment service account</span> .	Yes	Yes	Cloud
Local Password phrase	This field is used only for the POS payment terminal integration and should be left blank.	No	Yes	<i>Leave this field blank.</i>
Local Key Identifier	This field is used only for the POS payment terminal integration and should be left blank.	No	Yes	<i>Leave this field blank.</i>
Local Key Version	This field is used only for the POS payment terminal integration and should be left blank.	No	Yes	<i>Leave this field blank.</i>
Local Cryptor Version	Enter the Adyen cryptor version to use when you interact with the Adyen gateway. You should set this field to 1.	Yes	Yes	1
Cloud API Key	Enter the Adyen cloud API key. You can obtain this key by following the instructions on the <a href="#">How to get the API key</a> page on the Adyen website.	Yes	No	abcdefg

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Supported Currencies	Enter the currencies that the connector should process. In card-present scenarios, Adyen can support additional currencies through <a href="#">Dynamic Currency Conversion</a> after the transaction request is sent to the payment terminal. Contact Adyen support to get a list of supported currencies.	Yes	Yes	USD;EUR
Supported Tender Types	Enter the tender types that the connector should process.	Yes	Yes	Visa;MasterCard;Amex;Discover;Debit
Gift card provider	Enter the gift card provider that the connector should use to process gift cards. This field is case-sensitive.	No	No	"svs" or "givex"
Terminal gift card entry	<i>POS Only</i> Allows the customer to select between <b>Manual</b> or <b>Swipe</b> .	Yes	Yes	True/False
Allow saving payment information in e-commerce	<i>e-Commerce only</i> Gives signed-in users the option to save payment details for future online purchases.	Yes	Yes	True/False
Authorization stale period (days)	<i>POS Only</i> Number of days before an authorization is considered stale and should decline before going to the processor for capture.	Yes	Yes	"7"



FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Origin Key	Required when "V002" is designated for the version. You can obtain this key by following the instructions on the <a href="#">How to get an origin key</a> page on the Adyen website.			

- On the **Card verification value** tab, leave **Prompt for card verification value** and **Allow blank card verification value** set to **No**.

## POS payment terminal

### Onboard and configure an Adyen payment terminal

#### NOTE

These instructions assume that you have access to an Adyen payment terminal.

Go to the [Point of sale](#) page on the Adyen website, and follow the instructions to onboard your Adyen payment terminal. Skip any steps that instruct you to download Adyen-specific apps. During the onboarding process, make a note of the following information for each payment terminal. You will need this information in the [Configure the payment terminal IP address and EFT POS register number](#) section later in this topic.

- IP address of the payment terminal
- POIID (POIID is comprised of the serial number and model number of the device. It is used to uniquely identify the device.)

After the payment terminal is onboarded, sign in to the [Adyen Customer Area](#), go to the terminal that you want to configure, and make a note of the following information for each payment terminal. You will need this information in the [EFT service for local network communication](#) section later in this topic.

- Key identifier
- Key passphrase
- Key version

### Set up a Dynamics 365 POS hardware profile

- Sign in to Headquarters, and go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**.
- Select the hardware profile to add the Dynamics 365 Payment Connector for Adyen for.
- Follow the steps in the [EFT service](#) and [PIN pad](#) sections that follow.

### EFT service

The Adyen payment connector can be configured to communicate with devices via the local network or through Adyen's cloud backend. For environments with unreliable internet service, or where offline mode is required, the **Local** setup should be used. For environments with strong internet connections, or if a static IP address can't be assigned to the payment terminal, the **Cloud** architecture may work best.

#### EFT service for local network communication

- On the **EFT service** FastTab, in the **EFT Service** field, select **Payment Connector**.
- On the **Connectors** tab, select **New**, and then, in the **Connector** field, select **Dynamics 365 Payment**

**Connector for Adyen.** Make sure that the value in the **Sequence number** field is lower than the value for all other connectors.

3. In the **Connector properties** section, enter the following information.

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Assembly Name	Auto populated name of the assembly for the Dynamics 365 Payment Connector for Adyen.	Yes	Yes	<i>Binary name</i>
Service account ID	Auto populated unique identifier for the setup of the merchant properties. This identifier is stamped on payment transactions and identifies the merchant properties that downstream processes (such as invoicing) should use.	Yes	Yes	<i>Guid</i>
Version	Set to <b>V001</b> for EFT settings in the hardware profile. <b>V002</b> is required for call center and storefront only.	Yes	Yes	"V001"
Gateway environment	Enter the Adyen gateway environment to map to. The possible values are <b>Test</b> and <b>Live</b> . You should set this field to <b>Live</b> only for production devices and transactions.	Yes	Yes	Live

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Optional Domain	The optional domain is required for Live environments and should be obtained by contacting Adyen. This is the unique identifier for your Live environment in the form <b>[random]-[company name]</b> . This is present as the prefix inside the API URLs under <b>Account &gt; API URLs</b> in your company's Live account on the Adyen Customer Area portal. For additional details, see <a href="#">Live endpoints</a> .	Live only	No	Contact Adyen
Merchant account ID	Enter the unique Adyen merchant identifier. This value is provided when you sign up with Adyen as described in the <a href="#">Sign up with Adyen</a> section.	Yes	No	MerchantIdentifier
Terminal architecture	This must be set to <b>Local</b> for local communications. For more information about the different Terminal API architectures, see the <a href="#">Introducing the Terminal API</a> page on the Adyen website.	Yes	Yes	Local
Local Password phrase	Enter the Adyen key passphrase for the payment terminal. This value is provided when you sign up with Adyen as described in the <a href="#">Sign up with Adyen</a> section.	Yes	No	keypassphrase123

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Local Key Identifier	Enter the Adyen key identifier for the payment terminal. This value is provided when you sign up with Adyen as described in the <a href="#">Sign up with Adyen</a> section.	Yes	No	mykey
Local Key Version	Enter the Adyen key version for the payment terminal. This value is provided when you sign up with Adyen as described in the <a href="#">Sign up with Adyen</a> section.	Yes	No	0
Local Cryptor Version	Enter the Adyen cryptor version to use when you interact with the Adyen gateway. You should set this field to 1.	Yes	Yes	1
Cloud API Key	Enter the Adyen cloud API key. You can obtain this key by following the instructions on the <a href="#">How to get the API key</a> page on the Adyen website.	Yes	No	abcdefg
Supported Currencies	Enter the currencies that the connector should process. In card-present scenarios, Adyen can support additional currencies through <a href="#">Dynamic Currency Conversion</a> after the transaction request is sent to the payment terminal. Contact Adyen support to get a list of supported currencies.	Yes	Yes	USD;EUR

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Supported Tender Types	Enter the tender types that the connector should process. These values are case-sensitive.	Yes	Yes	Visa;MasterCard;Amex;Discover;Debit
Gift card provider	Enter the gift card provider that the connector should use to process gift cards. This field is case-sensitive.	No	No	"svs" or "givex"
Terminal gift card entry	<i>POS Only</i> Allows the customer to select between <b>Manual</b> or <b>Swipe</b> .	Yes	Yes	True/False
Allow saving payment information in e-commerce	<i>e-Commerce only</i> Gives signed-in users the option to save payment details for future online purchases.	Yes	Yes	True/False
Authorization stale period (days)	<i>POS Only</i> Number of days before an authorization is considered stale and should decline before going to the processor for capture.	Yes	Yes	"7"
Origin Key	<i>Card not present only</i>			

4. On the Action Pane, select **Save**.

**EFT service for cloud communication**

1. On the **EFT service** FastTab, in the **EFT Service** field, select **Payment Connector**.
2. On the **Connectors** tab, select **New**, and then in the **Connector** field, select **Dynamics 365 Payment Connector for Adyen**. Make sure that the value in the **Sequence number** field is lower than the value for all other connectors.
3. In the **Connector properties** section, enter the following information.

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
-------	-------------	----------	-------------------	--------------

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Assembly Name	Autopopulated name of the assembly for the Dynamics 365 Payment Connector for Adyen.	Yes	Yes	<i>Binary name</i>
Service account ID	Autopopulated unique identifier for the setup of the merchant properties. This identifier is stamped on payment transactions and identifies the merchant properties that downstream processes (such as invoicing) should use.	Yes	Yes	<i>Guid</i>
Version	Set to <b>V001</b> for EFT settings in the hardware profile. <b>V002</b> is required for call center and storefront only.	Yes	Yes	"V001"
Gateway environment	Enter the Adyen gateway environment to map to. The possible values are <b>Test</b> and <b>Live</b> . You should set this field to <b>Live</b> only for production devices and transactions.	Yes	Yes	Live

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Optional Domain	The optional domain is required for live environments and should be obtained by contacting Adyen. This is the unique identifier for your live environment in the form <b>[random]-[company name]</b> . This is present as the prefix inside the API URLs under <b>Account &gt; API URLs</b> in your company's live account on the Adyen Customer Area portal. For additional details, see <a href="#">Live endpoints</a> .	Live only	No	Contact Adyen
Merchant account ID	Enter the unique Adyen merchant identifier. This value is provided when you sign up with Adyen as described in the <a href="#">Sign up with Adyen</a> section.	Yes	No	MerchantIdentifier
Terminal architecture	This must be set to <b>Cloud</b> for cloud communication with the payment terminal. For more information about the different terminal API architectures, see the <a href="#">Introducing the Terminal API</a> page on the Adyen website.	Yes	Yes	Cloud
Local Password phrase	This setting is used for local payment terminal communication only.	No	No	<i>leave blank</i>
Local Key Identifier	This setting is used for local payment terminal communication only.	No	No	<i>leave blank</i>

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Local Key Version	This setting is used for local payment terminal communication only.	No	No	<i>leave blank</i>
Local Cryptor Version	This setting is used for local payment terminal communication only. You can leave the default as-is, set to 1.	Yes	Yes	1
Cloud API Key	Enter the Adyen cloud API key. You can obtain this key by following the instructions on the <a href="#">How to get the API key</a> page on the Adyen website.	Yes	No	abcdefg
Supported Currencies	Enter the currencies that the connector should process. In card-present scenarios, Adyen can support additional currencies through <a href="#">Dynamic Currency Conversion</a> after the transaction request is sent to the payment terminal. Contact Adyen support to get a list of supported currencies.	Yes	Yes	USD;EUR
Supported Tender Types	Enter the tender types that the connector should process. These values are case-sensitive.	Yes	Yes	Visa;MasterCard;Amex;Discover;Debit
Gift card provider	Enter the gift card provider that the connector should use to process gift cards. This field is case-sensitive.	No	No	"svs" or "givex"



FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Terminal gift card entry	<i>POS Only</i> Allows the customer to select between <b>Manual</b> or <b>Swipe</b> .	Yes	Yes	True/False
Allow saving payment information in e-commerce	<i>e-Commerce only</i> Gives signed-in users the option to save payment details for future online purchases.	Yes	Yes	True/False
Authorization stale period (days)	<i>POS Only</i> Number of days before an authorization is considered stale and should decline before going to the processor for capture.	Yes	Yes	"7"
Origin Key	<i>Card not present only</i>			

4. On the Action Pane, select **Save**.

#### **PIN pad**

1. On the **PIN pad** FastTab, in the **PIN pad** field, select **Network**.
2. In the **Device name** field, enter **MicrosoftAdyenDeviceV001**.

#### **Set up a Dynamics 365 register**

##### **NOTE**

These instructions assume that there is a dedicated mapping between a POS register and an Adyen payment terminal. For a hardware station that is based on Microsoft Internet Information Services (IIS), go to **Retail and Commerce > Channels > Stores > All stores**, and select the store that you're setting up. Then, on the page for that store, on the **Hardware Stations** FastTab, follow the same instructions.

Payment terminals may not be used by multiple hardware stations. If a payment terminal must be shared by multiple POS devices, an IIS hardware station must be deployed to manage communications with the payment terminal.

#### **Configure the payment terminal IP address and EFT POS register number**

1. Sign in to Headquarters, and go to **Retail and Commerce > Channel setup > POS setup > Registers**.
2. Select the register to link to the Adyen payment terminal.
3. On the **POS Registers** page, on the **General** FastTab, in the **EFT** section, in the **EFT POS register number** field, enter a unique number.
4. In the **Profiles** section, in the **Hardware profile** field, select the hardware profile that you configured earlier.
5. Save your changes.
6. On the Action Pane, on the **Register** tab, in the **Hardware** group, select **Configure IP addresses**. 7a. **If using the "Local" architecture:** On the **IP address configuration** page, on the **PIN pad** FastTab, in the **IP address** field, enter the IP address of the terminal in the following format:

`https://<IP address>:8443/nexo/<POIID>`. Here, **<IP address>** and **<POIID>** are the values that you made a

note of when you onboarded the Adyen payment terminal. Here is an example:

`https://192.168.1.3:8443/nexo/MX925-123456789`

The values in this URL are case-sensitive. 7b. If using the "Cloud" architecture: On the IP address configuration page, on the PIN pad FastTab, in the IP address field, enter the POIID value that you made a note of when you onboarded the Adyen payment terminal. Here is an example: `MX925-123456789`. The values in this field are case-sensitive.

7. If the payment terminal includes an onboard printer and you want to print receipts from the processor using that printer, enter 123 in the Port field that is separate from the IP address field in the PIN pad FastTab.

#### Update the Modern POS or IIS Hardware Station configuration

If you're packaging your own version of Modern POS by using the Retail SDK, you must follow these steps only one time in the SDK code before the installer is packaged. Otherwise, you must follow these steps after the standard Modern POS or IIS Hardware Station is installed.

1. Open the `dllhost.exe.config` file (for Modern POS) or the `web.config` file (for IIS Hardware Station).
2. Update the `PreloadedComposition` section as shown here, to switch from the legacy payment device adapter to the standard payment device adapter.

```
<PreloadedComposition>
  <composition>
    <add source="assembly"
value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.PaymentDeviceAdapter" />
    <!-- Switch from legacy to standard Payment Device Adapter.
    <add source="assembly"
value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.Legacy.PaymentDeviceAdapter" />
    -->
  </composition>
</PreloadedComposition>
```

3. Update the `appSettings` variable "PrintReceiptsOnCardDeclineOrVoid" value to `True` to print decline or void responses from the processor.

#### Call center

To configure the Dynamics 365 Payment Connector for Adyen for call center payments, follow the instructions in the [Set up a processor for new credit cards](#) section earlier in this topic.

#### e-Commerce

1. Sign in to Headquarters, and go to **Retail and Commerce > Channels > Online stores**.
2. Select the online store to add the Dynamics 365 Payment Connector for Adyen.
3. On the **Online store** page, on the **Payment accounts** FastTab, select **Add**.
4. In the **Connectors** field, select **Dynamics 365 Payment Connector for Adyen**.
5. Enter the following additional information.

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Assembly Name	Auto populated name of the assembly for the Dynamics 365 Payment Connector for Adyen.	Yes	Yes	<i>Binary name</i>

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Service account ID	Auto populated unique identifier for the setup of the merchant properties. This identifier is stamped on payment transactions and identifies the merchant properties that downstream processes (such as invoicing) should use.	Yes	Yes	<i>Guid</i>
Version	Enter the version of the Dynamics 365 Payment Connector for Adyen to use. "V002" should be used for all new implementations, as it leverages a newer Adyen API for card not present payments and is required for <a href="#">SCA support</a> .	Yes	Yes	"V001"/"V002"
Gateway environment	Enter the Adyen gateway environment to map to. The possible values are <b>Test</b> and <b>Live</b> .	Yes	Yes	Live
Optional Domain	Enter the domain to use when payment requests are made to Adyen. This is the unique identifier for your Live environment in the form <b>[random]-[company name]</b> . This is present as the prefix inside the API URLs under <b>Account &gt; API URLs</b> in your company's Live account on the Adyen Customer Area portal. For additional details, see, <a href="#">Live endpoints</a> .	Live only	No	Contact Adyen

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Merchant account ID	Enter the unique Adyen merchant identifier. This value is provided when you sign up with Adyen as described in the <a href="#">Sign up with Adyen</a> section.	Yes	No	MerchantIdentifier
Terminal architecture	This field is used only for the POS payment terminal integration and should be left blank.	No	Yes	<i>Leave this field blank.</i>
Local Password phrase	This field is used only for the POS payment terminal integration and should be left blank.	No	Yes	<i>Leave this field blank.</i>
Local Key Identifier	This field is used only for the POS payment terminal integration and should be left blank.	No	Yes	<i>Leave this field blank.</i>
Local Key Version	This field is used only for the POS payment terminal integration and should be left blank.	No	Yes	<i>Leave this field blank.</i>
Local Cryptor Version	Enter the Adyen cryptor version to use when you interact with the Adyen gateway. You should set this field to 1.	Yes	No	1
Cloud API Key	Enter the Adyen cloud API key. You can obtain this key by following the instructions on the <a href="#">How to get the API key</a> page on the Adyen website.	Yes	No	<i>The full cloud API key</i>
Supported Currencies	Enter the currencies that the connector should process.	Yes	Yes	USD;EUR

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Supported Tender Types	Enter the tender types that the connector should process.	Yes	Yes	Visa;MasterCard;Amex;Discover;Debit
Gift card provider	Enter the gift card provider that the connector should use to process gift cards. The possible values are <b>SVS</b> and <b>GIVEX</b> .	No	No	SVS
Terminal gift card entry	<i>POS Only</i> Allows the customer to select between <b>Manual</b> or <b>Swipe</b> .	Yes	Yes	True/False
Allow saving payment information in e-commerce	<i>e-Commerce only</i> Gives signed-in users the option to save payment details for future online purchases.	Yes	Yes	True/False
Authorization stale period (days)	<i>POS Only</i> Number of days before an authorization is considered stale and should decline before going to the processor for capture.	Yes	Yes	"7"
Origin Key	<i>e-Commerce Only</i> Only required when "V002" is designated for the version. You can obtain this key by following the instructions on the <a href="#">How to get an origin key</a> page on the Adyen website.			

6. On the Action Pane, select **Save**.

## Frequently asked questions

### Can I share a payment terminal with multiple hardware stations?

No. Payment terminals can only be used by a single hardware station or POS terminal. Attempting to connect multiple hardware stations to a single payment terminal will result in locking issues. If a payment terminal must be shared by multiple POS devices, an IIS hardware station must be deployed to manage the payment terminal.

### Can I reuse my existing payment terminal with the Adyen connector?

No. Adyen payment terminals are injected with the Adyen software. Therefore, existing payment terminals that aren't preconfigured with Adyen can't be reused with the Dynamics 365 Payment Connector for Adyen.

### Do I need a static IP address for the Adyen payment terminal?

Yes. Modern POS requires a known IP address to communicate with the Adyen payment terminal. Although the IP address of the Adyen payment terminal can be changed in the client, attempts to keep up with changing IP addresses involve significant overhead and could cause business disruption.

### Can I use my merchant bank?

Yes. Adyen can work with any merchant bank.

## Troubleshooting

### POS payment terminals

#### General issues

For all general issues, you should always consult the Modern POS or IIS Hardware Station event logs first. You can find these logs found under the following nodes in the Microsoft Windows event log:

- Application and Services Logs > Microsoft > Dynamics > Commerce-ModernPOS
- Application and Services Logs > Microsoft > Dynamics > Commerce-Hardware Station

#### Failing payment transactions

When payment transactions aren't successfully processed through the Adyen payment terminal, the corresponding error messages in the Dynamics 365 POS will contain a PSP reference number (PSP is the reference ID provided by Adyen used to uniquely identify each transaction). Provide this reference number when you contact Adyen support for help with specific transactions.

## Common issues

### POS payment terminals

#### The EFT terminal ID isn't set

<b>Title</b>	EFT Terminal ID is not set
<b>Symptom</b>	Payment authorization calls fail, and a hardware error occurs. An error message in the event log indicates that the <b>EFT Terminal ID</b> value isn't set.
<b>Root cause</b>	This issue can occur when the <b>EFT POS Register Number</b> field isn't set on the register or the IIS Hardware Station. It can also occur if the value is set but isn't correctly synced to the POS terminal. Finally, it can also occur when the value is cached.
<b>Fix</b>	Follow the instructions in the <a href="#">Set up a Dynamics 365 register</a> section earlier in this topic. Then run the <b>1070</b> and <b>1090</b> distribution schedules. If the issue isn't resolved, consider reactivating Modern POS, because the value of the <b>EFT POS Register Number</b> field might be cached and might need to be reset.

#### The Modern POS or IIS Hardware Station configuration isn't updated

<b>Title</b>	Config is not updated
--------------	-----------------------

<b>Symptom</b>	Modern POS error: "Sign in Error. The initialization data couldn't be loaded."
<b>Root cause</b>	This issue can occur when the POS is redeployed but the dllhost.config file hasn't been updated.
<b>Fix</b>	Follow the instructions in the <a href="#">Update the Modern POS or IIS Hardware Station configuration</a> section earlier in this topic. Then end the dllhost.exe task on the <b>Details</b> tab in Task Manager, and reopen Modern POS. If you're using an IIS Hardware Station, reset IIS.

#### Invoicing sales orders failed due to stale authorization

TITLE	CAPTURE FAILED DUE TO STALE AUTHORIZATION
<b>Symptom</b>	Invoicing sales orders fails with "Exception has been thrown by the target of an invocation. System.ArgumentNullException: Value cannot be null." The underlying error in the logs is "The following error occurred during the capture call - Dynamics 365 Payment Connector for Adyen: Error code Decline message Capture failed due to stale authorization."
<b>Root cause</b>	This error happens when an authorization older than the <b>Authorization stale period (days)</b> is sent to the payment connector for capture.
<b>Fix</b>	Ensure the value of <b>Number of days before expired in Accounts receivable parameters, Credit Card</b> is set to <b>1 less day</b> than the value set in merchant properties for all channels and then retry invoicing. The recommended value for <b>Authorization stale period (days)</b> is 14 in Adyen merchant properties and 13 in Accounts receivables parameters.

## Additional resources

- [Payments FAQ](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Saving online payment instruments with the Adyen connector

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes the setup and functionality that are related to saving payment instruments when you use the Adyen "card not present" payment connector for the Dynamics e-Commerce platform.

## Key terms

TERM	DESCRIPTION
Token	A string of data that a payment processor provides as a reference. Tokens can represent payment card numbers, payment authorizations, and previous payment captures. Tokens are important because they help keep sensitive data out of the point of sale (POS) system.
Card token	A token that a payment processor provides for storage in the POS system. The card token, or card reference, can be used only by the merchant who receives it
Authorization (Auth) token	After a POS system makes an authorization request to a payment processor, the payment processor provides a unique ID to the POS system as part of the response to that request. This authorization token, or authorization reference, can be used later, when the processor is called to perform actions such as reversing or voiding the authorization. However, an authorization token is most often used to capture funds when an order is fulfilled or when a transaction is being finalized.
List PI	A frequently used generic name for the capability that is described in this topic. List PI refers to the ability to save payment instruments and to list previously used payment instruments during future checkouts that are done through the same e-commerce website.
Named user	An e-commerce customer who is signed in to the online storefront at the time of checkout. Named users have a unique customer ID, and their online purchases are always mapped to the same customer ID whenever they are signed in to the online storefront.

## Overview

When e-commerce orders are created, retailers often offer to save the customer's payment card information so that it can be used for future transactions. This topic explains how that capability ("List PI") is delivered through the Microsoft Dynamics 365 Payment Connector for Adyen. Although the Adyen payment connector supports this capability out of the box, third-party payment connectors require customization. Additionally, not all payment processors might support the same method of saving payment card information.

The out-of-box implementation of the List PI capability relies on the payment processor to keep a mapping of an



online customer's unique ID to the payment instruments that have previously been processed through that payment connector. Only customers who are signed in to the website as named users have the option to save their payment card information for their next online visit. Customers who use a "guest checkout" option when they create an online order won't be able to save payment card information for future transactions.

## Prerequisites

The List PI capability requires the following elements:

- An e-commerce integration with Microsoft Dynamics 365 Commerce
- A payment connector that is compatible with the List PI capability
- A payment processor that maps unique customer IDs to the payment instruments that the customers want that payment processor to save

For more information about how to implement payment connectors and the software development kit (SDK) in general, visit the [Commerce for IT pros and developers home page](#).

## Setup

The List PI capability requires the following components and setup steps:

- **E-commerce integration** – An online storefront integration with Commerce is required. For more information about the e-Commerce SDK, see [e-Commerce platform software development kit \(SDK\)](#).
- **Online payments configuration** – The Dynamics 365 Payment Connector for Adyen supports List PI out of the box. For information about how to configure payments for online stores, see [Dynamics 365 Payment Connector for Adyen](#).

In addition to completing the ecommerce setup steps that are described in that topic, you must set the **Allow saving payment information in e-commerce** option in the Payment accounts fasttab of the **Online store** form to **Yes**.

- **Omni-channel payments configuration** – In the back office, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**. Then, on the **Omni-channel payments** tab, set the **Use omni-channel payments** option to **Yes**.

## Functional experience

### Guest checkout

When e-commerce visitors choose to check out as guests, customer records aren't created during checkout, and the customers can't save payment instruments for their next visit.

### Named user checkout

When named users (signed-in customers) go to the payment step of the checkout process, they will experience the List PI capability. The first time that a named user checks out, a **Save for my next payment** check box appears in the section where credit card information is entered.

A screenshot of a credit card payment form. At the top left, there is a radio button and a credit card icon, followed by the text "Credit Card". Below this, there are four input fields: "Cardholder name:" with the value "J. Smith", "Card Number:" with the value "1234 5678 9012 3456", "Expiry Date:" with the placeholder "MM/YY", and "CVC / CVV:" with the value "123". At the bottom, there is a checkbox labeled "Save for my next payment".

If this check box is selected, when a new credit card is submitted for payment, the named user's unique customer ID is sent to the payment processor, and the credit card is securely saved and mapped to the that unique customer ID.

If the same customer signs in during future visits to the storefront, they will be able to select the same credit card for payment at checkout.

A screenshot of a saved credit card payment method. At the top left, there is a radio button, the VISA logo, and the text "VISA (\*\*\*\* 1111)". Below this, there are two input fields: "Expiry Date:" with the value "10/2020" and "CVV:" with the value "123". At the bottom, there is a plus sign and the text "More payment methods".

### Order fulfillment and processing

E-Commerce orders where the customer applied a tender line by using the List PI capability work in the same way as orders that were created without using a saved card payment. From the standpoint of order processing and fulfillment, the two types of payment are indistinguishable.

## Details of eCommerce payment card tokenization

### Standard flow

In e-Commerce integrations, the payment card is typically entered as part of the checkout process and is saved together with the order before finalization. The card details are entered directly on a payment acceptance page that a payment processor provides. After card details are entered and the customer moves on to the next step of the checkout process, the processor creates a token that is used later in the order creation process.

When the customer finalizes the online order, the payment card token is sent to the payment processor as part of an authorization request. If the payment authorization request is successful, the payment processor replies by sending an authorization token. This authorization token is saved together with the customer's order and is referenced when that order is fulfilled from the back office.

### List PI flow

The main difference between the standard flow and the List PI flow is that the customer doesn't have to enter the full credit card number. Instead, the customer just has to select a previously saved credit card and provide the Card Verification Value (CVV number). If the customer provides the correct CVV number and moves on to

the next step of the checkout process, the payment processor provides a payment card token that will be included in the authorization request.

## Related articles

- [Payments FAQ](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Strong Customer Authentication (SCA) using the Adyen connector

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes Strong Customer Authentication (SCA) support built into the Adyen connector.

## Key terms

TERM	DESCRIPTION
Redirect	The action of moving an online shopper's browsing session out of the context of the merchant's storefront.
SCA	Strong Customer Authentication. Part of the EU Payment Services Directive 2.0 (PSD2.0) that requires online shoppers to be authenticated outside of their online shopping experience when paying with an electronic payment method.
Issuing bank	The financial institution that issues a payment instrument to a customer.

## Overview

PSD2.0 requires that SCA be supported during online shopping checkout so a customer can be authenticated by the bank that issued their payment method. The authentication commonly occurs when a shopper is going through the checkout for an online order and after they have provided their payment details. Those details are evaluated, and based on criteria provided by PSD2.0, the customer may be redirected to their bank. After being redirected to their bank, the customer is required to provide some form of authentication to confirm that they are an authorized user for the payment instrument. If the user is confirmed to be the cardholder, they are then redirected back to the storefront where the payment was previously submitted, after which checkout is allowed to proceed. If SCA fails, the process will not allowed to continue for the transaction.

## Prerequisites for SCA support

Support for SCA is provided by the out-of-box [Dynamics 365 Payment Connector for Adyen](#). This can be implemented by any third-party connector using the payments SDK.

## Setup

Setup details will vary by payment connector. For setup details related to the out-of-box Adyen connector, see the [e-Commerce section](#) of the Adyen connector topic.

## Functional experience

When a customer is redirected for SCA, they will be presented with a challenge by their bank, typically within a new browser window or iFrame. After they have been authenticated, they will be redirected back to the checkout session. If the validation fails, they will not be allowed to continue with checkout.

## Additional resources

- [Payments FAQ](#)
- [Dynamics 365 Payment Connector for Adyen](#)
- [Checkout module](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dedicated payment terminals and prompts for a printer and cash drawer

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic provides information about the capability to have a dedicated payment terminal and prompt the user to select a cash drawer and a receipt printer.

## Overview

Modern retailers are searching for ways to streamline the in-store checkout experience. Recent trends toward paperless checkout through electronic payments not only help to make the purchasing experience smoother but also reduce the need to have a full complement of peripheral devices available for every store associate.

Microsoft Dynamics 365 Commerce supports these trends by enabling a scenario where a point of sale (POS) device has a dedicated payment terminal assigned to it all the time, but it doesn't have its own receipt printer or cash drawer. When associates must print a receipt or take a cash payment, they are prompted to select a hardware station where those devices are configured.

## Key terms

TERM	DESCRIPTION
Register	The entity that is used to configure an instance of a POS register.
Device	A representation of the physical instance of a POS register and the Modern POS application that is assigned to it.
Dedicated hardware station	The hardware station business logic that is built into the Modern POS for Windows and Modern POS for Android applications.
Drawer kick (d/k) port	A traditional method for connecting a cash drawer to a receipt printer.
Network peripherals	Built-in support for network-enabled payment terminals, receipt printers, and cash drawers.

## Supported POS clients and devices

The functionality that is described in this topic is supported by the Modern POS for Windows and Modern POS for Android POS clients.

This functionality supports network-enabled payment terminals and receipt printers. You can provide cash drawer support by connecting the cash drawer to the network-enabled receipt printer via the d/k port.

Out-of-box support for this functionality is provided by the [Dynamics 365 Payment Connector for Adyen](#). However, other payment connectors might be supported via the Commerce software development kit (SDK) for payments. Supported receipt printers include network-enabled receipt printers from Star Micronics and Epson.

To set up Star Micronics receipt printers, use the Star Micronics Printer Utility to configure the device so that it

can be used over the network. This utility will also provide the IP address of the device.

To set up Epson receipt printers, use the Epson ePOS-Print utility to set up the device to use network protocols.

For more information about how to set up network peripherals, see [Network peripheral support overview](#).

## Set up a dedicated payment terminal and a prompt for a printer and cash drawer

### Set up hardware profiles

You must have two types of hardware profile. The first is assigned to the register. The second is assigned to a hardware station at the store level, and is used to logically group network receipt printers and cash drawers.

#### Set up a hardware profile for the register

To set up the hardware profile that is assigned to the register, follow these steps.

1. In Dynamics 365 Commerce, search for **Hardware profile**.
2. Select **New**.
3. Assign a hardware profile number, and then enter a description. This hardware profile will be assigned to the register itself. Therefore, a description such as **Dedicated with fallback** will suffice.
4. On the FastTabs for different device types, set up the following device types.

DEVICE	TYPE	DEVICE NAME	ADDITIONAL DETAILS
Printer	Fallback	<i>Any</i>	The device name is case-sensitive. The <b>Receipt profile ID</b> should be the same as the <b>Receipt profile ID</b> that is mapped to the network printer that is set up in the hardware profile that is assigned to the hardware station at the channel level.
Cash drawer	Fallback	<i>Any</i>	The device name is case-sensitive. Set the <b>Use shared shift</b> option to <b>Yes</b> .
EFT service	Adyen	Not applicable	For information about how to set up the out-of-box Adyen connector, see <a href="#">Dynamics 365 Payment Connector for Adyen</a> . Other payment connectors can be supported via the <a href="#">Commerce software development kit (SDK) for payments</a> .
PIN pad	Network	<b>MicrosoftAdyenDevice V001</b>	None.

5. In Dynamics 365 Commerce, search for **Registers**.

6. Select a register by selecting the register number, and then select **Edit**.
7. Assign the hardware profile that you just created to the register that should use a dedicated payment terminal. The device that is mapped to this register must use either the Modern POS for Windows application or the Modern POS for Android application.
8. Select **Save**.
9. On the Action Pane, on the **Registers** tab, select **Configure IP addresses**.
10. On the **PIN pad** FastTab, enter the IP address of the payment terminal. For information about how to get the IP address of the payment terminal by using the Adyen connector, see [Dynamics 365 Payment Connector for Adyen](#).
11. Select **Save**.

#### Set up a hardware profile for the receipt printer and cash drawer

To set up the hardware profile that is used to group the network receipt printer and cash drawer, follow these steps.

1. In Dynamics 365 Commerce, search for **Hardware profile**.
2. Select **New**.
3. Assign a hardware profile number, and then enter a description. This hardware profile will be used to group the receipt printer and cash drawer. Therefore, a description such as **Network printer and cash drawer** will suffice.
4. On the FastTabs for different device types, set up the following device types.

DEVICE	TYPE	DESCRIPTION	ADDITIONAL DETAILS
Printer	Network	<b>Epson or Star</b>	The device name is case-sensitive. The <b>Receipt profile ID</b> should be the same as the <b>Receipt profile ID</b> that is mapped to the printer that is set up in the hardware profile that is assigned to the register.
Cash drawer	Network	<b>Epson or Star</b>	The device name is case-sensitive. set the <b>Use shared shift</b> option to <b>Yes</b> .

5. Select **Save**.

#### Set up hardware stations

You must have two hardware stations. The first will be mapped to the register. The second will be selected as it's required, whenever a receipt must be printed or a cash drawer must be opened.

#### Register a hardware station

1. In Dynamics 365 Commerce, search for **All stores**.
2. Select a store by selecting its **Retail Channel Id** values, and then select **Edit**.
3. On the **Hardware stations** FastTab, select **Add**.
4. Set the **Hardware station type** field to **Dedicated**.
5. Enter a description, but don't set any other values for this hardware station. This hardware station will be



used for the register at all times.

#### **Set up a hardware station for the receipt printer and cash drawer**

1. In Dynamics 365 Commerce, search for **All stores**.
2. Select a store by selecting its **Retail Channel Id** values, and then select **Edit**.
3. On the **Hardware stations** FastTab, select **Add**.
4. Set the **Hardware station type** field to **Dedicated**.
5. Enter a description. This hardware station will be used for the receipt printer and cash drawer.
6. In the **Hardware profile** field, select the hardware profile that you previously created for the receipt printer and cash drawer.
7. Select **Save**.
8. While the hardware station for the receipt printer and cash drawer is still selected, select **Configure IP addresses**.
9. Obtain the IP address for the printer, and enter it as the IP address for both the receipt printer and the cash drawer.
10. Select **Save**.
11. Search for **Distribution schedules**.
12. Select distribution schedule **1090**, and then select **Run now**.
13. Select distribution schedule **1070**, and then select **Run now**.

#### **Set up the system to prompt for receipt printer and cash drawer selection as it's required**

1. In a supported POS client, close the current shift, if a shift is open.
2. Sign in, and then select **Non-drawer drawer operations**.
3. Use the **Manage hardware stations** operation to turn on a hardware station.
4. Select the hardware station that you created for the register to make it active.
5. Sign out of the POS, sign back in, and open a shift.

The payment terminal that is assigned to the hardware profile will now always be active, and you will be prompted if a receipt printer or cash drawer is required.

Many merchants who requested this feature are interested in reducing waste by providing email receipts and encouraging electronic payments. Depending on the configuration of the POS, store associates are prompted to select a receipt printer or cash drawer only when a customer wants a physical receipt or wants to pay with cash.

Store associates are prompted to select a hardware station only one time per transaction, unless a receipt must be printed and cash is used for payment, but the hardware profile that was originally selected doesn't include both devices. In that case, the store associate will be prompted again to select a hardware station that can be used to complete the transaction.

## Related articles

- [Set up POS hybrid app on Android and iOS](#)
- [Dynamics 365 Payment Connector for Adyen](#)
- [Network peripheral support overview](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Seamless offline switch for gift card and credit memo operations

2/18/2021 • 2 minutes to read • [Edit Online](#)

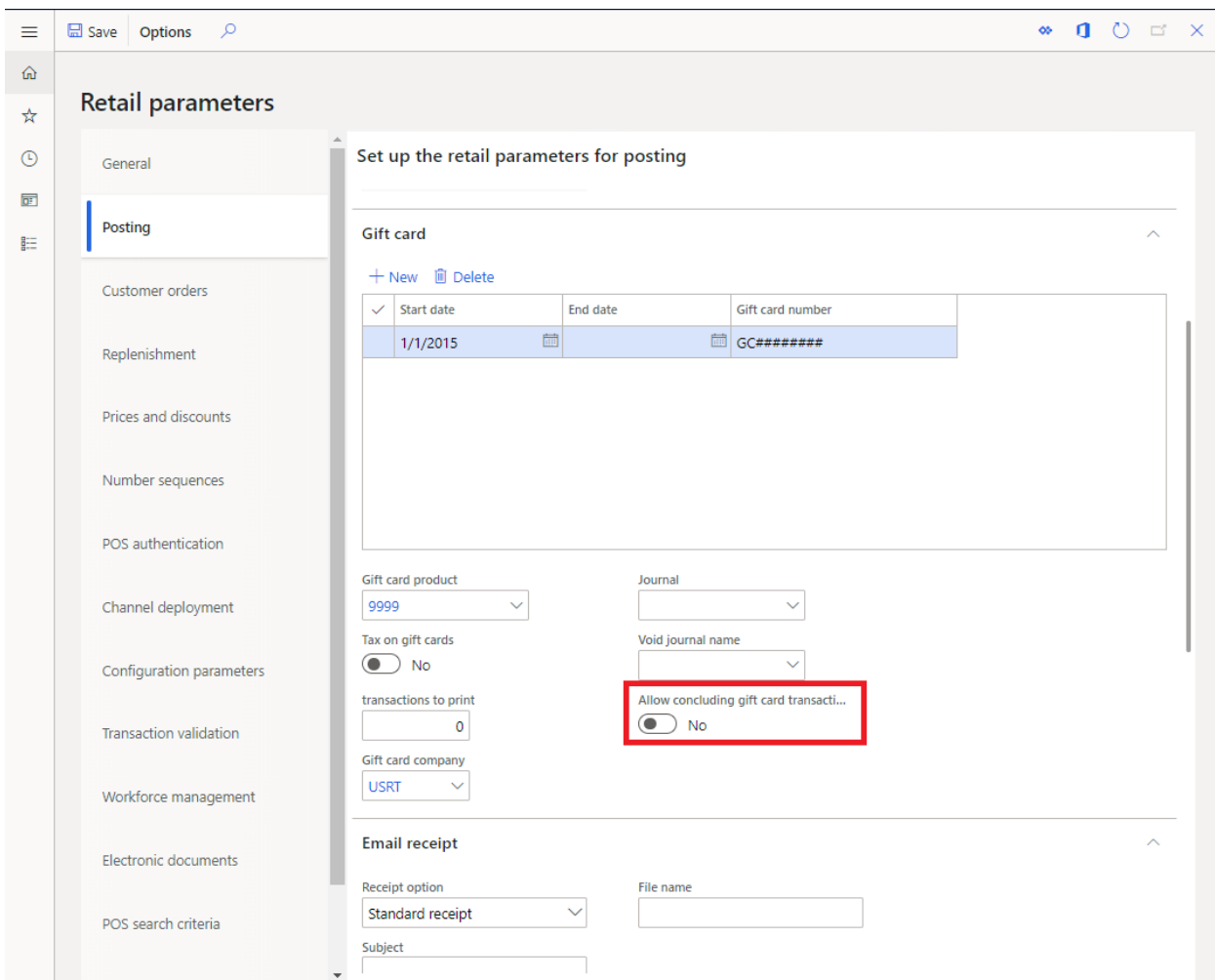
If a point of sale (POS) device loses its connection to the channel database, most POS operations and transactions that were in progress can proceed after the cashier receives a warning message about the loss of connectivity. However, in some cases, transactions have elements that rely on the real-time service, and those elements aren't supported when the POS is offline. This topic describes some functionality that helps reduce the impact of lost connectivity in these scenarios.

## Completing gift card transactions in offline mode

Internal gift cards depend on the real-time service, because the balance for the gift cards must be centrally maintained in Microsoft Dynamics 365 Commerce Headquarters. To help prevent fraud or other synchronization issues, gift cards are locked as soon as they are added to a transaction. The locking function ensures that a gift card can't be used on multiple terminals at the same time. When a transaction is completed, the gift card is updated and unlocked.

However, if the POS loses connectivity after a gift card has been added to a transaction, the gift card can become unusable. To help prevent this situation, Dynamics 365 Commerce has a parameter that enables transactions that include a gift card line to be completed while the POS is offline. When this parameter is turned on, gift card transactions that are forced offline will be saved together with offline transactions, and they will be synced to Commerce Headquarters when the offline transactions are synced. The synchronization will also unlock the gift card so that it can be used at another terminal.

To enable the functionality to conclude gift card transactions after switching to offline mode, go to the **Posting** tab on the **Commerce parameters** page. On that tab, locate the **Gift card** fasttab and set **Allow concluding gift card transactions in offline mode** to **Yes**.

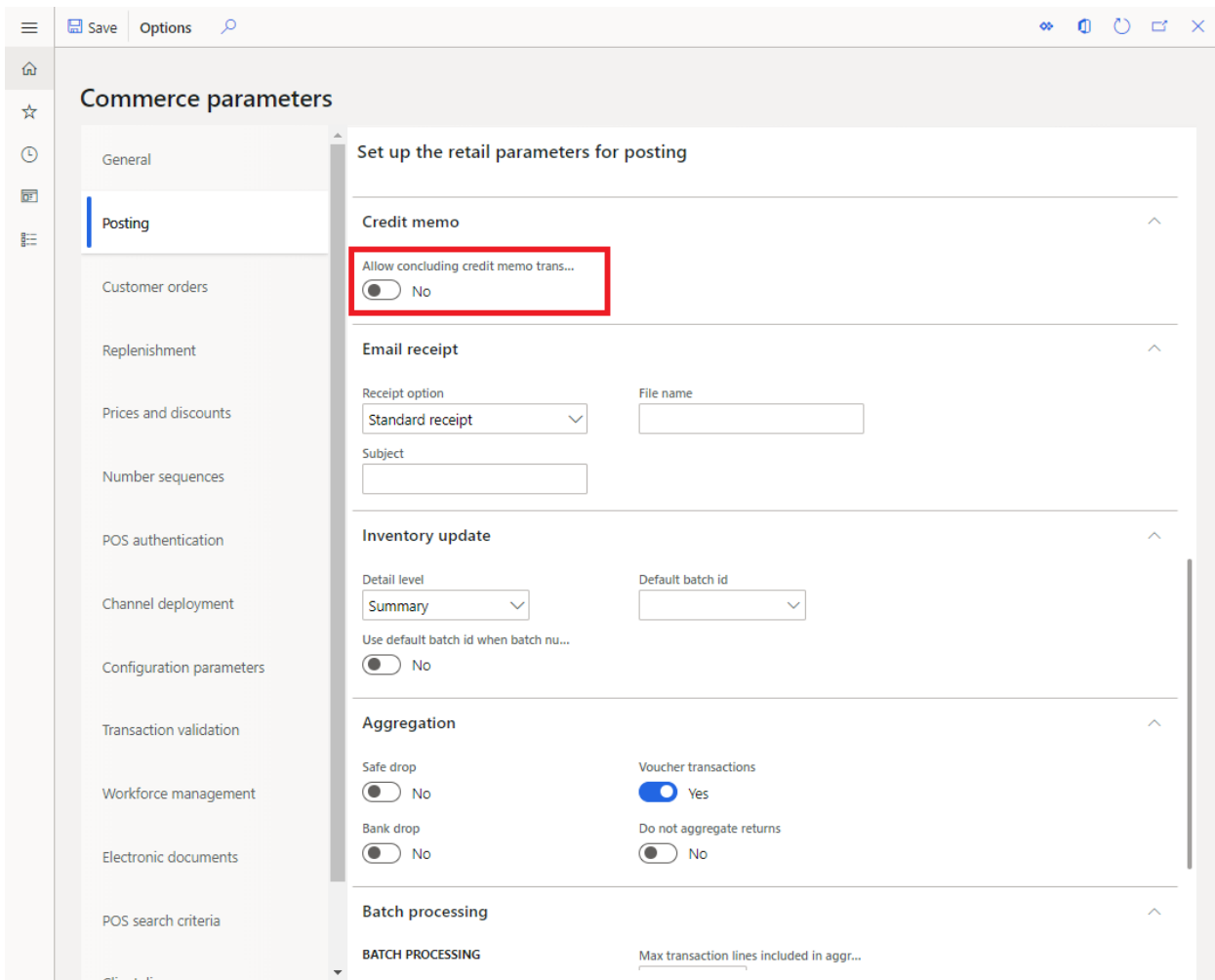


Commerce parameters are typically cached. Therefore, after the setting of this parameter is updated, and the distribution schedule is initiated to sync the change to the channel, the change can take up to 24 hours to take effect. To make the change effective immediately, reset Microsoft Internet Information Services (IIS).

## Completing credit memo transactions in offline mode

Like internal gift cards, credit memos are centrally maintained in Commerce Headquarters. Commerce has a parameter that enables credit memo transactions to be completed while the POS is offline. This parameter works like the gift card parameter that was mentioned in the previous section. When the parameter is turned on, credit memo transactions that are forced offline will be synced back to the channel database, together with other transactions that were performed while the POS was offline.

To enable the functionality to conclude credit memo transactions after switching to offline mode, go to the **Posting** tab on the **Commerce parameters** page. On that tab, locate the **Credit memo** fasttab and set **Allow concluding credit memo transactions in offline mode** to **Yes**.



Commerce parameters are typically cached. Therefore, after the setting of this parameter is updated, and the distribution schedule is initiated to sync the change to the channel, the change can take up to 24 hours to take effect. To make the change effective immediately, reset IIS.

## Related topics

- [Offline point of sale \(POS\) functionality](#)
- [Online and offline point of sale \(POS\) operations](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up pay invoice scenarios

2/18/2021 • 2 minutes to read • [Edit Online](#)

The Pay invoice functionality in Dynamics 365 Commerce has been expanded to support:

- Payoff of multiple sales order invoices in a single POS transaction.
- Payment of various customer invoice types including free text invoices, project-based invoices, and credit notes.

To enable these scenarios, the functionality profile for stores must be configured as outlined in below.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles** and select a profile that's linked to the stores that you want to make the changes for.
2. On the **Functions** tab, configure the following parameters as needed.
  - **Sales order invoice** – Select **Yes** to allow users to pay one or more sales order-based invoices in a single POS transaction.
  - **Free text invoice** – Select **Yes** to allow users to pay one or more free text-based invoices in a single POS transaction.
  - **Project invoice** – Select **Yes** to allow users to pay one or more project-based invoices in a single POS transaction.
  - **Sales order credit note** – Select **Yes** to allow users to settle multiple sales order-based credit notes against open invoices or process a refund to the customer for an open credit note.

## NOTE

Payment or settlement of partial amounts is not yet supported.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Duplicate payments prevention

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic provides an overview of the duplicate payments protection feature for Dynamics 365 Commerce Modern POS.

## Overview

This topic describes the user experience when the point of sale (POS) recovers from a loss of communication with the payment terminal, which causes the POS and the payment terminal to be out of sync.

The duplicate payment protection feature ensures that Modern POS can seamlessly recover from a loss of communication without requiring the shopper to process another payment through the payment terminal, which can lead to duplicate payments.

### NOTE

The duplicate payments protection feature is only supported for payments made using payment terminals.

This topic covers the following aspects of the duplicate payment protection feature:

- [Prerequisites](#) – Set of prerequisites to leverage this feature in Modern POS.
- [Scenario details](#) – Detailed description of the scenarios covered by the duplicate payment protection feature.
- [Troubleshooting](#) – Steps to take when encountering issues with the duplicate payment protection feature.
- [Additional resources](#) – List of related articles you might find useful when using the duplicate payment protection feature.

## Prerequisites

- The payment connector and corresponding payment gateway or processor must support this feature. The *payment connector* is an extension which facilitates communication between Commerce (and associated components) and a payment service. The connector described in this topic was implemented using the standard payments SDK.
- If a connector implements the corresponding duplicate payment protection interfaces, the feature is automatically enabled in Modern POS. Otherwise, it is automatically turned off.

## Scenario details

The duplicate payment protection feature is applicable to any scenario in which a payment is initiated and completed on a payment terminal, but Modern POS is unable to receive the corresponding response. As a result, the customer's card (such as a credit card) is charged but the payment line is not added to the POS. In most cases, the cashier will trigger a subsequent payment on the payment terminal, which results in a duplicate payment for the customer.

### How duplicate payments scenarios are triggered

#### 1. Cashier initiates payment

The cashier initiates a card payment by clicking **Pay card**, navigates to the **Payment** page, and clicks **Tender**.

## 2. Customer interacts with payment terminal

After the payment is initiated, the payment terminal prompts the customer for payment. The customer initiates the payment process on the payment terminal.

## 3. Modern POS loses connectivity to the payment terminal

- While the customer is running a payment on the payment terminal, Modern POS loses connectivity to the payment terminal because it either crashes, loses network connectivity, is closed, or the terminal is rebooted.
- The cashier will re-launch Modern POS and address any connectivity issues.

## 4. Customer completes payment on the payment terminal

As Modern POS is being reset, the customer completes the payment on the payment terminal and is charged.

## 5. Modern POS is launched

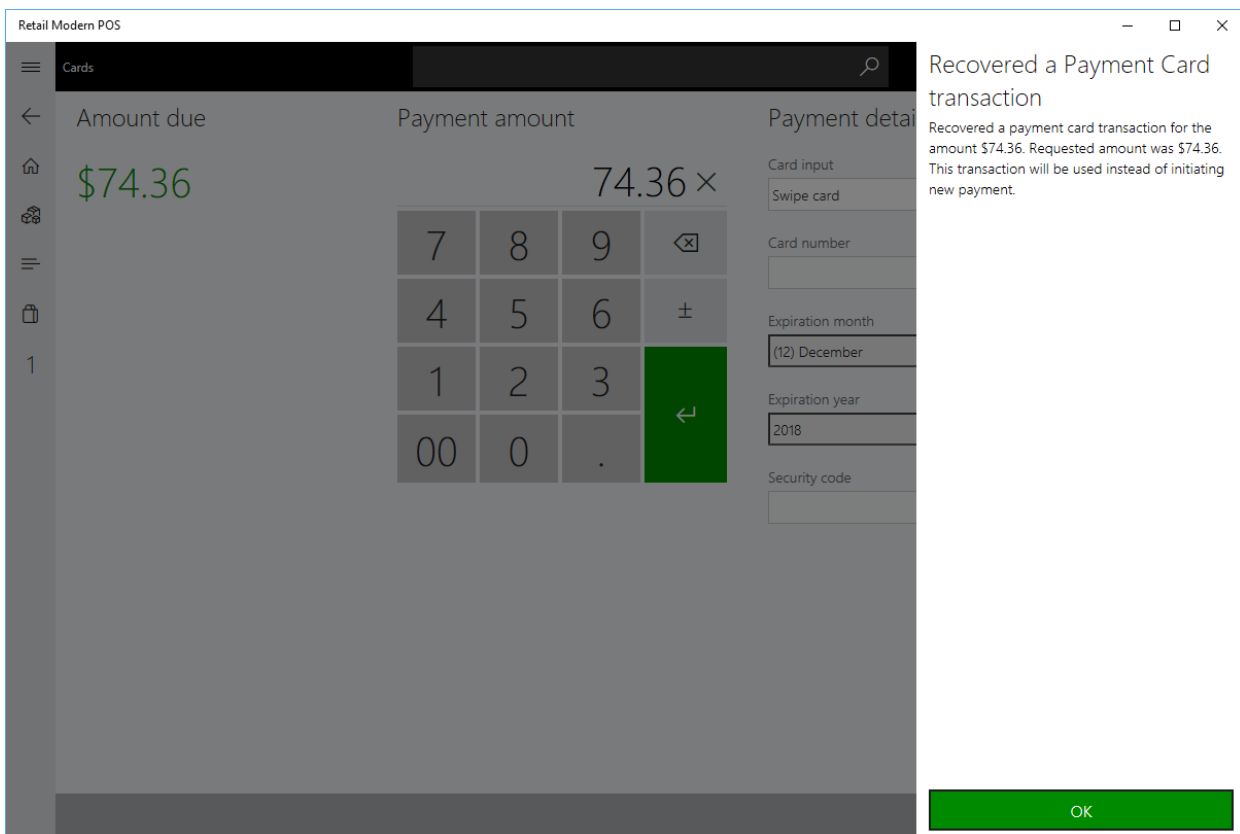
The cashier completes the reset/launch of Modern POS but the payment is not added to the cart.

### Payment recovery scenarios

Once the POS or network communications have been recovered, there are several scenarios that will result in the cashier being prompted to use the previous payment. Here are a few scenarios that can trigger payment recovery:

If there is an unrecovered payment and the cashier takes one of the following actions, the cashier is shown a dialog box indicating that a payment has already been made.

- Invokes another payment for any amount using a card payment.
- Invokes another payment for any amount using a cash payment.
- Attempts to void a line on the cart.
- Attempts to void the transaction.
- Attempts to suspend the transaction.



When the cashier clicks **OK**, the payment is recovered and added as a payment line to the cart.

The primary function of the duplicate payment protection feature is to put Modern POS back into the same state it would be if the original payment would have been successfully processed and the corresponding payment line was added to the cart.

### How to skip payment recovery

In some cases, the cashier might explicitly choose to skip the duplicate payment protection and opt not to recover a previous payment. In those cases, the cashier can use the following steps described to void the transaction without recovering the payment.

#### 1. Re-launch Modern POS

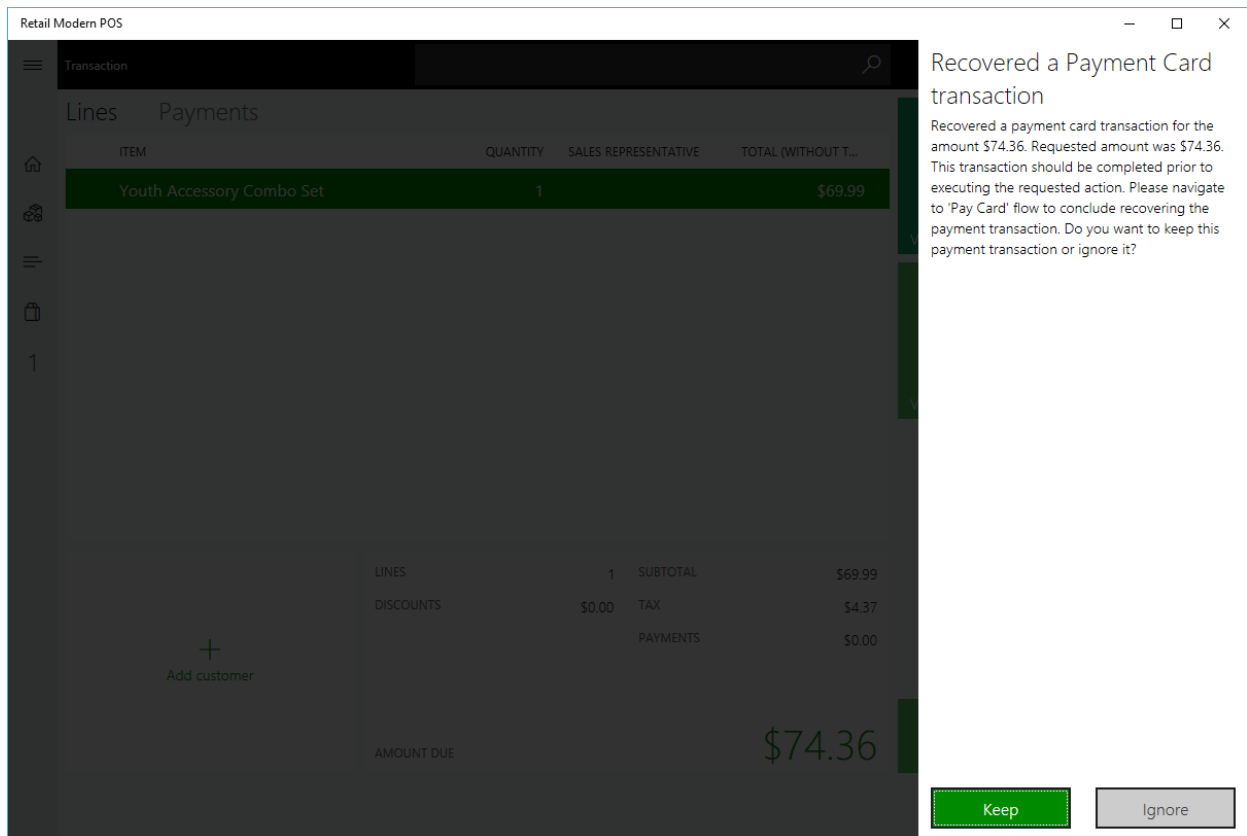
After Modern POS has lost connectivity to the payment terminal, re-launch the POS.

#### 2. Void the transaction

Navigate to the cart page and click **Void Transaction**.

#### 3. Ignore the recovered payment

A new dialog box will appear indicating that a recovered payment is available. Click **Ignore** to skip the payment recovery.



### What to do if the customer leaves the store

In some cases, the customer might leave the store before the cashier can finalize the transaction. In those cases, follow the steps described in the [How to skip payment recovery](#) section to void the transaction and manually void the payment on the portal of the payment gateway/processor.

## Troubleshooting

### General issues

For all general issues, you should always consult the Modern POS or IIS Hardware Station event logs first. The logs can be found under these nodes in the Windows event log:



- [Application and Services Logs > Microsoft > Dynamics > Commerce-ModernPOS](#)
- [Application and Services Logs > Microsoft > Dynamics > Commerce-Hardware Station](#)

### **Validate that the customer is not double charged**

Even if the duplicate payment protection feature is enabled, it is generally recommended that the merchant verifies that no double charge has occurred. To do this, check all transactions on the corresponding payment gateway/processor portal.

### **Payment recovery fails**

An error may occur while a previous payment is being recovered on Modern POS. This can happen if there is an issue in the payment connector or payment gateway/processor that does not allow the previous payment to be recovered. To resolve this issue, because the previous payment cannot be recovered, the cashier must skip the recovery as described in the [How to skip Payment Recovery](#) section.

## Additional resources

- [Payments FAQ](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Omni-channel payments overview

2/18/2021 • 16 minutes to read • [Edit Online](#)

This topic provides an overview of omni-channel payments in Dynamics 365 Commerce. It includes a comprehensive list of supported scenarios, information about functionality, setup, and troubleshooting, and descriptions of some typical issues.

## Key terms

TERM	DESCRIPTION
Token	A string of data that a payment processor provides as a reference. Tokens can represent payment card numbers, payment authorizations, and previous payment captures. Tokens are important because they help keep sensitive data out of the point of sale (POS) system. They are sometimes also referred to as <i>references</i> .
Card token	A token that a payment processor provides for storage in the POS system. A card token can be used only by the merchant who receives it. Card tokens are sometimes also referred to as <i>card references</i> .
Authorization (auth) token	A unique ID that a payment process provides as part of the response that it sends to a POS system after the POS system makes an authorization request. An authorization token can be used later if the processor is called to perform actions such as reversing or voiding the authorization. However, it's most often used to capture funds when an order is fulfilled or a transaction is finalized. Authorization tokens are sometimes also referred to as <i>authorization references</i> .
Capture token	A reference that a payment processor provides to a POS system when a payment is finalized or captured. The capture token can then be used to reference the payment capture in subsequent operations, such as refund requests.
Card not present	A term that refers to payment transactions where a physical card isn't presented. For example, these transactions can occur in e-commerce or call center scenarios. For these transactions, the payment-related information is manually entered on an e-commerce website, in a call center flow, or on the POS or payment terminal.
Card present	A term that refers to payment transactions where a physical card is presented and used on a payment terminal that is connected to the Microsoft Dynamics 365 POS system.

## Overview

In general, the term *omni-channel payments* describes the ability to create an order in one channel and fulfill it in another channel. The key to omni-channel payment support is preserving payment details together with the

rest of the order details, and then using those payment details when the order is recalled or processed in another channel. A classic example is the "Buy online, pick up in store" scenario. In this scenario, the payment details are added when the order is created online. They are then recalled at the POS to charge the customer's payment card at the time of pickup.

All the scenarios that are described in this topic can be implemented by using the standard Payments software development kit (SDK) that is provided with Commerce. The [Dynamics 365 Payment Connector for Adyen](#) provides an out-of-box implementation of every scenario that is described here.

## Prerequisites

Every scenario that is described in this topic requires a payment connector that supports omni-channel payments. The out-of-box Adyen connector can also be used, because it supports the scenarios that are made available through the Payments SDK. For more information about how to implement payment connectors, and about the Retail SDK in general, visit the [Retail for IT pros and developers home page](#).

## Supported versions

The omni-channel payment capabilities that are described in this topic were released as part of Microsoft Dynamics 365 for Retail version 8.1.3.

### "Card present" and "card not present" connectors

The Payments SDK relies on two sets of application programming interfaces (APIs) for payments. The first set of APIs is named `iPaymentProcessor`. It's used to implement "card not present" payment connectors that can be used in call centers and with the Microsoft Dynamics e-Commerce platform. For more information about the `iPaymentProcessor` interface, see the [Implement a payment connector and a payment device](#) white paper, which covers payments.

The second set of APIs is named `iNamedRequestHandler`. It supports the implementation of "card present" payment integrations that use a payment terminal. For more information about the `iNamedRequestHandler` interface, see [Create a payment integration for a payment terminal](#).

## Setup and configuration

The following components and setup steps are required:

- **eCommerce integration:** An integration with Commerce is required to support scenarios where an order originates in an online storefront. For more information about the Retail e-Commerce SDK, see [e-Commerce platform software development kit \(SDK\)](#). In a demo environment, the reference storefront supports omni-channel payment scenarios.
- **Online payments configuration:** The setup of the online channel must include a payment connector that has been updated to support omni-channel payments. Alternatively, the out-of-box payment connector can be used. For information about how to configure the Adyen payment connector for online stores, see [Adyen payment connector](#). In addition to the eCommerce setup steps that are described in that topic, the **Allow saving payment information in e-commerce** parameter must be set to **True** in the settings for the Adyen connector.
- **Omni-channel payments configuration:** In the back office, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**. Then, on the **Omni-channel payments** tab, set the **Use omni-channel payments** option to **Yes**. In Commerce versions 10.0.12 and later, this setting is in the **Feature Management** workspace. Select the **Omni-channel payments** feature and click **Enable now**.
- **Payment services:** The call center uses the default payment connector on the **Payment services** page to process payments. To support scenarios such as "Buy in call center, pick up in store," this default payment connector must be the Adyen payment connector or a payment connector that meets the implementation requirements for omni-channel payments.
- **EFT service:** Payments through a payment terminal must be set up on the **EFT service** FastTab of the hardware profile. The Adyen connector supports omni-channel payments scenarios out of the box. Other

payment connectors that support the `iNamedRequestHandler` interface can also be used if they support omni-channel payments.

- **Payment connector availability:** When an order is recalled, the payment tender lines that are recalled together with the order include the name of the payment connector that was used to create the authorizations that are associated with that order. When the order is fulfilled, the Payments SDK tries to use the same connector that was used to create the original authorization. Therefore, a payment connector that has the same merchant properties must be available for capture.
- **Card types:** For omni-channel scenarios to work properly, each channel must have the same setup for tender types that can be used for omni-channel. This setup includes payment method IDs and card type IDs. For example, if the `Cards` tender type has an ID of 2 in the online store setup, it should have the same ID in the retail store setup. The same requirement applies to card type IDs. If card number 12 is set to `VISA` in the online store, the same ID should be set up for the retail store.
- The Retail Modern POS for Windows or Android with built-in hardware station -or-
- Modern POS for iOS or Cloud POS with connected shared hardware station.

### Basic principle supporting omni-channel payments

Payment connectors and payment processors use tokens, or references, to reference interactions that are related to card payments. For example, when a payment authorization is requested, a reference to that authorization is provided. Therefore, the authorization can be referenced later, when funds are captured at the time of fulfillment. This authorization is unique to the merchant, payment connector, and processor.

If an order that was created online is being picked up in the store, the same payment details for that order must be recalled and used. When the original details are provided as part of the request to capture a payment against the original authorization, the payment processor will be able to handle the request and capture the payment.

To correctly reference the online order, a "card not present" payment connector that supports the same processor must also be available. In this way, the POS system can have one processor for "card present" payments, but it can also have access to other payment connectors so that it can fulfill orders that are created in other channels by using different payment processors.

## Supported scenarios

The following omni-channel payment scenarios are supported:

- Buy online, pick up in store
- Buy in call center, pick up in store
- Buy in store A, pick up in store B
- Buy in store A, ship to customer

#### NOTE

Payments made in the call center that map to the "Normal" payment function must be marked as **Prepay = Yes** to be reflected in the amount due when recalling the order in the POS. Non-prepay payments of type "Normal" are not recognized when the order is recalled in POS.

Variations of these scenarios are also supported. For example, an online order might include both lines that will be shipped to the customer and lines that will be picked up in a store. All order fulfillment options are supported via omni-channel payments.

The following sections describe the steps for each scenario and show how to run the scenario by using demo data.

## Buy online, pick up in store

Before you start, make sure that the following prerequisites are in place:

- You have a reference storefront where the Adyen connector is configured.
- The **Omni-channel payments** option on the **Commerce shared parameters** page is set to **True**. In later versions this setting is moved to the **Feature Management** workspace where you can select the **Omni-channel payments** feature and click **Enable now**.
- The Adyen payment connector is configured for the Houston POS register.
- The Retail Modern POS for Windows or Android with built-in hardware station -or-
- Modern POS for iOS or Cloud POS with connected shared hardware station.

Follow these steps to run the scenario.

1. In the reference storefront, create an order for in-store pickup. Be sure to select the **Houston** store.
2. Go through the checkout steps, and pay by using a test credit card number. You can find test credit card numbers on the [Adyen test card numbers page](#).
3. In Commerce, use the **Synchronize orders** batch job and the **P-001** distribution schedule to create the orders in the back office.
4. In the POS, on the welcome page, select the **Orders to pickup** operation to see the orders for in-store pickup.
5. Select one or more lines from the order that was created in the reference storefront, and then select **Pick up**.

The order is retrieved from the back office.

6. When the order line details are retrieved from the back office, and a card payment that can be used for omni-channel is detected, you're informed that a payment method is available.
7. Select **Use available payment method** to complete the transaction by using the card details that were entered in the reference storefront.

The order lines are loaded on the transaction page, and the balance due is 0 (zero).

8. Select the **Payments** tab to view the tender line that was pulled from the online order.
9. Select any payment method to complete the transaction.

## Buy in call center, pick up in store

1. In Commerce, on the **Customer service** page, enter **Karen Berg** in the search bar, and then select **Search**.
2. Select **Karen Berg** in the search results.
3. After Karen is loaded onto the **Customer service** page, select **New sales order**.
4. On the new sales order page, select **Header** to view the order header.
5. On the **Order header** page, set the site to **Central** and the warehouse to **Houston**.
6. On the **Deliver** tab, set the **Mode of delivery** field to **60** for customer pickup.
7. Select **Lines**, and then add one or more lines to the order.
8. Select **Complete** to enter the order completion flow.
9. Scroll down to the payments section, select **Add**, and then select a line where the payment method type is set to **Cards**.

10. Select the plus sign (+) to add a card payment.
11. Enter the details for a test credit card number that you found on the [Adyen test card numbers page](#), and then select **OK**.

**NOTE**

If the card brand for the card number that you entered differs from the brand that was selected when the payment was initiated, the payment will still go through. However, it will be posted to the accounts that are mapped to the card brand that you selected in step 10.

12. Select **OK** again to close the **Order completion payments** dialog box.
13. On the **Sales order summary** page, select **Submit**.
14. In the POS, on the welcome page, select the **Orders to pickup** operation to see the orders for in-store pickup.
15. Select one or more lines from the order that was created in the reference storefront, and then select **Pick up**.  
  
The order is retrieved from the back office.
16. When the order line details are retrieved from the back office, and a card payment that can be used for omni-channel is detected, you're informed that a payment method is available.
17. Select **Use available payment method** to complete the transaction by using the card details that were entered in the reference storefront.  
  
The order lines are loaded on the transaction page, and the balance due is 0 (zero).
18. Select the **Payments** tab to view the tender line that was pulled from the online order.
19. Select any payment method to complete the transaction.

**Buy in store A, pick up in store B**

1. Start the POS for the Houston store.
2. On the **Transaction** page, add Karen Berg to the transaction by using the number pad to enter **2001**.
3. Add one or more lines to the transaction.
4. Select **Orders** to see the order options.
5. Select **Pick up all**, and then, when you're prompted, select **Customer order**.
6. In the search bar, enter **Seattle**, and then select the **Seattle** store for pickup.
7. Select **OK** to accept the current date as the date of pickup.
8. Select **Pay card** to initiate the payment.
9. Tender the card payment for the amount that is due for the deposit.
10. Complete the deposit payment on the payment terminal.
11. After the deposit is paid, select the option to use the same card for fulfillment, and wait for the order to be completed.
12. Start the POS for the Seattle store.
13. In the POS, on the welcome page, select the **Orders to pickup** operation to see the orders for in-store

pickup.

14. Select one or more lines from the order that was created in the reference storefront, and then select **Pick up**.

The order is retrieved from the back office.

15. When the order line details are retrieved from the back office, and a card payment that can be used for omni-channel is detected, you're informed that a payment method is available.
16. Select **Use available payment method** to complete the transaction by using the card details that were entered in the reference storefront.

The order lines are loaded on the transaction page, and the balance due is 0 (zero).

17. Select the **Payments** tab to view the tender line that was pulled from the online order.
18. Select any payment method to complete the transaction.

### **Buy in store A, ship to customer**

1. Start the POS for the Houston store.
2. On the **Transaction** page, add Karen Berg to the transaction by using the number pad to enter **2001**.
3. Add one or more lines to the transaction.
4. Select **Orders** to see the order options.
5. Select **Ship all**, and then, when you're prompted, select **Customer order**.
6. In the shipping method page, select **Standard overnight**, and then select **OK** to accept today's date as the shipping date.
7. Select **OK** to accept the current date as the date of pickup.
8. Select **Pay card** to initiate the payment.
9. Tender the card payment for the amount that is due for the deposit.
10. Complete the deposit payment on the payment terminal.
11. After the deposit is paid, select the option to use the same card for fulfillment, and wait for the order to be completed.

When the order is picked, packed, and invoiced in the back office, the payment details that are provided at the POS will be used to capture the funds for the goods that are being shipped to the customer.

## Scenario details

In addition to the basic scenarios that were just described, several enhancements have been made to the Payments SDK to support omni-channel payments.

### **POS**

#### **Single swipe/dip for customer orders**

Before the omni-channel payments feature was implemented, when customer orders that included deposits were created at the POS, customers were required to swipe (or dip) their card two times: one time to pay the deposit and one time to tokenize the card for subsequent order fulfillment. When the omni-channel tokenization feature is turned on, customers must swipe their card only one time to both pay the deposit and authorize the amount that is due for goods that will be fulfilled later. At the time of fulfillment, the authorized funds are captured. Before the omni-channel tokenization feature was implemented, only a recurring card token was created for subsequent order fulfillment. Therefore, the funds for the pending fulfillment weren't authorized, and because those funds weren't being held for that specific purchase, it was less likely that they could be captured later.

#### NOTE

Single swipe isn't supported in Retail version 8.1.3. Customer orders in version 8.1.3 use the same flow that was used before the omni-channel tokenization feature was implemented.

### Cards that can't issue recurring card tokens

Some cards can't be used for omni-channel payments, because they don't support issuing recurring card tokens. When an order is created at the POS, if the deposit is paid by using a card that doesn't support recurring card tokens, the previous card tokenization flow is used. Therefore, a customer who wants to provide a payment that will be used for subsequent order fulfillment must present a second card. If the second card doesn't support recurring card tokens, the tokenization action will be declined, and the cashier will be prompted to ask the customer to provide a different card.

### Using a different card

A customer who comes to the store for order pickup has the option to use a different card. When the cashier receives the **Use available payment method** prompt at the time of order pickup, he or she can ask whether the customer wants to use the same card. If the customer has lost the card that was used to create the order and wants to pay for the order by using a different card, the cashier can select **Use a different payment method**. If the customer comes back later to pick up more items for the same order, if the original card authorization is still valid, the cashier can again ask whether the customer wants to use that card.

### Invalid authorizations

If the card that was used to create an order is no longer valid, when products are selected for pickup, the payment capture request will fail. The POS payment connector will then try to create a new authorization and capture by using the same card details. If the new authorization or capture fails, the cashier will be informed that the payment could not be processed. The cashier must then get a new payment from the customer.

### Multiple available payments

When an order that has multiple tenders and multiple lines is picked up, the cashier first receives the **Use available payment method** prompt. If there are multiple cards, when the cashier selects **Use available payment method**, existing card tender lines will be captured until the balance is met for the goods that are currently being picked up. The cashier won't have the option to select the card that should be used for the goods that are being picked up.

## Related topics

- [Payments FAQ](#)
- [Dynamics 365 Payment Connector for Adyen](#)
- [Configure BOPIS in a Dynamics 365 Commerce evaluation environment](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Omni-channel Commerce order payments

2/18/2021 • 12 minutes to read • [Edit Online](#)

This topic describes the omni-channel Commerce order payments feature in Microsoft Dynamics 365 Commerce. This feature lets you edit e-commerce and point of sale (POS) order payments from Commerce headquarters.

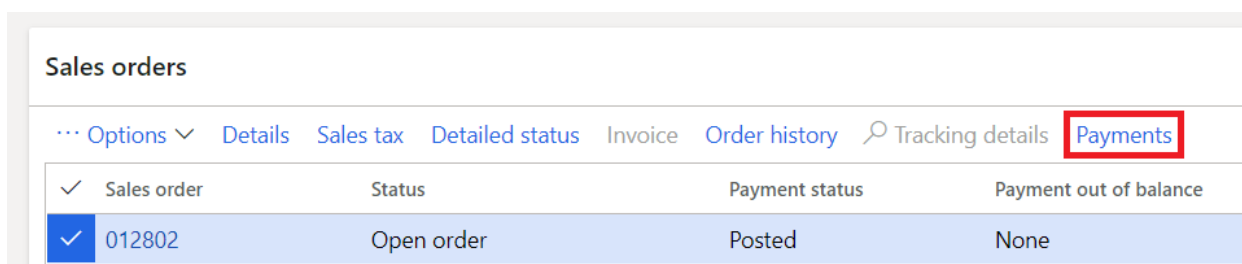
## Key terms

TERM	DESCRIPTION
Commerce payment	A payment that is associated with a customer order that was generated at the POS or in the e-commerce storefront.
Order completion	The business logic in the call center that ensures that payments have been collected before an order is submitted. The <b>Enable order completion</b> setting in the call center parameters is used to turn on this business logic. For more information, see <a href="#">Enable order completion</a> .
Call center order	An order that a call center user creates in Commerce headquarters.
Accounts receivable (AR) sales order	An order that a user who isn't a call center user creates through Accounts receivable in Commerce headquarters. Payments for AR sales orders can't be edited through call center order completion.

## Overview

Dynamics 365 Commerce consists of three main channels: POS, e-commerce, and call center. In Commerce version 10.0.12 and earlier, the management of payment lines for orders that are created in each channel isn't uniform. For example, when orders are created and edited in the call center, an order completion flow ensures that payments are specified for those orders before fulfillment. However, POS and e-commerce orders don't support call center order completion. To see the lack of uniformity, go to the **Customer service** page in Commerce headquarters, and notice which orders you can access the **Payments** page for by using the **Payments** button.

The following illustration shows an order that was created in the call center. Notice that the **Payments** button is available when the row for this order is selected.



The screenshot shows the 'Sales orders' interface. At the top, there are navigation options: 'Options', 'Details', 'Sales tax', 'Detailed status', 'Invoice', 'Order history', 'Tracking details', and 'Payments'. The 'Payments' button is highlighted with a red box. Below the navigation bar is a table with the following data:

✓ Sales order	Status	Payment status	Payment out of balance
✓ 012802	Open order	Posted	None

The following illustration shows an order that was created at the POS. Notice that the **Payments** button is unavailable when the row for this order is selected.

## Sales orders

Options ▾ Details Sales tax Detailed status Invoice Order history Tracking details **Payments**

✓ Sales order	Status	Payment status	Payment out of balance
012802	Open order	Posted	None
012801	Invoiced	Paid	None
✓ 012783	Invoiced	Not submitted	None

In Commerce version 10.0.13 and later, you can access the **Payments** page for orders that were created in e-commerce and the POS. Additionally, when the omni-channel Commerce order payments feature is turned on, the orders can be edited by using the order completion function that was previously available only for call center orders.

With this feature enabled, the **Sales order summary** dialog can be used to edit payments for orders originating in POS and e-commerce.

?

## Sales order summary

Order ▾ Hold ▾

012781 ^

<b>DETAILS</b>	Payment total	<b>INSTALLMENT BILLING</b>
Sales order	69.99	Eligible amount
012781	Balance	0.00
Sales status	0.00	Ineligible amount
Open order	Payment method	0.00
Payment status	1	<b>DATA</b>
Not submitted	Payment method name	Weight
Currency	Cash	0.00
<b>TOTALS</b>	Payment amount	Scripts
Sales total	69.99	
69.99		

---

**Amount** ▾

---

**Payments** ^

+ Add Remove Edit Installment

Payment method	Payment method name	Percent amount	Payment amount	Currency	Status
1	Cash	0.00	69.99	USD	Not submitted

Submit Cancel

# Prerequisites

To turn on the omni-channel Commerce order payments feature, you must first turn on several other features and complete other configurations. Aside from being requirements for enabling **Omni-channel Commerce order payments**, these features should be turned on as a best practice because they address functional gaps that are related to orders.

If any of the prerequisites are missing when you try to turn on the omni-channel Commerce order payments feature, you receive a message that states that you can't continue until the prerequisite features and configurations are in place.

### Omni-channel Commerce order paym... ✕

Feature added 6/14/2020  
Retail and commerce  
[Learn more](#)

You cannot enable this feature. This feature requires that the following features are enabled:

- Unified payment posting journal defaults for Commerce
- Omni-channel payments
- Duplicate payment protection on invoicing
- Enable refunds over multiple captures
- Enable manual void of expired credit card payment lines when authorizations are expired.

This feature also requires that channel payment methods have operation mappings. For more information, click [learn more](#).

**Description**

This feature adds support for managing MPOS and Storefront order payment lines in a way similar to Call Center payment lines. Specifically, when this feature is enabled, credit card payments for orders created in POS and e-commerce will be editable from the Call Center. In addition to editing capabilities, orders created when this feature is enabled will support linked refunds when the order is returned in the back office. For this feature to be fully supported, at least one Call Center channel must be set up and users editing orders in the back office must be set up as users in that call center channel.

**Comments**

[Schedule](#) [Enable now](#)

## Prerequisite features

The following features are required for omni-channel Commerce order payments to work correctly.

FEATURE NAME	DESCRIPTION
Unified payment posting journal defaults for Commerce	This feature changes the way that business logic creates customer payment and customer refund payment journals for orders that are created through the call center, POS, or e-commerce channel.
Omni-channel payments	This feature enables omni-channel payment scenarios, such as buy online, pick up in store. For more information, see <a href="#">Omni-channel payments overview</a> .

FEATURE NAME	DESCRIPTION
Duplicate payment protection on invoicing	This feature enables duplicate payment protection for invoicing scenarios. Commerce payments functionality might affect customizations in invoicing scenarios. If your organization has invoicing customizations, make sure that they are refactored before you turn on Commerce payments functionality in production environments.
Enable refunds over multiple captures	This functionality improves that capability to do multiple linked refunds against an order.
Enable manual void of expired credit card payment lines when authorizations are expired	This feature adds support for manual deletion of payment lines if they expire and the authorization cannot be refreshed.

### Configure prerequisites

#### Map payment methods to operations

You must map payment methods in all channels to corresponding operations, so that the management of order payments is supported in Commerce headquarters. Map payment methods before you turn on the omni-channel Commerce order payments feature, to avoid receiving warnings for each payment method that doesn't have an equivalent operation mapping.

The following illustration shows the mapping of a payment method to an operation in call center.

#### Configure a call center

To manage POS and e-commerce order payments through Commerce headquarters, you must configure at least one call center channel. For more information about how to create a call center channel, see [Set up a call center channel](#).

#### Set up users as call center users

Users who will edit Commerce payments in Commerce headquarters must be set up as users of the call center channel. For more information about how to set up call center users, see [Set up a call center channel user](#).

#### Turn on order completion for call centers

The order completion function must be turned on for call centers. Order completion enforces business logic that makes sure that orders can be paid during fulfillment. For more information about order completion, see [Enable order completion](#).

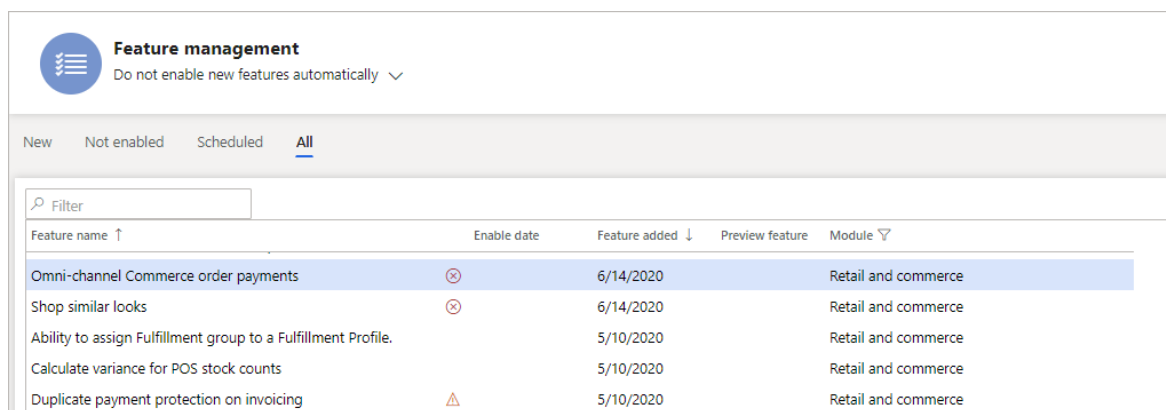
### Remove the Pay later option from the POS

When customer orders are created at the POS, the store associate can either collect a card payment for fulfillment or select **Pay later** to skip collection of card details. When the omni-channel Commerce order payments feature is turned on, the **Pay later** option should be removed from the POS. To remove it, search for **Functionality profiles** to open the **Functionality profiles** page. Select the relevant functionality provide and click **Edit**. On the **General** FastTab for the functionality profile, change the value of the **Require payment for fulfillment** field to **Card required**. This change must be synced to the channel database before it takes effect at the POS.

## Turn on the omni-channel Commerce order payments feature

After the prerequisites that are described in the previous section are in place, you can turn on the omni-channel Commerce order payments feature.

1. In the **Feature management** workspace, select the **All** tab to view the list of all features, and then search for **Omni-channel Commerce order payments**.



**Feature management**  
Do not enable new features automatically

New Not enabled Scheduled **All**

Filter

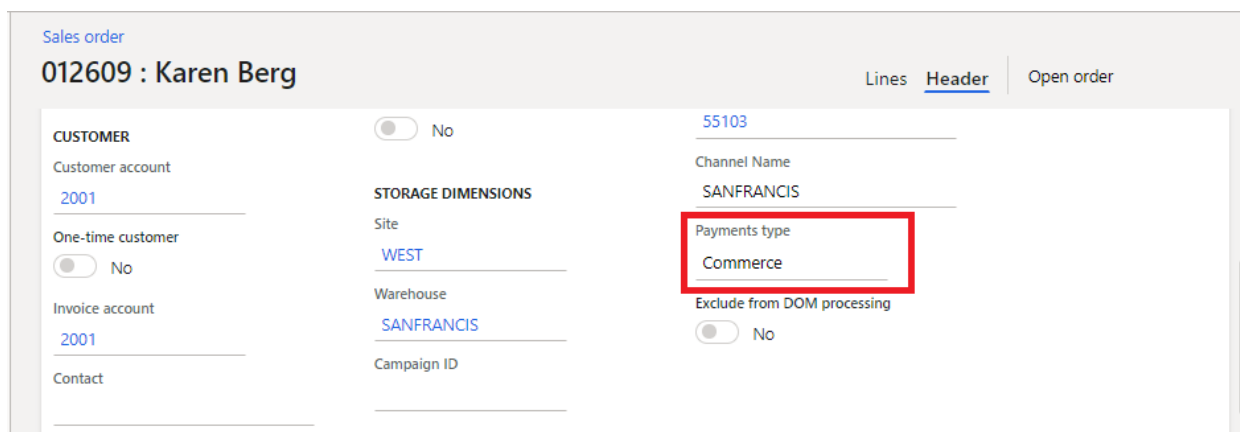
Feature name ↑	Enable date	Feature added ↓	Preview feature	Module ▾
Omni-channel Commerce order payments	⊗	6/14/2020		Retail and commerce
Shop similar looks	⊗	6/14/2020		Retail and commerce
Ability to assign Fulfillment group to a Fulfillment Profile.		5/10/2020		Retail and commerce
Calculate variance for POS stock counts		5/10/2020		Retail and commerce
Duplicate payment protection on invoicing	⚠	5/10/2020		Retail and commerce

2. Select the feature, and then select **Enable now**.

### IMPORTANT

The omni-channel Commerce order payments feature includes many changes to payments and order management workflows. You should do exhaustive testing before you turn on this feature in a production environment.

To distinguish channel orders that are created while the omni-channel Commerce order payments feature is turned on from other orders, the system shows a **Payments type** field on the order header when the feature is turned on. For POS and e-commerce orders, this field is set to **Commerce**.



Sales order  
**012609 : Karen Berg** Lines **Header** Open order

<b>CUSTOMER</b>	<input type="checkbox"/> No	55103
Customer account		Channel Name
2001		SANFRANCIS
<b>STORAGE DIMENSIONS</b>		<b>Payments type</b>
Site		Commerce
WEST		Exclude from DOM processing
Warehouse		<input type="checkbox"/> No
SANFRANCIS		
Campaign ID		

For call center orders, the **Payments type** field is set to **Call Center**. For sales orders that are created in Accounts receivable, the field isn't shown.

## Fulfill orders after the omni-channel Commerce order payments feature is turned off

POS and e-commerce orders that are created while the omni-channel Commerce order payments feature is turned on must be fulfilled while the feature is turned on. If the feature is later turned off, further processing of the orders will be prevented until the feature is turned back on.

## Manage orders that were created before the omni-channel Commerce order payments feature is turned on

Orders that were created before the omni-channel Commerce order payments feature is turned on can be processed after the feature is turned on. The editing experience for those orders won't change after the feature is turned on, and the orders won't be changed to accommodate omni-channel Commerce order payment workflows. Additionally, sales orders that non-call center users create in Accounts receivable will continue behave as they did before the feature was turned on.

### Key scenarios

When the omni-channel Commerce order payments feature is turned on, credit card payments for e-commerce and POS orders can be managed through order completion. For example, a customer places an online order and then calls into the call center to request a change to the order. In this case, the order completion function enables the payments on that order to be adjusted to support the new balance due.

The following properties on an order line can be edited before payment capture:

- Card type
- Card number
- Payment amount
- Percent amount

#### **Edit order payments**

The following scenarios in call center order completion apply to order payments that were created at the POS or in the e-commerce storefront.

#### **Uncaptured card payments**

For any card payment line on an order that hasn't yet been partially invoiced, the following properties can be edited before payment capture:

- Card type
- Card number
- Payment amount
- Percent amount

After the payments are edited, the order submission process corrects any changes that are required for edited payment lines.

SCENARIO	DESCRIPTION	SUPPORTED
----------	-------------	-----------

SCENARIO	DESCRIPTION	SUPPORTED
Edit to specify a higher amount.	For card payments that have been authorized but haven't yet been captured, the payment amount can be increased. When the amount on a payment line is increased, a new authorization is created for the new amount, and the old authorization is voided.	Yes
Edit to specify a lower amount.	For card payments that have been authorized but haven't yet been captured, the payment amount can be reduced. When the amount on a payment line is reduced, a new authorization is created for the new amount, and the old authorization is voided.	Yes
Remove an old card, and add a new card.	Uncaptured card payment authorizations can be removed from orders and replaced by a payment on a different card. The authorization for the first card is canceled, and an authorization for the new card will be obtained when the order is submitted.	Yes

#### Partially and fully captured card payments

SCENARIO	DESCRIPTION	SUPPORTED
Edit a payment that was used to invoice part of the order.	When an order that has omni-channel Commerce payments has been partially invoiced, the card payment amount for the existing card can be edited through call center order completion, down to the amount that has already been captured. A new card can then be applied to cover the balance due for the order.	Yes
Edit fully captured card payment lines to specify a higher amount.	If a card payment was fully captured, but the amount for that card payment is increased through call center order completion, a new authorization for the card is created for the increased amount when the order is submitted.	Yes

#### Remove order payments

SCENARIO	DESCRIPTION	SUPPORTED
Authorized payments	Omni-channel Commerce order card payments can be removed from an order through order completion, but only if they weren't partially captured.	Yes

SCENARIO	DESCRIPTION	SUPPORTED
Prepayments	Prepayments can't be removed through order completion. Prepayments can't be removed from an order after they are applied. Payment vouchers are already associated with them.	No
Partially captured payments	If the payment is in a <b>Paid</b> state but hasn't been fully captured, it can't be removed. However, the payment amount can be reduced to the amount that was already posted. When this happens, the a request is sent to the payment provider to reduce the authorization amount to equal the new payment amount.	No
Fully captured credit card payments and prepayments	Fully captured credit card payments and prepayments can't be removed from the order.	No

### Cancel order and sales lines

SCENARIO	DESCRIPTION	SUPPORTED
Order cancellation for credit card payments that aren't captured	If an order is canceled, card payment authorizations that haven't yet been captured are canceled.	Yes
Order cancellation for credit card payments that are captured but aren't invoiced	If an order is created at the POS, and a card payment is used to capture a deposit, the order is canceled before invoicing. The card payment is automatically refunded as part of order cancellation.	Yes
Order cancellation for orders that are partially shipped and invoiced	For orders that have been partially shipped and invoiced, cancellation will cancel the fulfillment of lines that haven't been invoiced. Open credit card authorizations for the remaining balance on the order aren't automatically canceled.	Manual refund is required.
Order cancellation for orders that are invoiced but aren't shipped	If an order is fully invoiced, but some of the items haven't been shipped, the order can be canceled. However, payments that are captured for that order won't automatically be refunded. Open authorizations for items that haven't been invoiced won't be canceled but will expire according to the authorization expiration policies of the bank that issued the card.	Manual refund is required.



SCENARIO	DESCRIPTION	SUPPORTED
Line cancellation for items that aren't fulfilled or invoiced	If an order line that hasn't been fulfilled or invoiced is canceled, the order completion process will require that payments are reduced to equal the new order total.	

## Refunds

SCENARIO	DESCRIPTION	SUPPORTED
Linked refunds for POS and e-commerce orders	Return orders that are generated from orders that originate from the POS and e-commerce channels can issue linked refunds against the cards that were charged during invoicing.	Yes
Linked refunds for AR sales orders	Although the payments can't be edited through order completion, returns that are issued for AR sales orders can be subject to a linked refund to the original card that was charged during invoicing.	
Unlinked refunds	If the merchant's return policies and the payment processor allow this approach, unlinked refunds can be specified for return orders in cases where the order was originally paid in cash, for example, or in cases where the original card that was used for payment is no longer active.	Yes
Refunds to non-card prepayments	Return orders that were originally paid through non-card prepayments, such as cash or credit memo payments, won't be subject to linked refund. An appropriate payment method, such as <b>Check</b> , must be specified for the refund payment. Organizations that allow unlinked refunds can refund non-card prepayments to credit cards that weren't previously used for the order, if the payment processor allows this approach.	Yes

## Edit and remove orders that have prepayments

SCENARIO	DESCRIPTION	SUPPORTED
Edit prepayment tender lines.	Payment vouchers are associated with prepayment tender lines. Therefore, prepayment tender lines can't be edited or removed.	No

## Related changes

To support omni-channel Commerce order payments, changes to existing functionality were introduced in

Commerce version 10.0.13.

## Consistent selection of payment journals when sales orders and refund payments are posted

In Commerce version 10.0.12 and earlier, payment journal assignment is inconsistent across channels. In Commerce version 10.0.13 and later, if the omni-channel Commerce order payments feature is turned on, all channels use the payment vouchers that are specified on the **Posting** tab of the **Commerce parameters** page.

The screenshot shows the 'Commerce parameters' configuration page, specifically the 'Posting' tab. The left sidebar lists various configuration categories, with 'Posting' selected. The main content area is titled 'Set up the retail parameters for posting' and includes several sections: 'Safe drop' (No), 'Bank drop' (No), 'Voucher transactions' (Yes), and 'Do not aggregate returns' (No). Below these are 'Batch processing' settings for parallel statements, threads, and transaction lines. The 'Invoice' section contains three sub-sections: 'PAYMENTS' with a 'Journal name' dropdown set to 'CustPay', 'TAXES' with a 'Tax calculation behavior' dropdown set to 'Recalculate', and 'REFUNDS' with a 'Journal name' dropdown set to 'ARREF'. The 'PAYMENTS' and 'REFUNDS' dropdowns are highlighted with red boxes. The 'Statement' section is partially visible at the bottom.

### Check payment method

Orders that are created at the POS don't include a check number when they're created in Commerce headquarters. When the omni-channel Commerce order payments feature is turned on, 9999 will be entered as the check number for orders that are created at the POS and paid for by check.

Additionally, the check number won't be required when **Check** is specified as the refund method of payment.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Process credit cards without a hardware station

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to configure the point of sale (POS) to process "card not present" transactions in POS clients that don't include a hardware station. This feature specifically targets emerging scenarios such as curbside pickup.

When this feature is turned on, clients such as Cloud POS and Modern POS for iOS can make credit card processing calls through Commerce Scale Unit. They don't have to depend on a standalone hardware station that is deployed on the local network. Therefore, any POS client can support curbside pickup, and fewer setup steps are required.

## NOTE

This feature should not be turned on for registers that support offline mode. The feature routes all "card not present" payment requests through the Commerce Scale Unit, but the Commerce Scale Unit isn't available when the register goes offline.

## Key terms

TERM	DESCRIPTION
BOPIS	This abbreviation is short for "buy online, pick up in store."
Curbside pickup	This scenario resembles BOPIS. However, instead of picking up items in the store, customers don't usually enter the store and often don't even leave their vehicle.
Card not present	This term is sometimes abbreviated CNP. It describes scenarios where the credit card or other form of electronic payment isn't physically present. In BOPIS and curbside pickup scenarios, customers make a payment online or over the phone, and the payment is then captured from the POS at the time of pickup.
Hardware station	This term describes the business logic that drives interactions between the POS and payment terminals or retail peripherals such as receipt printers. The hardware station is built into the Modern POS for Windows and Modern POS for Android clients. The Cloud POS and Modern POS for iOS clients require a standalone deployed hardware station to interact with physical devices.

## Overview

When this feature is turned off, Cloud POS and Modern POS for iOS can't process "card not present" credit card requests by themselves, because they don't have a built-in hardware station. When the feature is turned on, the Commerce Scale Unit can be used to facilitate the requests for those clients.

Although this feature can also be used for Modern POS for Windows and Modern POS for Android, in addition to Cloud POS and Modern POS for iOS, it isn't supported for offline mode. Therefore, the feature should not be

used in scenarios where a Windows client uses offline mode.

## Supported scenarios

The following scenarios are supported for POS clients that don't have a built-in hardware station.

SCENARIO	DESCRIPTION
Payment capture	An order can be recalled for pickup, and the credit card payment that is associated with the order can be captured.
Linked refund	A refund can be linked to the original payment instrument for return orders and cash-and-carry transactions.
Order editing	Orders can be recalled and edited in the POS, and the same payment card can be authorized to support the new order total.
Order cancellation	For orders that are canceled, the balance that is due back to the customer can be refunded to the original payment card.

## Unsupported scenarios

Creation of credit card authorizations isn't supported. Only existing card payments can be captured, refunded, or edited.

SCENARIO	DESCRIPTION
Creating a payment	This feature doesn't support the creation of new customer orders and authorization of payments for fulfillment. Creation of new payments will continue to require a hardware station.
Changing the payment card	If an order is recalled in the POS, the same payment method must be used for pickup. Store associates can substitute a different card for the card that is associated with an order only if a hardware station is available.
Offline mode	When this feature is turned on, "card not present" requests are always sent to the Commerce Scale Unit. If a register goes offline, "card not present" requests will fail, because the Commerce Scale Unit is no longer available. The feature should not be turned on for registers that are configured to support offline mode.

## Set up the POS to process "card not present" transactions without a hardware station

The configuration to turn on this feature is completed at the register level.

1. In the back office, go to **Retail and Commerce > Channel setup > POS setup > Registers**.
2. Select the relevant register, and then select **Edit**.
3. On the **General** FastTab, in the **Card not present processing** field, select **Use retail server**. (By default, this field is set to **Use hardware station**.)

POS registers

### SANFRAN-1: San Francisco - Register 1

**General**

Register number SANFRAN-1	<b>DESCRIPTION</b> Name San Francisco - Register 1 Placement	<b>STORE</b> Store number SANFRANCIS	<b>HARDWARESTATION</b> Select upon tendering No	EFT POS register number
Device SANFRANCIS-1		<b>PROFILES</b> Hardware profile HW002 Visual profile F4MP	<b>EFT</b> Store number SANFRANCIS	Default payment method
Support offline No				<b>Card not present processing</b> Use hardware station Use hardware station Use retail server

Fiscal information

Display

4. Select **Save**.

5. After the change is saved, run the 1090 distribution schedule to sync the changes to the POS.

## Additional resources

[Omni-channel payments overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Payment Connector for PayPal

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic provides an overview of the Microsoft Dynamics 365 Payment Connector for PayPal (PayPal connector). It includes a list of supported features and functionality, a guide to setting up and configuring the connector, troubleshooting information, and descriptions of some common issues.

## Key terms

TERM	DESCRIPTION
PayPal Wallet	Also known as the PayPal "button", PayPal Wallet describes the customer experience and integration supported by the PayPal Connector.
Wallet	A payment type that does not include traditional payment characteristics, such as the BIN range and expiration date, which are used to differentiate among credit and debit card types.

## Overview

Microsoft Dynamics 365 Commerce offers an out-of-box integration for PayPal Wallet. When the PayPal Connector is configured, the PayPal button is a selectable payment method as part of online order checkout. When users select **PayPal** they are directed to complete their payment directly with PayPal and then are returned to the online storefront for order completion.

The PayPal Connector is implemented using the same payments SDK that is leveraged for credit card payments. To better support PayPal payments, support for non-credit card payments has also been enhanced with the addition of support for "wallet" payment types. Specifically, PayPal payments do not return a BIN range. To support PayPal and other wallet payments, a new mapping for payments has been introduced that does not include BIN range. This new mapping can also be used to augment existing BIN range mapping for credit card payment methods. For more details, see [Wallet payment support](#).

The Microsoft Dynamics 365 Payment Connector for PayPal is not available in China. For other locales where Dynamics 365 Commerce is available, there are currently no restrictions.

## Functional overview

### PayPal Wallet in storefront

The connector supports the use of the PayPal Wallet, or PayPal button, for e-commerce payments. When the connector is configured for the online storefront, customers will be presented with the option to pay using the PayPal button at the time of payment. When the customer selects the PayPal button, they will be redirected to a PayPal Wallet mini-browser window where they will be authenticated by PayPal and can select their method of payment. Upon successful authentication and selection of a payment method, the customer will be redirected back to the storefront with the PayPal payment loaded into the checkout form. When the order is placed, the PayPal payment will be included as a payment line on the order and it will be synchronized to Commerce headquarters.

For more information on PayPal Wallet, visit the [PayPal Checkout page](#) hosted by PayPal.

## Fulfillment

Orders with PayPal payment lines are fulfilled in the same manner as orders that are paid using a credit card. When an order is created, the PayPal payment is added in an "authorized" state. Upon fulfillment, whether the order is shipped to the customer from a distribution center or picked up in a store, the payment authorization associated with the order is then "captured" using the same payments SDK requests used to capture credit card payments.

Fulfillment for PayPal orders supports incremental capture. This means that if an order is partially fulfilled and invoiced, a portion of the original authorization will be captured and, rather than getting a new authorization for the remainder, the same original authorization will be referenced when the remainder of the order is fulfilled.

## Authorization expiration

Orders made using the PayPal Payment Connector should be fulfilled within 30 days. If an order cannot be fulfilled or invoiced within 30 days, the original authorization will expire. The PayPal Connector does not currently support billing agreements. Recurring billing agreements, similar to recurring card references/tokens are required to automatically generate new authorizations after original authorization expiration. This means that if an authorization expires, the order will fall into a "do not process" state and the customer must be contacted to arrange for an alternate form of payment.

Billing agreement support, which allows for creation of new authorizations upon expiration of the original authorization, will be added in a future release.

# Testing the PayPal Payment Connector

## Creating a PayPal developer account

To test the PayPal Payment Connector, you must first create PayPal developer credentials and a PayPal sandbox environment.

1. Go to the [Test and go live](#) page provided as part of PayPal's development resources.
2. Select [Get Started](#) on the **Test and go live** page.
3. On that page, select **Log in to the Developer Dashboard**.
4. If prompted to sign in, select **Sign up**.
5. Select **Business Account**, then select **Next**.
6. Provide the email address you want to associate with your PayPal account and create a password for your PayPal account.
7. On the next page, enter contact information details, then read the PayPal user agreement and Privacy statement. If you agree to the terms, select **Agree and Create Account**.

### NOTE

The terms agreed to for the creation of a PayPal developer account are between the organization or individual creating the account and PayPal. Microsoft is in no way liable and makes no warranty as to the terms specified the agreement. These instructions are for informational purposes only.

8. After you agree to the terms, specify your business type and select **Continue**.
9. Next, go to the [PayPal Developer page](#) and select **Log in to Dashboard**.
10. Sign in using the credentials used when creating your PayPal account.
11. In the developer dashboard, select the **Default Application** in the list of RestAPI apps.

- Note the **Client ID** and **Secret** for your Sandbox account. These will be used to set up the connector in Dynamics 365 Commerce.

**NOTE**

To collect **Client ID** and **Secret** for a live environment, select the **Live** tab for the selected RestAPI app.

## Set up the connector in Dynamics 365 Commerce

### Map the PayPal wallet payment method to a processor payment method

**NOTE**

Some of these steps leverage a new capability for supporting wallet payment methods. For more information on this feature, see [Wallet payment support](#).

- Go to **Retail and Commerce > Channel Setup > Payment methods > Payment methods**.
- Select **New**.
- Specify a **Payment method ID** and **Payment method name**, such as **Wallet**. Set the **Default function** to **Wallet**, and then select **Save**.
- Go to **Retail and Commerce > Channel Setup > Payment methods > Card types**.
- Select **New**.
- Specify an **ID**, such as **PayPal**. Set an **Electronic payment name**, such as **PayPal**. Set **Type** to **Wallet**, and then specify a name for the **Issuer**, such as **PayPal**. Select **Save**.
- Select the entry previously created and select **Processor mapping**.
- On the **Processor payment method mapping** page, select the previously created **PayPal** card type. In the middle column, select the **Dynamics 365 Payment Connector for PayPal** and select **Add**.

### Set up the PayPal Payment Connector in payment services

Follow these steps to configure the PayPal payment connector in **Payment Services**.

- In Commerce headquarters, go to **Accounts receivable > Payments setup > Payment services**.
- On the Action Pane, select **New**, and then on the **Setup** tab, enter the following information.

FIELD	DESCRIPTION	SAMPLE VALUE
Payment service	Enter the name of the payment service to configure.	PayPal
Payment connector	Select the PayPal Payment Connector.	Dynamics 365 Payment Connector for PayPal
Test mode	For the PayPal Connector, in production and test environments you should set this field to <b>False</b> .	False
Default processor for credit cards	This should be set to <b>No</b> because the call center uses the default processor.	No



FIELD	DESCRIPTION	SAMPLE VALUE
Bypass payment processor for zero transactions	Specify whether this payment processor should be skipped for transactions that have a 0 (zero) amount.	Yes

3. On the **Payment service account** tab, enter the following information.

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Assembly Name	Auto-populated name of the assembly for the Dynamics 365 Payment Connector for PayPal.	Yes	Yes	<i>Binary name</i>
Service account ID	Auto-populated unique identifier for the setup of the merchant properties. This identifier is stamped on payment transactions and identifies the merchant properties that downstream processes should use (such as invoicing).	Yes	Yes	<i>Guid</i>
Merchant client ID	Enter the Sandbox <b>Client ID</b> collected from the PayPal developer dashboard under <b>Default application</b> .	Yes	Yes	<i>String</i>
Merchant API key	Enter the Sandbox <b>Secret</b> collected from the PayPal developer dashboard under <b>Default application</b> .	Yes	Yes	<i>String</i>
Supported currencies	Enter the supported currencies, semicolon separated, to be supported for the PayPal connector. The default is <b>USD</b> .	Yes	Yes, but can be edited.	USD; CAD

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Supported tender types	Other payment connectors may support multiple tender types. For PayPal, the only payment method will be <b>PayPal</b> .	Yes	Yes	PayPal
Supported payment method variants	Other payment connectors may return multiple payment method variants. For PayPal, the only variant will be <b>PayPal</b> .	Yes	Yes	PayPal
Environment	This field is used to specify whether transactions should be sent to Sandbox or Live environments.	Yes	Yes	<i>Sandbox or Live</i>

#### NOTE

When testing payments in a Sandbox environment, the **Environment** field should never be set to live and live environment. **Merchant client ID** and **Merchant API keys** must never be used. Sandbox environments are for Sandbox testing only.

### Set up the PayPal Payment Connector for the online store

1. In Commerce headquarters, go to **Retail and Commerce > Channels > Online stores**.
2. Select the online store to add the Dynamics 365 Payment Connector for PayPal.
3. On the **Online store** page, on the **Payment accounts** FastTab, select **Add**.
4. In **Connectors**, select **Dynamics 365 Payment Connector for PayPal**.
5. Enter the following additional information.

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Assembly Name	Auto-populated name of the assembly for the Dynamics 365 Payment Connector for PayPal.	Yes	Yes	<i>Binary name</i>

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Service account ID	Auto-populated unique identifier for the setup of the merchant properties. This identifier is stamped on payment transactions and identifies the merchant properties that downstream processes should use (such as invoicing).	Yes	Yes	<i>Guid</i>
Merchant client ID	Enter the Sandbox <b>Client ID</b> collected from the PayPal developer dashboard under <b>Default application</b> .	Yes	Yes	<i>String</i>
Merchant API key	Enter the Sandbox <b>Secret</b> collected from the PayPal developer dashboard under <b>Default application</b> .	Yes	Yes	<i>String</i>
Supported currencies	Enter the supported currencies, semicolon separated, to be supported for the PayPal connector. The default is <b>USD</b> .	Yes	Yes, but can be edited.	USD; CAD
Supported tender types	Other payment connectors may support multiple tender types. For PayPal, the only payment method will be <b>PayPal</b> .	Yes	Yes	PayPal
Supported payment method variants	Other payment connectors may return multiple payment method variants. For PayPal, the only variant will be <b>PayPal</b> .	Yes	Yes	PayPal

FIELD	DESCRIPTION	REQUIRED	AUTOMATICALLY SET	SAMPLE VALUE
Environment	This field is used to specify whether transactions should be sent to Sandbox or Live environments.	Yes	Yes	<i>Sandbox or Live</i>

#### NOTE

When testing payments in a Sandbox environment, the **Environment** field should never be set to live and live environment. **Merchant client ID** and **Merchant API keys** must never be used. Sandbox environments are for Sandbox testing only.

After the above changes have been made in Commerce headquarters, synchronize the changes using the **1070** distribution schedule.

#### Configure PayPal for the Storefront checkout module

For details related to configuring storefront to use PayPal in the checkout module, see [Payment module](#).

## Entering a merchant relationship with PayPal

To create a **Live** merchant account with PayPal, visit the [PayPal Checkout page](#) for self-service or [connect with a sales representative](#) to discuss custom rates.

## Additional resources

- [Payments FAQ](#)
- [Checkout module](#)
- [Payment module](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Wallet payment support

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic provides an overview of wallet payment support for Microsoft Dynamics 365 Commerce.

## Key terms

TERM	DESCRIPTION
Wallet	A payment type that does not include traditional payment characteristics, such as the BIN range and expiration date, which are used to differentiate among credit and debit card types.
Processor payment method	A property payment card property in the payments SDK. When this property is added to supported payment methods within a connector, those payment methods can be mapped to cards or wallets configured in Commerce headquarters to avoid the traditional BIN range mapping.

## Overview

Unlike traditional credit and debit cards, wallet payment authorization responses do not include BIN ranges. Traditionally, BIN ranges have been used to match payment authorization responses to predefined card types with BIN ranges. This feature adds support for **Processor payment methods** and mapping of those variants to card types set up in Commerce headquarters.

**Processor payment method** is a property for the payments SDK that can be applied to payment methods supported for a particular payment connector. When an authorization response is received that includes the processor payment method, a lookup is performed to determine if that processor payment method has been mapped to a card or wallet type. If a mapping is found, that payment is mapped to the matching card or wallet type. If a match can't be found, a BIN lookup is performed following the traditional BIN range settings for Commerce.

Because wallet payments don't include BIN range, if a payment connector such as PayPal supports wallet payments, the payment connector should be updated to the latest payments SDK and the processor payment method property should be populated at least for all supported wallet payments.

Processor payment method mappings are also useful for traditional debit and credit card payments. Mapping processor payment methods to cards is more straightforward than BIN range mapping and less prone to errors because it is easy to ensure all possible payment methods supported by a connector are mapped to a card or wallet type.

## Wallet payment method support and processor payment methods

This feature supports a new payment method and card type called **Wallet**. The primary characteristic of a wallet payment is that it does not have a BIN range. However, wallet payment methods may not return expiration dates and some properties that have traditionally been considered mandatory. Wallet payment methods must be mapped to processor payment methods as an alternative to the traditional BIN range mapping.

### Adding support of processor payment methods

To support processor payment methods, payment connectors need to populate the **PaymentMethodVariant** property in **PaymentCardProperties**. If the payments SDK in use does not include this property, it should be updated.

### Processor payment method mapping

The **Processor payment method mapping** page can be used to map processor payment methods to configured card or wallet types. To access this page, select the **Processor mapping** link on the **Card types** page.

ID ↑	Electronic payment name	Type	Issuer
AMEXPRESS	American Express	International credit card	Credit provider
DISCOVER	Discover card	International credit card	Credit provider
EUROCARD	EuroCard	International credit card	Credit provider
GIFTCARD	Gift Card	Gift card	Gift card provider
LOYALTY	Loyalty Card	Loyalty card	US Loyalty
MAESTRO	Maestro	International debit card	Debit provider
MASTER	Mastercard	International credit card	Credit provider
PAYPAL	PayPal	Wallet	PayPal
VISA	Visa Card	International credit card	Credit provider
VISAELEC	Visa Electron	International debit card	Debit provider

When this page opens, it queries available payment connectors to collect a set of payment methods with the **PaymentMethodVariant** field populated. It then checks to determine if those payment methods have an existing mapping to a card or wallet. Payment methods that do not have a mapping are listed in the center column of the page.

Processor payment method mappings

CARD TENDER TYPES

Card ID ↑	Card type name	Card types	Card issuer
AMEXPRESS	American Express	International credit card	Credit provider
DISCOVER	Discover card	International credit card	Credit provider
EUROCARD	EuroCard	International credit card	Credit provider
GIFTCARD	Gift Card	Gift card	gift card provider
LOYALTY	Loyalty Card	Loyalty card	US Loyalty
MAESTRO	Maestro	International debit card	Debit provider
MASTER	Mastercard	International credit card	Credit provider
PAYPAL	PayPal	Wallet	PayPal
VISA	Visa Card	International credit card	Credit provider
VISAELEC	Visa Electron	International debit card	Debit provider

UNMAPPED PROCESSOR PAYMENT METHODS

Connector	Processor payment method mappings
✓ Dynamics 365 Payment Connect...	PayPal

MAPPED PROCESSOR PAYMENT METHODS

Card ID	Connector	Processor payment method mappings
✓	Connector 1	

We didn't find anything to show here.

OK Cancel

To map a processor payment method to a card or wallet, select the card or wallet, select the processor payment method, and then select **Add**. The processor payment method moves to the **Mapped** column. When a matching payment authorization is received, it will be mapped to the chosen card or wallet.

## Processor payment method mappings

Card ID ↑	Card type name	Card types	Card issuer
AMEXPRESS	American Express	International credit card	Credit provider
DISCOVER	Discover card	International credit card	Credit provider
EUROCARD	EuroCard	International credit card	Credit provider
GIFTCARD	Gift Card	Gift card	Gift card provider
LOYALTY	Loyalty Card	Loyalty card	US Loyalty
MAESTRO	Maestro	International debit card	Debit provider
MASTER	Mastercard	International credit card	Credit provider
PAYPAL	PayPal	Wallet	PayPal
VISA	Visa Card	International credit card	Credit provider
VISALEEC	Visa Electron	International debit card	Debit provider

Card ID	Connector ↑	Processor payment method mappings
PAYPAL	Dynamics 365 Payment Connect...	PayPal

### When not to use processor payment method mapping

In certain cases, processor payment method mapping may not be granular enough for reporting needs. For example, some retailers differentiate external gift cards from the same provider by their BIN range. In this scenario, the gift cards should not be mapped using the above scenario. Instead, they should continue to use traditional BIN range mapping.

### Support for unidentified card types

In some scenarios, a payment connector may return a card that does not have a BIN range or processor payment method mapping. If this occurs, the payment is authorized by the payment terminal, but is then reversed when the point of sale (POS) can't map the authorization response to a specific card type. To address this, a capability is provided to map unknown authorization responses to a default card type.

Stores  
HOUSTON: Houston

General 052 | HOUSTON | TX | 100001

IDENTIFICATION	POS REGISTER	SALES TAX	ADDRESS BOOK
Retail Channel Id 000017	Functionality profile FN001	Tax identification number (TIN)	Customer address book RetailCust
Name Houston	Inventory lookup <input type="radio"/> No	Sales tax group TX	Employee address book Houston;USRTCenfr;U...
Store number HOUSTON	PROFILES	Prices include sales tax <input type="radio"/> No	CUSTOMER
Operating unit number 052	Channel profile scu5nphoam097518925	Use destination-based tax <input type="radio"/> No	Default customer 100001
Legal entity usrt	Live channel database scu5nphoam097518925	Use customer-based taxes <input type="radio"/> No	EMAIL NOTIFICATION
Warehouse HOUSTON	Offline profile AX7	Calculate customer tax exempt <input type="radio"/> No	Email notification profile
Shipping warehouse HOUSTON	REGIONAL SETTINGS	Sales tax override group Default	ORDER FULFILLMENT
Store time zone (GMT-06:00) Central Time (US ...	Language en-us		Manual accept <input type="radio"/> No
	Currency USD		ELECTRONIC PAYMENTS
			Default for unmapped processor pay...

This capability ensures that the payment is never authorized by the terminal and then reversed by the POS. This helps avoid confusion for customers and store associates. When this setting is used, the default card for unknown authorizations should be checked periodically to ensure that wanted card types are not accidentally being mapped to the default for unknown card types. If a card type is truly unwanted for processing, it should be turned off at the processor level.

## Additional resources

- [Payments FAQ](#)
- [Dynamics 365 Payment Connector for PayPal](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Payments FAQ

2/18/2021 • 2 minutes to read • [Edit Online](#)

## What payment scenarios are supported?

- Set up a merchant account.
- Process a call center order.
- Process an online order.
- Process a POS cash-and-carry transaction by using an accepting page.
- Process a POS cash-and-carry return by using an accepting page.
- Process a POS cash-and-carry return by using an accepting page.
- Process a POS customer order by using an accepting page.
- Process a POS cash-and-carry transaction by using Microsoft Dynamics Commerce Retail Hardware Station.
- Process a POS cash-and-carry return by using Hardware Station.
- Process a POS customer order by using Hardware Station.
- Buy online, pick up in store.
- Buy in call center, pick up in store.

## Which payment providers are supported and in what regions?

- Adyen is supported for card present and card not present transactions. For a list of supported regions, visit the [Dynamics 365 Payment Connector for Adyen overview page](#).
- PayPal is supported for online purchases. For a list of supported regions, visit the [Dynamics 365 Payment Connector for PayPal overview page](#).
- The **TestConnector** isn't supported for user acceptance testing (UAT) or for production (prod). The **TestConnector** is for sandbox and development use only, and is for sample use only.
- Mastercard Simplify is no longer supported for new customers.

## What is a payment connector and in what cases do I need to deploy and implement a payment connector?

Payment connectors are software components that can be set up which enable an application to process payments for transactions where the card is not present and transactions where the card is present.

Microsoft-provided connectors such as Adyen can be used, or custom connectors can be built by ISV partners. A connector is typically built to meet the business needs of a customer. Custom connectors are often created when there is a scenario that requires a new type of payment type (for example, linked refunds). Customers doing business in certain geographies may need new connectors if the out-of-box connectors do not support those regions.

## Are other payment connector providers supported?

Yes, but you must connect them using customization.

## What is the Service level agreement (SLA) for out-of-box payment connectors like Adyen?

For the Adyen connector, refer to the Adyen connector [overview page](#) if the issue is related to set up. For other setup or functional issues with the connector itself, create a support request with Microsoft. If the issue is originating from the device itself or Adyen's processing service, contact Adyen support at support-dynamics365@adyen.com.

## If a supported payment provider issues an update, will Microsoft automatically update the payment connector or do I need to work with the payment provider to get the updated payment connector?

If a payment connector update is issued by the payment connector provider, the updated version of the payment connector will be included in the next planned release of Dynamics 365 Commerce. However, the customer can also work directly with the payment connector provider to uptake it earlier.

### Related topics

- [Create an end-to-end payment integration for a payment terminal](#)
- [Deploy payment connectors](#)
- [Create Windows installers for payment connectors](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage Bing Maps for your organization

2/18/2021 • 2 minutes to read • [Edit Online](#)

Learn how you can manage Bing Maps for your application. When Bing Maps is turned on, people can view a map of a store, warehouse, or other location when creating orders.

## Enable Bing Maps

1. Go to **Commerce shared Parameters > Bing Maps**.
2. Select **Enable Bing Maps** to turn on Bing Maps functionality.

## Enter a Bing Maps key

Go to the [Bing Maps licensing page](#) for details about how to obtain a key.

1. Go to **Commerce shared Parameters > Bing Maps**.
2. Enter the license key in the **Bing Maps Key** field.

## Privacy notice

If you enable the Bing Maps feature, address information is automatically sent over the internet to the Bing Maps service to display an online map of the address within this application. If you click on Bing Maps within this application, you will be redirected to [www.bing.com/maps](http://www.bing.com/maps). Your use of Bing Maps is also governed by the [Bing Maps End User Terms of Use](#).

Administrators can turn the Bing Maps feature on or off under **Commerce shared Parameters > Bing Maps**. Turning the Bing Maps app off makes the feature unavailable in this application.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create and update time slots for customer pickup

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes how to create, configure, and update customer pickup time slots in Commerce headquarters.

The time slot feature gives retailers a way to define a time slot for items that the customer pickup delivery mode is turned on for. Time slots let retailers define the days and times when orders can be picked up from a store. Retailers can also define the number of orders that can be picked up during a given period. In this way, retailers can limit the number of orders that can be picked up on a given day and at a given time. The result is a better-quality experience for their customers.

## NOTE

The time slot feature is available in Microsoft Dynamics 365 Commerce version 10.0.15 and later.

The following illustration shows an example of time slot selection during e-commerce checkout.

The screenshot shows a checkout page with the following sections:

- Checkout**
- 1. PICKUP INFORMATION**
  - Pick up, San Francisco** (1 items)
  - Pickup date: 9/24/2020
  - Pickup time slot: 8:00 AM - 9:00 AM
  - Save & continue** button
- 2. PAYMENT METHOD**
- 3. CONTACT INFORMATION**
- 4. TERMS AND CONDITIONS**
- Back to shopping** button
- Place order** button

## Time slot properties

A time slot is a specific interval when a customer can choose to pick up an order from a specific store or location. The time slot management feature is available only when the customer pickup delivery mode is configured in Dynamics 365 Commerce.

A time slot is defined by using the following properties:

- **Mode of delivery** – Specify the pickup mode of delivery that the time slot applies to.
- **Minimum Days** and **Maximum Days** – Specify the earliest and latest dates that can be selected for

pickup relative to the date when the order is placed.

The **Minimum Days** property ensures that there is enough time for the retailer to process the order before it's ready for pickup. The **Maximum Days** property ensures that the user can't select a date that is too far in the future. For example, if the minimum value is set to **1**, and an order is placed on September 20, the earliest day that the order will be available for pickup is the next eligible day (September 21). In a similar way, by setting a maximum value, you can define the maximum number of days that an order can be picked up. When minimum and maximum values are defined, site users can see and select only a specific set of days during their checkout experience.

You can set the minimum value to a decimal value that is less than 1. For example, if pickup is available four hours after an order is placed, set the minimum value to **0.17** ( $= 4 \div 24$ , rounded up to two decimal places). However, if you set the minimum value to a decimal value that is more than 1, it's always rounded up to the nearest whole number. For example, a value of **1.2** will be rounded up to **2**. Similarly, if you set the maximum value to a decimal value, it's always rounded up to the nearest whole number.

- **Start Date and End Date** – Specify the start and end dates of the time slot. Each time slot entry has a start date and an end date. Therefore, you have the flexibility to add different time slots throughout the year (for example, pickups during holiday hours). If a time slot's start and end dates are changed after an order is placed, the changes won't apply to that order. When you define start and end dates, you must consider store closure dates (for example, Christmas day) and ensure that time slots aren't defined for those days.
- **Active Hours of Pickup** – Specify the period when pickup is allowed. For example, the pickup times might be between 2 PM and 5 PM every day. This property enables the pickup times to be independent of store hours. Therefore, the retailer can configure pickup times that meet its specific business requirements. When you define the active hours of pickup, you must consider store hours and ensure that pickup times aren't defined for times when the store is closed.

#### NOTE

The hours for store pickup must be defined in the time zone of the appropriate store.

- **Time Slot Interval** – Specify the duration that can be allotted to each time slot. For example, the duration of each time slot might be in increments of 15 minutes, 30 minutes, or one hour. If time slot value is 0, the time slot is available for the entire duration between the start and end time.
- **Slots Per Interval** – Specify the number of customer or orders that can be served for pickup during each time slot interval. For example, enter **1**, **2**, **3**, or any other whole number.
- **Active Days** – Specify the days of the week when the pickup time slots are active. This property lets the retailer define the days when it wants to support pickup orders.
- **Retail Channels** – Specify the retail channels. Each time slot can be associated with one or more retail stores. Depending on each store's hours of operation, one or more time slot entries can be created and associated with a channel.

Only one time slot template can be configured per channel. These channels include brick-and-mortar stores, call centers, mobile devices, and e-Commerce sites.

## Configure the time slot feature in Commerce headquarters

Time slots must be defined for each pickup mode of delivery in Commerce headquarters, so that point of sale (POS) and e-commerce channels can reference them.

- Only one time slot template can be associated with each store or channel.

- Each time slot that is created should be unique to each delivery mode in each template.
- After the time slot feature is configured, the time slot calendar will be available to the selected stores or store groups. It will also be visible at the POS, for reference.

To configure the time slot feature in Commerce headquarters, follow these steps.

1. Go to **Commerce > Channel setup > Store pickup time slot**.
2. Select **New** to create a new time slot template. To use an existing template, select the template in the left pane.
3. Enter values in the **Time Slot ID** and **Description** fields.
4. On the **Order Pickup - Time Settings** FastTab, select **Add**.
5. In the **Order Pickup - Time Settings** dialog box, define the date range, mode of delivery, active hours of delivery, active days, time slot interval, slots per interval, and other settings.

If time slots will be static for the foreseeable future, set the **End Date** field to **Never**.

#### NOTE

You can create multiple templates, but only one template can be associated with a single channel or store.

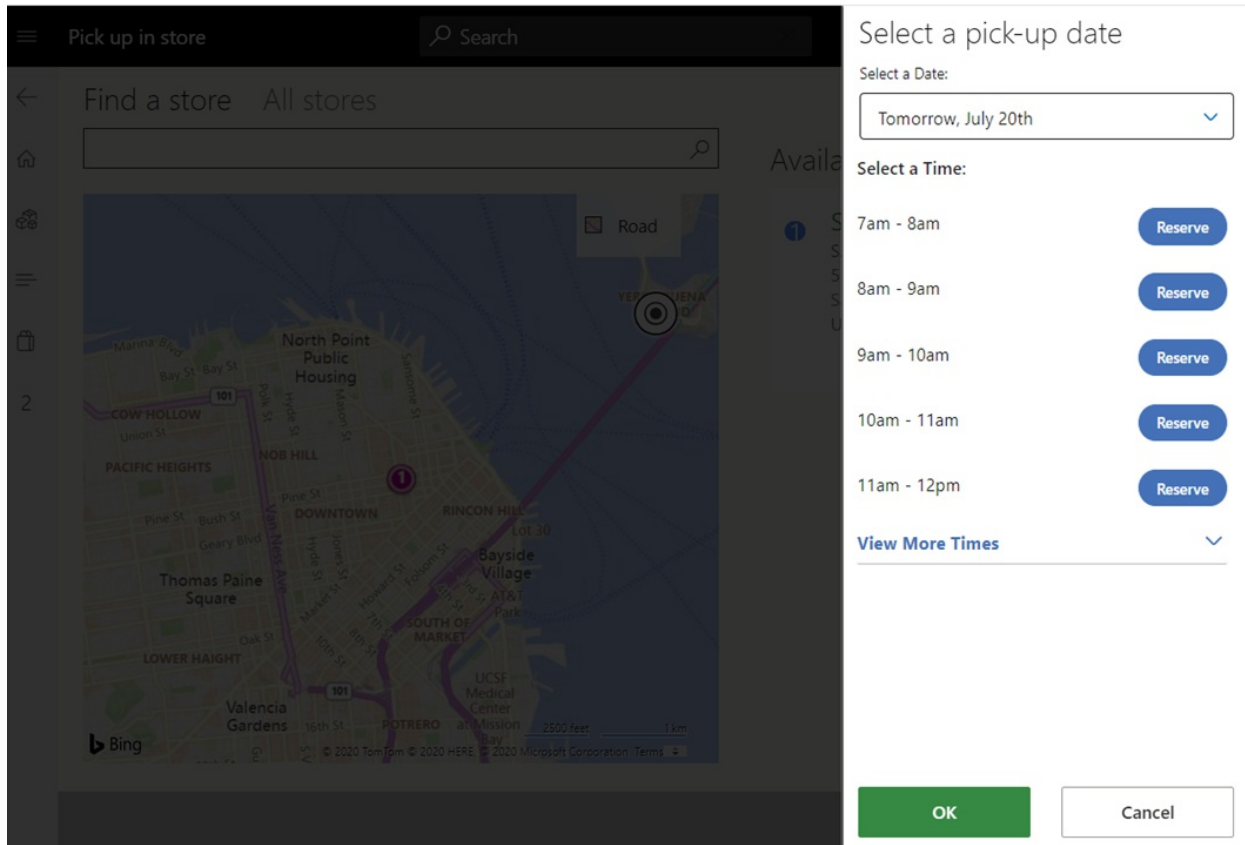
The screenshot shows the 'Order Pickup - Time Settings' dialog box. The left pane displays a table of existing time slot settings and a 'Retail Channels' list. The right pane contains configuration options for 'General', 'Setting', 'ACTIVE HOURS OF DELIVERY', 'ACTIVE DAYS', 'TIME SLOT INTERVAL', and 'SLOTS PER INTERVAL'. The 'General' section has 'Start Date' (9/22/2020), 'End Date' (9/22/2030), and 'Mode of delivery' (60). The 'Setting' section has 'Start Time' (08:00) and 'End Time' (15:00). The 'ACTIVE DAYS' section has 'Yes' selected for Monday through Sunday. The 'TIME SLOT INTERVAL' section has 'Hours' (1) and 'Minutes' (0). The 'SLOTS PER INTERVAL' section has 'Slots Per Interval' (10). 'OK' and 'Cancel' buttons are at the bottom right.

6. When you've finished, select **OK**.
7. If the time slots in a day will vary, create additional entries on the **Order Pickup - Time Settings** FastTab to ensure that the dates and times don't overlap.
8. On the **Retail Channels** FastTab, select **Add** to associate the time slot template with the stores or channels where it will be used.
9. In the **Choose organization nodes** dialog box, use the arrow buttons to select (or clear the selection of) the stores, regions, and organizations that the template should be associated with.
10. When you've finished, select **OK**.
11. On the **Distribution schedule** page, run the **1070** and **1135** jobs to sync the data to the channels.

## Time slot selection for POS orders

At the POS, when an order or order line is identified for pickup, the cashier can select the pickup store or location, and a date and time slot. If a customer has a pickup order for a different store, the cashier can select dates when the pickup will be available in that store. The store lookup will provide a reference to the dates and store times.

The following illustration shows an example of time slot selection for a POS order.



## Time slot selection for e-commerce orders

For information about how to make time slot selection available for e-commerce orders, see [Pickup information module](#).

### NOTE

Users can view or edit pickup time slots on a Commerce site's checkout page only if the pickup information module has been added to that page. If the checkout page doesn't include the pickup information module, orders will be placed without letting users specify or view time slot information.

The following illustration shows an example of an e-commerce order where a pickup time slot has been selected.

## Checkout

## 1. PICKUP INFORMATION

[Change](#)**Pick up, San Francisco** (1 items)**Pickup location**

San Francisco  
555 California St.  
San Francisco, CA, SAN FRANCI 94104  
USA

**Pickup date and time slot**

9/24/2020 8:00 AM - 9:00 AM

## 2. PAYMENT METHOD

[Change](#)

**Important Notice.** Payment services are provided by designated third-party payment providers. Payment information is for demonstration purposes only and does not process actual transactions; however, any payment information that you input will be sent to the designated third-party payment provider, and will be handled in accordance with the terms and conditions and privacy statement of the payment provider.

**Payment information**

Karen Berg Visa Card ending in 1111 Expires 10/2020

**Billing address**

Home  
Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
📞 4257058000

## 3. CONTACT INFORMATION

[Change](#)

karenb@microsoft.com

## 4. TERMS AND CONDITIONS

[Change](#) I have reviewed Fabrikam Terms and conditions.[Back to shopping](#)[Place order](#)

## Order summary

Subtotal	<b>\$158.51</b>
Tax	<b>\$12.74</b>
<b>AMOUNT DUE</b>	<b>\$188.50</b>

[Place order](#)[Back to shopping](#)

## Shopping bag

[Edit Cart](#)

## In-store pickup (1 item)

**Camel Polka-dot Trench**

Size: XS  
\$158.51  
Quantity: 1

Pick up at a store  
**San Francisco**

## Time slot selection for call center orders

In the call center app, call center agents can select the pickup store or location, as well as a date and time slot as highlighted in the following illustration.



**Sales order lines**

+ Add line + Add lines Add products Remove Sales order line Financials Inventory Update line Retail

	Ty...	Variant number	Item number	Product name	Deliver remainder	Delivery type	Adjusted unit ...
			91002	Round Wrap Style Sunglasses	Apply Pickup Time Slot	Stock	0.00000

PROCESS  
Registration  
Pick  
STATUS  
Reset fulfillment status  
DOM  
Suggest fulfillment location  
Reject fulfillment request

---

**Line details** 0.00

General Setup Address Product Packing **Delivery** Sourcing Price and discount Financial dimensions

**DELIVERY DATE**  
Delivery alternatives  
Requested ship date: 10/15/2020  
Requested receipt date: 10/15/2020  
Simulate delivery dates

Confirmed ship date  
Confirmed receipt date  
Underdelivery: 0.00  
Mode of delivery: 60  
SHIPPING LOCATION TIME ZONE  
Time zone: (GMT-08:00) Pacific Time (US & ...)  
**PICKUP TIME SLOT**  
Pickup Time Range: 09:00 - 10:00

**MISC. DELIVERY INFO**  
Overdelivery: 0.00  
**CARRIER INFORMATION**  
Shipping carrier  
Carrier service

## Additional resources

### Pickup information module

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product information overview

2/18/2021 • 5 minutes to read • [Edit Online](#)

## NOTE

Effective November 2020:

- Common Data Service has been renamed to Microsoft Dataverse. For more information, see [Power Automate Blog](#).
- Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. For more information, see [Terminology updates](#).

This topic will be updated soon to reflect the latest terminology.

This topic provides information about product information management. Product information management works with a shared product definition, categorization, and identifiers across all legal entities, and also specific configurations of a product, to fit into the business processes.

Product information is the backbone of supply chain and commerce applications across all industries. It refers to processes and technologies that focus on centrally managing information about products (for example, across supply chains). It's crucial that shared product definitions, documentation, attributes, and identifiers be used. In the various modules of a business solution, product-specific information and configuration are required in order to manage the business processes that are related to specific products, product families, or product categories.

## Product definition

A product is primarily defined by a product number, name, and description. However, other data is also required in order to describe a product or service:

- Product type: Item or service
- Product subtype: Distinct products or product masters
- Definition of the product variant model:
  - Product dimensions and dimension groups
  - Product nomenclature
  - Product configuration models
- Association of the product with one or more categories
- Definition of the product and category attributes
- Product images
- Attachments
- Units of measure and related conversions
- Translations for all names and descriptions

## Distribution, export, and import of product data

The product definition can be created in Supply Chain Management. It can also be imported from product lifecycle management (PLM), product data management (PDM), or product information management (PIM)

systems. When more than one instance of Supply Chain Management is used, one instance is typically used as the master of the product data for all other instances. This approach is supported by a large set of data entities that enable the export and import of product definition data from one instance to another.

To support the distribution of product data to many instances, Supply Chain Management lets you use Microsoft Dataverse. The product definitions can be exported from an instance of Supply Chain Management to Microsoft Dataverse. The product definitions can then be used to provision other business applications, such as Dynamics 365 Sales, with product data.

Note that, in dynamic and agile organizations, product information data changes every day. Therefore, maintenance of accurate and actual product data is a critical business process on its own.

## Product masters and product variants

In an agile world, where products must be quickly adapted to customer requirements, product definitions specify a set of products instead of distinct products. In Supply Chain Management, those generic products are known as *product masters*. Product masters hold the definition and rules that specify how distinct products are described and behave in business processes. Based on these definitions, distinct products can be generated. These distinct products are known as *product variants*.

A product master is associated with a product dimension group and a configuration technology to specify the business rules. The product dimensions (Color, Size, Style, and Configuration) are a specific set of attributes that can be used throughout the application to define and track specific behaviors of the related products. These dimensions also help users search for and identify the products.

## Configuration technologies

You can choose among three configuration technologies:

- The predefined variants are defined by predefined product dimensions. The variant definition includes the definition of a specific valid combination of dimensions, such as Color, Style, and Size. Each combination produces a distinct product variant.
- The dimension-based configuration is typically used in manufacturing scenarios and lets you use the Configuration dimension in the definition of the bills of materials (BOMs). After a specific configuration is selected, the system uses the subset of BOM lines that are valid for that configuration for planning and production. This concept is also known as *global BOM*, because one shared BOM is used for all configurations of a product.
- The constraint-based configuration uses a product configuration model to describe all possible attributes and components that are required in order to describe all possible variants of a product in a single model. The constraints of combinations of attributes can be described through regular expressions or table-based constraints. Configuration models and configurators become more important in product information management and are used across all industries.

When you plan the implementation of Supply Chain Management, it's very important that you choose the correct configuration technology for a business process. A product can't be converted from one model to another after implementation.

## Product variant model definition workspace

The **Product variant model definition** workspace gives an overview of the product masters. It also shows the status of the release of masters and related variants to specific legal entities.

## Released products

The products that are released to a specific legal entity are known as *released products*. Products can be

released in bulk to one legal entity or many legal entities at a time. Because various properties and attributes of the products might have to be added per legal entity, the **Released product maintenance** workspace lets you monitor and complete the recently released products in each legal entity, or in the suborganizations of a legal entity.

### **Released product maintenance workspace**

You can configure the **Released product maintenance** workspace from the **Configure my workspace** menu item. Select a category hierarchy and category to filter the workspace by. To adjust the relevant product data in the workspace, you can also define, in days, the time fences for **Recently released products** and **Stopped released products**.

The workspace consists of a summary of tiles and two lists. The **Open cases** list shows product change cases that have products in the selected product category hierarchy that aren't completed and closed. The **Recently released** list shows products that have been released within the time fence that is set in the workspace configuration. For each item in the list, validation is run, and the validation status is shown. This status might indicate that the required configurations for the legal entity hasn't been completed. From the list, you can directly access the **Released product details**, **Product attribute maintenance**, **Product category maintenance**, **Default order settings**, and **Text translations** pages to complete the required configuration of the product.

### **Manually creating a new released product**

You can manually create a released product in a single run, depending on the organization's business processes and any rules about whether this function should be used. This function creates a new product and automatically releases it to the current legal entity. To create a new product, click **Released products** in the **Released product maintenance** workspace or on the **Released product** list page.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up retail products

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes how to set up products in Dynamics 365 Commerce.

Before you can offer products for resale in your commerce channels, you must create and configure the products. Commerce creates organization-wide products in the product master. You can create the products, define the product properties and attributes, and assign the products to commerce category hierarchies. To make the products available to your channels and add them to an active assortment, you must release the products to the legal entities where they are available. To set up the products that you sell by using channels, complete the following tasks.

1. **Define a product hierarchy.** By using the category hierarchy features in Commerce, you can define category hierarchies to group and categorize the products that you distribute to your channels. User-defined and system attributes can be defined at the category level. Then, all products that are assigned to the category inherit those attributes. Multiple category hierarchies can be defined, and each product can be assigned to multiple hierarchies. However, in a single hierarchy, each product can be assigned to only one category.
2. **Add products and product variants to the product master.** Products that are added to the product master represent a global list of products. You can add products manually, one at a time, or you can import product data from your vendors.
3. **Release the products to legal entities.** Only products that have been released to legal entities can be made available to your channels. When you first define a product, you define it on an organization-wide level. You can then select one or more legal entities to release the product to. The product then becomes available to multiple channels across your organization. You can use this functionality to create a product one time, add and update product attributes and properties in one place, and then distribute the product across your organization, to the channels where it's available.
4. **Add products to assortments.** An assortment represents a collection of products that you offer in your channels. You can define one or more assortments, and each product can be assigned to one or more assortments. To assign products to channels, you assign the assortments to those channels. When you create an assortment, you can add products that haven't yet been released to a legal entity. However, you must release the products to a legal entity before those products can be made available to the channels.
5. **Add products to navigation hierarchies.** Before products can be browsed online or in point of sale (POS), they must be categorized in a Commerce navigation hierarchy.
6. **Add products to catalogs.** Although this step is optional for POS, online stores require that products be included in at least one catalog.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce hierarchies

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes hierarchies in Dynamics 365 Commerce.

You can create a category hierarchy to organize the products that you sell through your channels. You can use product hierarchies to categorize or group products. You can then use these products to create product assortments and customer loyalty programs. You can also assign product attributes and properties, assign a pricing structure, include the products in product promotions, and use the products for reporting. You can create one category hierarchy to represent all the products and categories in your organization, and then use that category hierarchy for multiple purposes. Alternatively, you can create multiple category hierarchies for special purposes, such as product promotions. When you create a product hierarchy, you must assign a category hierarchy type to identify the purpose of the category hierarchy. For example, only product hierarchies that are assigned the **Commerce navigation hierarchy** type are referenced when you browse products by category online or in point of sale (POS).

## Hierarchy types

The following table lists the types of category hierarchies that are available and the general purpose of each type.

CATEGORY HIERARCHY TYPE	PURPOSE
Product hierarchy	Use this hierarchy type to define the overall product hierarchy for your organization. You can use this hierarchy type for merchandising, pricing and promotions, reporting, and assortment planning. Only one product hierarchy can be assigned this hierarchy type.
Supplemental hierarchy	Use this hierarchy type for any additional category hierarchies that you want to create. For example, in the spring, you have a promotion for swimwear. Therefore, you include your swimwear products in a separate category hierarchy and apply the promotional pricing to the various product categories.
Navigation hierarchy	Use this hierarchy type to group and organize products into categories so that the products can be browsed online or in POS.

By using a category hierarchy to structure your products, you can set up and maintain product attributes and properties at the category level. These attributes and properties include settings for product dimensions and POS settings. Any products that you assign to the categories automatically inherit the attributes and properties that you define. You can also copy the property settings for any product to multiple products in a selected category at the same time.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a new product hierarchy

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a new product hierarchy in Microsoft Dynamics 365 Commerce.

## Overview

Dynamics 365 Commerce supports multiple retail channels. These retail channels include online stores, call centers, and retail stores (also known as brick-and-mortar stores). Each retail store channel can have its own payment methods, price groups, point of sale (POS) registers, income accounts and expense accounts, and staff. You must set up all of these elements before you can create a retail store channel.

A Commerce product hierarchy is used to define the overall product hierarchy for your organization. You can use a Commerce product hierarchy for merchandising, pricing and promotions, reporting, and assortment planning. Only one Commerce product hierarchy can be assigned per organization.

## Create and configure a product hierarchy

To create and configure a Commerce product hierarchy, follow these steps.

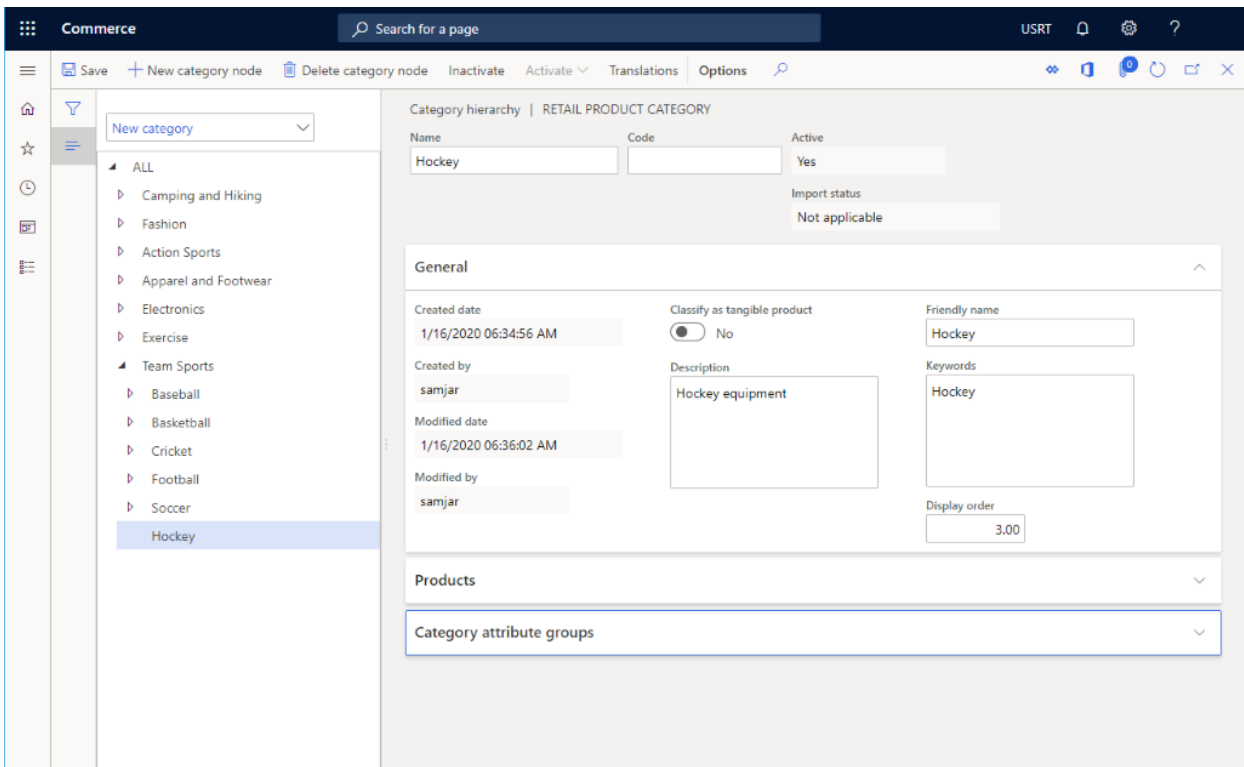
1. In the navigation pane, go to **Modules > Retail and commerce > Products and categories > Commerce product hierarchy**.
2. If no hierarchy exists yet, on the **Action** pane, select **New** to create the root of the hierarchy.
3. Under **General**:
  - a. In the **Name** box, enter a name.
  - b. In the **Description** box, enter a description.
  - c. In the **Friendly name** box, enter a friendly name.
  - d. Set **Active** to **Yes**.

## Add hierarchy nodes

To add hierarchy nodes, follow these steps.

1. On the action pane, select **Edit category hierarchy**.
2. Select the parent node you want to add a new node to, and then select **New category node**.
3. In the **General** section provide a **Name**, **Description**, **Friendly name** and **Keywords**.
4. Under **General**:
  - a. In the **Name** box, enter a name.
  - b. In the **Description** box, enter a description.
  - c. In the **Friendly name** box, enter a friendly name.
  - d. In the **Keywords** box, enter relevant keywords.
  - e. In the **Display order** box, enter a number for the display order (optional).
5. On the action pane, select **Save**.
6. Repeat the steps above to add additional nodes.

The following image shows the creation of a new product hierarchy node.



## Other settings

Category attribute groups can also be assigned to each group as required.

## Additional resources

[commerce hierarchies](#)

[Manage product categories and products](#)

[Change the sort order for merchandizing entities](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Manage product categories and products

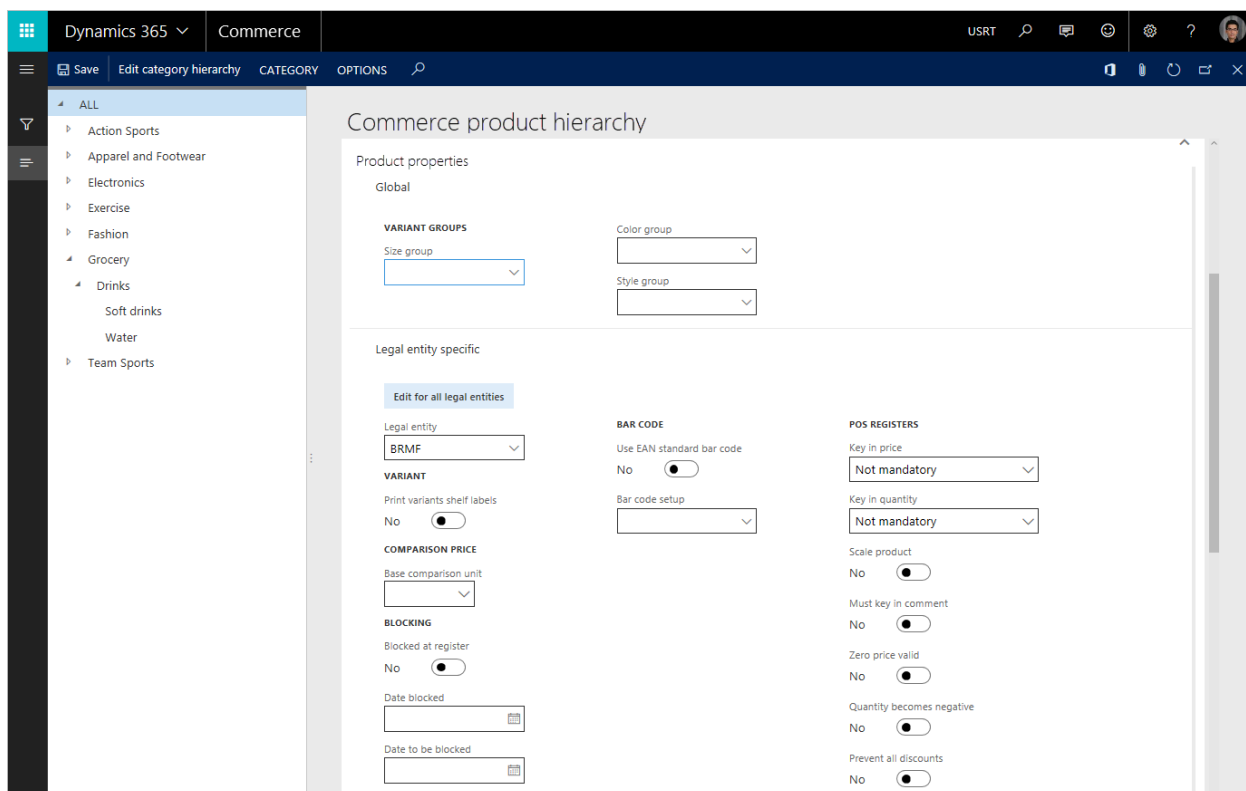
2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes an enhanced way to manage product categories and products in Dynamics 365 Commerce. The enhancements let merchandising managers view a structure of product properties that is shared between the product hierarchy and released product details.

To learn more about how to manage product categories, in the **Category and product management** workspace, select the **Commerce product hierarchy** tile.

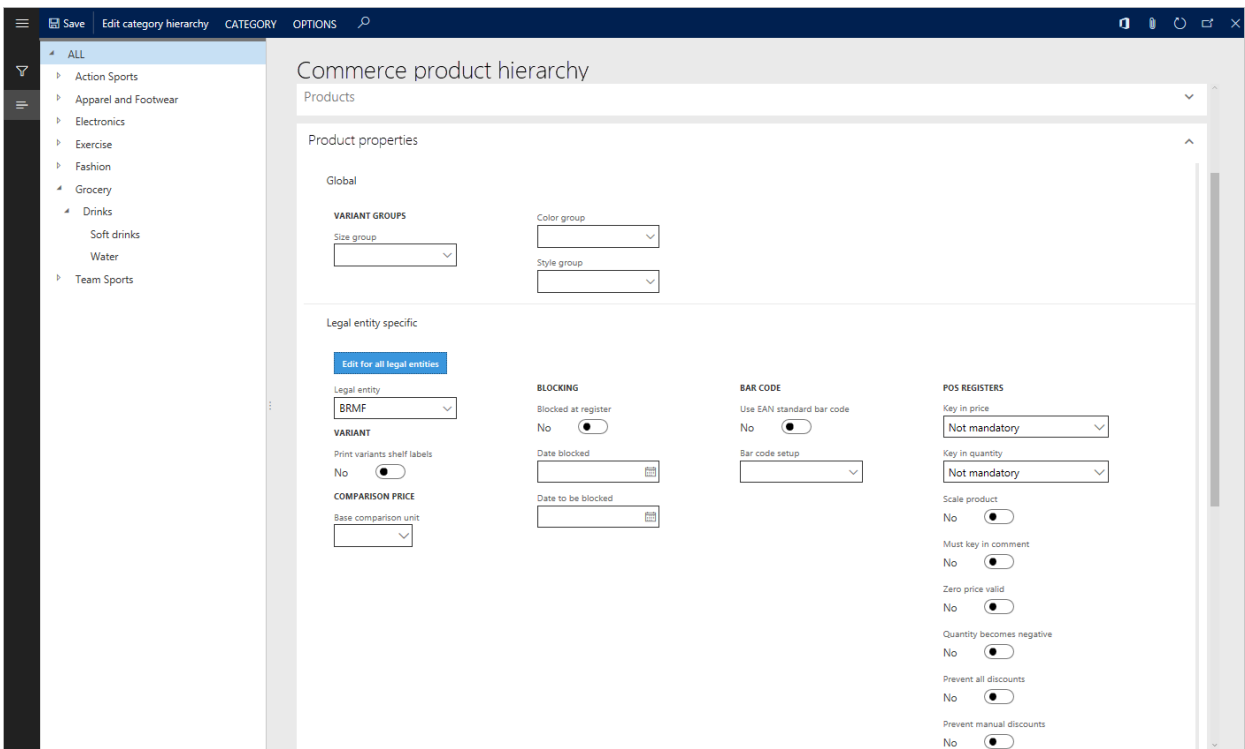
Notice the enhanced structure of the **Commerce product hierarchy** page that appears. In previous versions of the app, product properties were divided into *basic product properties* and *Retail product properties*, based on the scope of their applicability. Retail product properties are *global* in their scope of applicability. In other words, for a given product property, the same value is shared across all legal entities. By contrast, basic product properties are *legal entity–specific*. In other words, for a given basic product property, the value can differ across legal entities, depending on the individual business requirements of each legal entity.

In the enhanced product category structure, product properties are logically separated based on their applicability in a group, to reflect the structure of the released product details form structure.

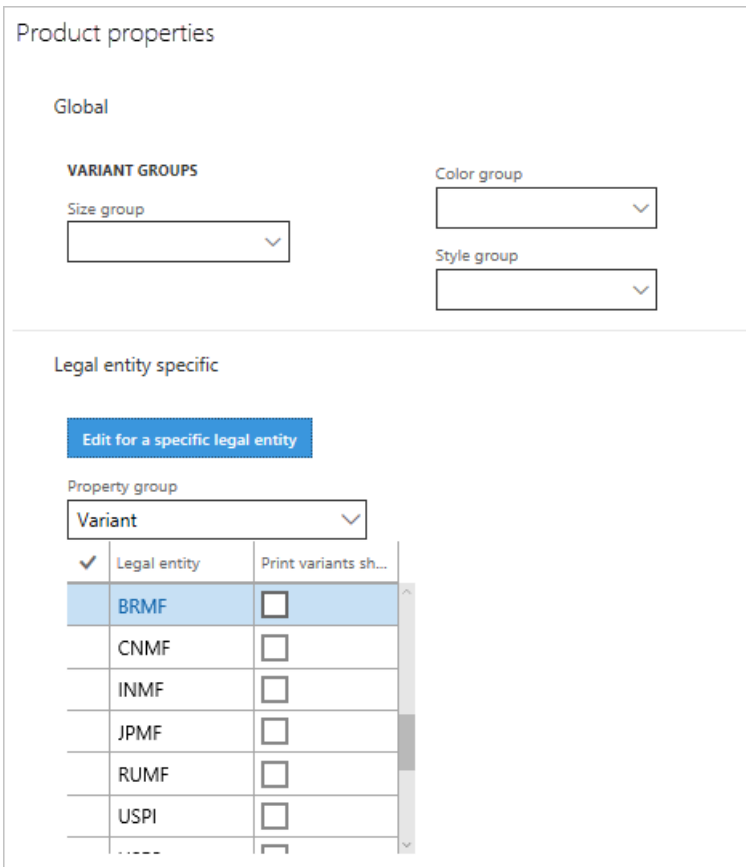


You can switch between managing legal entity–specific properties across all legal entities and managing them for a specific legal entity.

To manage properties across all legal entities, select **View for all legal entities** (or **Edit for all legal entities**).



To manage properties for a specific legal entity, select **View** for a specific legal entity (or **Edit** for a specific legal entity).

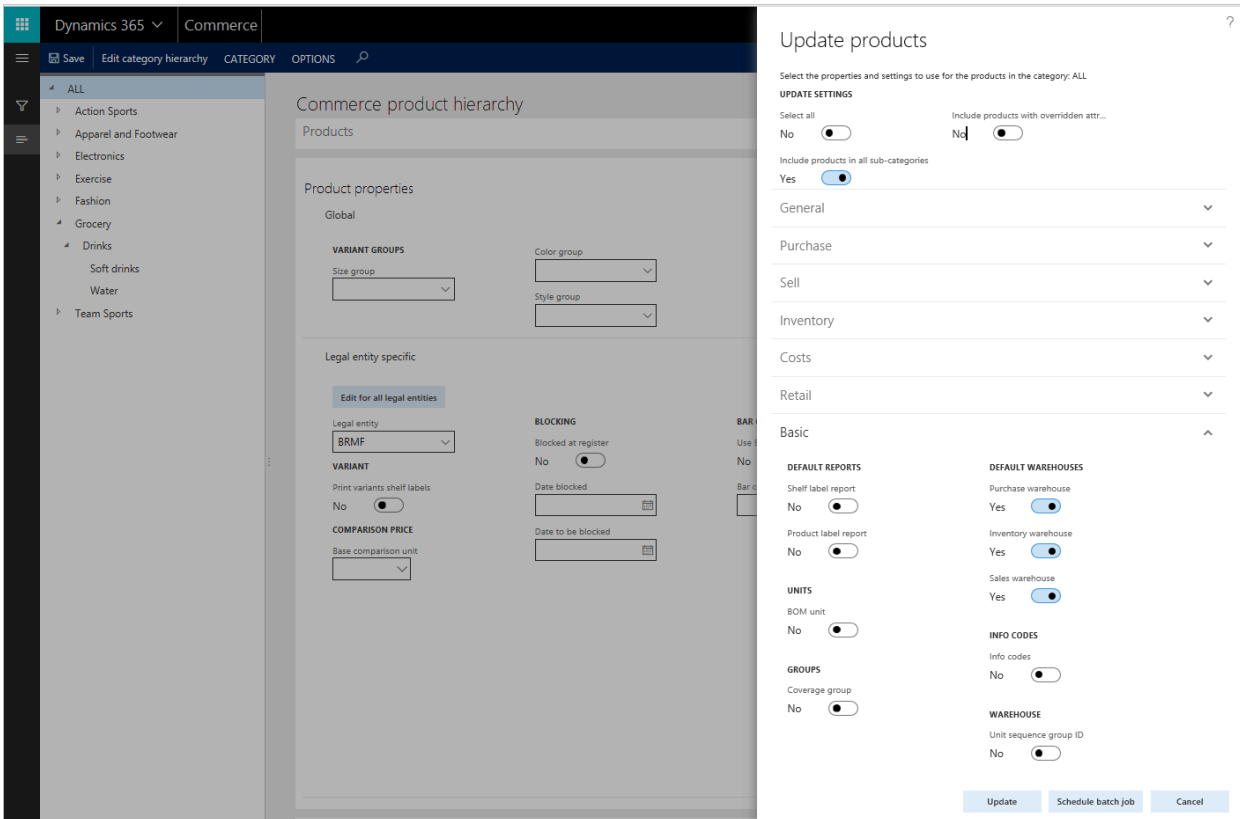


Additionally, in the enhanced product category structure, a merchandising manager can now define default values for an additional set of product properties at the level of the individual category. Then, when products are created, they inherit default values for their product properties, based on the association of those properties with an individual category in the product hierarchy. These inherited product properties can also be modified for each product to meet individual business requirements.

Selecting properties to update products on the Commerce product

# hierarchy page

You can use the new enhanced structure for product properties to select updated product properties that must be pushed to the associated products. On the **Commerce product hierarchy** page, on the Action Pane, select **Category**, and then select **Update products** to open the **Update products** dialog box.



## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Change the sort order for merchandising entities

2/18/2021 • 2 minutes to read • [Edit Online](#)

Retailers consider product discovery a primary tool for customer interaction across all channels. Various functionality can help customers easily discover products. For example, they can browse categories, search, and filter.

This topic explains the concepts that are related to controlling the display order for various merchandising-related entities. It also explains how to change the sort order.

## Overview

The support for sorting various merchandising-related entities has been enhanced. This support is now better aligned with existing customer scenarios that previously required extensions from implementation partners.

In versions of Retail that are earlier than version 10.0.5, the sort order for categories in the navigation hierarchy was alphabetical. The new custom sort order functionality lets merchandising managers configure the sort order for various merchandising-related entities across all end-user clients. These clients include headquarters (HQ) and call centers.

## Configure the display order for categories in the product hierarchy

Before you can complete this procedure, demo data must be installed in your environment.

1. Go to **Retail and Commerce > Products and categories > Commerce product hierarchy**.
2. Click **Edit category hierarchy**.
3. Click **Edit**.
4. In the tree, expand **ALL > Action Sports**.
5. In the tree, expand **ALL > Team Sports**.
6. In the **Display order** field, enter a number. (The number can be negative.)
7. Repeat steps 4 through 6 for any additional categories that you want to change the order of.

The display order for the channel navigation hierarchy will be reflected in HQ for the commerce product hierarchy and released products by category.

**Finance and Operations** Commerce product USRT

Category Options

- ALL
  - Team Sports
    - Soccer
      - Soccer Balls
      - Soccer Coach & Team Equipment
      - Soccer Shin Guards
    - Baseball
    - Basketball
    - Cricket
    - Football
    - Action Sports
      - BMX Biking
      - In-Line Skating
      - Skateboarding
    - Apparel and Footwear
    - Exercise
    - Fashion
    - Electronics
      - Audio
      - Cameras
      - Computers
      - Home appliances
      - Phones
      - TV and video

Commerce product hierarchy

**General**

Name: Team Sports Modified date: 7/26/2019 10:11:45 AM

Commodity code: Description: Team Sports

Active:  Yes Friendly name: Team Sports

Activated date: 10/6/2016 12:59:35 PM Search text: Team Sports

Display order: -10.00

Products

Retail product properties

General product properties

Purchase product properties

Sell product properties

Manage inventory product properties

**Finance and Operations** released product USRT

Product Purchase Sell Manage inventory Engineer Plan

Maintain: Apply template, Validate; Modify hierarchy: Add products, Remove products; New: Product, Template; Product master: Released product variants, Product dimensions; Languages: Translations; Set up: Dimension groups, Reservation hierarchy, Counting reason code policy; Product attributes, Related Unit cor

Released product details

Category hierarchy: Retail Product Category

- ALL
  - Team Sports
    - Soccer
    - Baseball
    - Basketball
    - Cricket
    - Football
    - Action Sports
      - BMX Biking
      - In-Line Skating
      - Skateboarding
    - Apparel and Footwear
    - Exercise
    - Fashion
    - Electronics

Filter

Item number ↑	Product name	Search name	Product
0001	Youth Accessory Combo Set	YouthAccessoryComboS	Item
0002	Adult Helmet Accessory Combo...	AdultHelmetAccessory	Item
0003	Signature Mountain Bike Tire	SignatureMountainBik	Item
0004	Premium Mountain Bike Tire	PremiumMountainBikeT	Item
0005	Basic Inner Tube	BasicInnerTube	Item
0006	Inner Tube Patches	InnerTubePatches	Item
0007	Full Finger BMX Gloves	FullFingerBMXGloves	Item
0008	Mesh BMX Gloves	MeshBMXGloves	Item
0009	Premium Full Finger Gloves	PremiumFullFingerGlo	Item
0010	Signature BMX Gloves	SignatureBMXGloves	Item
0011	Curved Grind Rail	CurvedGrindRail	Item
0012	Straight Grind Rail	StraightGrindRail	Item
0013	Adult Baseball Infield Glove	AdultBaseballInfield	Item
0014	Adult Baseball Outfield Glove	AdultBaseballOutfiel	Item
0015	Youth Utility Baseball Glove	YouthUtilityBaseball	Item
0016	Adult Catchers Mitt	AdultCatchersMitt	Item
0017	Youth Catchers Mitt	YouthCatchersMitt	Item
0018	Adult First Base Mitt	AdultFirstBaseMitt	Item

Related information

# Configure the display order for categories in the channel navigation hierarchy

Before you can complete this procedure, demo data must be installed in your environment.

1. Go to **Retail and Commerce > Products and categories > Channel navigation categories**.
2. In the list, select the **Fashion navigation** hierarchy.
3. Click **Edit category hierarchy**.
4. Click **Edit**.
5. In the tree, select **Fashion > Womenswear > Womens Shoes**.
6. In the **Display order** field, enter a number.
7. In the tree, select **Fashion > Womenswear > Tops**.

Likewise, you can define the sort order for the sub-categories.

8. In the tree, select **Fashion > Menswear > Casual Shirts**.
9. In the **Display order** field, enter a number.
10. In the tree, select **Fashion > Menswear > Coats & Jackets**.
11. In the **Display order** field, enter a number.
12. Repeat for any additional categories that you want to change the order of.

The display order for the channel navigation hierarchy is reflected in HQ, catalog, and channels.

The screenshot displays the SAP Finance and Operations interface. The top navigation bar includes 'Finance and Operations', a search bar with 'channel naviga', and user information 'USRT'. The main content area is titled 'Retail product hierarchy | FASHION NAVIGATION HIERARCHY'. On the left, a category tree is shown with 'Fashion' expanded to 'Womenswear', which is further expanded to 'Womens Shoes', and 'Sweaters' is selected. The right pane shows the configuration for the 'Sweaters' category. The 'General' tab is active, displaying fields for Name (Sweaters), Modified date (7/26/2019 10:12:59 AM), Commodity code, Description (Sweaters), Active status (Yes), Friendly name (Sweaters), Activated date (3/6/2017 12:12:52 AM), Search text (Sweaters), and Display order (-9.00). The 'Display order' field is highlighted with a red box. Below the 'General' tab, there are sections for 'Products' (81216), 'Product attribute groups', and 'Category attribute values'.

**Finance and Operations** | Catalog | USRT | ? | AH

[Edit](#) | [New](#) | [Delete](#) | [Configuration status](#) | [Workflow](#) | **Catalogs** | [Options](#)

[Maintain](#) | [Validate](#) | [Publish](#) | [Products](#) | [Pricing](#) | [Media](#) | [Call center](#) | [List](#) | [Language](#)  
[Copy](#) | [Validate catalog](#) | [Publish](#) | [Inactivate](#) | [Add products](#) | [Price groups](#) | [Images](#) | [Catalog pages](#) | [Open in Excel](#) | [Translati](#)  
[View results](#) | [Activate](#) | [Retract](#) | [Remove products](#) | [Price list](#) | [Payment schedules](#) | [Free products](#)

### Fabrikam Base Catalog

Category hierarchy: Fashion navigation ...

- Fabrikam Base Catalog
  - Uncategorized
  - Fashion
    - Womenswear
      - Womens Shoes
      - Sweaters
      - Tops
      - Coats
      - Dresses
      - Skirts
      - Womens Jeans
      - Fashion Accessories
      - Menswear

General	
Name	Fabrikam Base Catalog
Description	Fabrikam Base Catalog
Owner	Emma Harris
Catalog number	C0003
Take snapshot	<input type="checkbox"/> No
Status	Draft
Effective date	11/20/2013
Expiration date	Never
Last published date	2/24/2019 10:20:36 PM
Last modified date	6/8/2019 02:55:57 PM

Retail channels: None

Source codes: [Dropdown]

Scripts: [Dropdown]

All categories | Search | 1 - HOUSTON38

Womenswear >

- Womenswear
- Womens Shoes
- Sweaters
- Tops
- Coats
- Dresses
- Skirts
- Womens Jeans

Fashion Accessories >

- Fashion Accessories
- Fashion Sunglasses
- Gloves & Scarves
- Handbags
- Jewelry
- Watches

Menswear >

- Menswear
- Casual Shirts
- Dress Shirts
- Mens Shoes
- Pants
- Suits & Sportcoats
- Mens Jeans
- Coats & Jackets

**NOTE**

By default the custom sort order feature is turned off. To learn how to turn on this feature and other features, see [Feature management](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product dimensions

2/18/2021 • 7 minutes to read • [Edit Online](#)

There are five product dimensions: color, configuration, size, style, and version. You combine product dimensions in dimension groups and assign dimension groups to product masters. The combinations of product dimensions determine how product variants are defined.

Product dimensions are characteristics that serve to identify a product variant. You can use combinations of product dimensions to define product variants. You must define at least one product dimension for a product master in order to create a product variant.

## Product variants

Product variants are also referred to as items. An item is a tangible product, which isn't the same as a service. It's also possible to define a product master with the service type. By using the service type, you can specify product variants that include services. For example, you can specify a product master for consultancy work and product variants for work that is performed by senior consultants and junior consultants.

## Product dimensions

A product variant can be generated based on the product dimension values.

Product dimension values for the size, color, and style dimensions can be created in the following locations:

- **Size page** (**Product information management > Setup > Dimension and variant groups > Sizes**)
- **Color page** (**Product information management > Setup > Dimension and variant groups > Colors**)
- **Style page** (**Product information management > Setup > Dimension and variant groups > Styles**)

Product dimension values for the configuration dimension are typically created by using either the Product configurator or the Dimension-based configurator.

Product versions are usually created for specific versions as the product evolves during its lifecycle. Product versions are covered in detail later in this topic.

Product dimensions can also be created and maintained on the **Product dimensions** page, which can be accessed from the following locations:

- Go to **Product information management > Products > Product masters**. On the Action Pane, select **Product dimensions**.
- Go to **Product information management > Products > All products and product masters**. Select a product master. On the Action Pane, select **Product dimensions**.
- Go to **Product information management > Released products**. Select a product master. On the Action Pane, on the **Product** tab, in the **Product master** group, select **Product dimensions**.

The number of variants that you can create for an item is limited by the number of possible product dimension combinations.



**TIP**

When you use a product on an order line, for example, you select the product dimensions to identify the product variant that you want to work with.

## Example

A company sells denim jeans. The item, *Jeans*, uses the color and size product dimensions. The jeans are sold in three different colors and six different sizes. The colors are blue, black, and brown. The sizes are XS, S, M, L, XL, and XXL. Not all sizes are available in all three colors. If all combinations were available, there would be 18 different types of jeans. However, in this example, only the following nine product variant combinations are produced.

COLOR	SIZE
Blue	XS
Blue	S
Blue	M
Black	M
Black	L
Black	XL
Brown	L
Brown	XL
Brown	XXL

## The version product dimension

Version is a product dimension that is intended to help you maintain and track multiple versions of a product throughout the supply chain. Version tracking is essential to the success of manufacturers that operate in a world of constantly shrinking product lifecycles, increased quality and reliability requirements, and increased focus on product safety.

As a standard product dimension, version will behave similarly to the existing product dimensions (size, style, color, and configuration). Therefore, you can use it for other purposes besides tracking product versions.

### Turn on the version dimension

#### Before you turn on the version dimension

When you turn on the version dimension, some functionality could become broken or stop working as expected if you've installed other solutions that add customizations to the inventory dimensions. For the version dimension to be fully functional, you might have to update those solutions so that they include the version dimension in their references to the inventory dimensions.

When you're testing your solutions for compatibility with the version dimension, look for the following elements:

1. **Functionality:** Most importantly, any customizations that involve the inventory dimensions must be assessed to ensure that they can work in conjunction with the version dimension.
2. **References to the inventory dimensions:** Look out for references to the inventory dimensions (that is, places where the dimensions are explicitly referenced). References to `InventDimId` should work out of the box, but look out for references to style or color. For example, be sure to check the following elements:
  - API calls in extended classes
  - All references to specific inventory dimensions in extension code (This code must float the version dimension together with the style, color, and size dimensions.)
3. **References to obsolete API calls:** In its introduction of the version dimension, Microsoft has tried to make as few APIs as possible obsolete. The few APIs that have been made obsolete will issue a warning when you turn on the **Product dimension - version** configuration key. Calls to those API must be fixed in your extended solutions before you turn on the version dimension in a production system. Here are the version-specific obsolete APIs:
  - `RetailTransactionServiceInventory::getProductRecordId`
  - `EcoResProductNumberIdentifiedProductVariantEntity::find`
  - `EGAISAlcoholProduction_RU::findByItemDim`
  - `PCVariantConfiguration::findByProductMasterAndDimensions`
4. **Maps:** If any maps use the inventory dimensions, the corresponding relation mapping to these maps must be updated so that they include the version dimension. In the extended model or table extensions, look out for tables where the fields include inventory dimensions.
5. **Microsoft Dynamics 365 Commerce functionality:** After it's turned on, the version dimension will appear throughout the Commerce-specific code in Dynamics 365 Supply Chain Management. However, the version dimension isn't yet supported by the Commerce channel database or in the Point of Sale (POS) or e-Commerce applications. These Commerce-specific applications won't support users selling/shipping or returning/receiving inventory by version dimension. Inventory availability lookup functions won't discern inventory by version dimension in Commerce apps. This behavior resembles the current behavior of the config dimension throughout Commerce.

#### Turn on the version dimension

Before you can use the version dimension, it must be turned on in your system. This task requires admin permissions.

1. Go to **System administration > Workspaces > Feature management**.
2. Turn on the feature that is named *Product dimension version*. (For more information, see [Feature management](#).)
3. Put your system into [Maintenance mode](#).
4. Go to **System administration > Setup > License configuration**.
5. On the **Configuration keys** tab, expand **Trade**, and select the check box for **Product dimension - version**.
6. Turn off [Maintenance mode](#).

#### Areas where the version dimension isn't supported

The following areas don't support the version dimension (you can still use these areas but you won't be able to add versioned products (products where the version dimension is used) to them). For example, you can't add a versioned item to a vendor catalog. This is because adding products with the version dimension to these areas would cause breaking changes.

- Cost object monthly statement
- Cost object statement cache
- MCR sales statistics per item

- Vendor catalogs
- EcoResProductDimensionGroupEntity

In addition, the order creation and order processing features in Commerce (for example, for POS, call center, and e-commerce orders) don't support the version dimension. There is no confirmed timeline as to when Commerce orders will be enhanced to support it.

### Functional characteristics of the version dimension

The version dimension works like the other product dimensions. However, because of its specific nature, and because it's intended to maintain and track multiple versions of a product, it behaves slightly differently. Here are some of the differences and similarities:

- **There is no version group.**

Although the concept of groups exists for size, color, and style (color group, size group, and style group), no version group concept exists. Groups let you predefine the applicable values so that when, for example, you assign a color group to a product, the product can use all the colors in that color group. This concept doesn't apply to the version dimension, because the versions that a product takes aren't predefined when the product is created. Instead, versions are created during the lifecycle of the product, as they are required. Typically, if the form, fit, and function of the product remain the same, you create a new version instead of a new product.

- **Product variant suggestions work as they currently do.**

Product variant suggestions will create suggestions for all version dimension values, just as they do for other dimensions. The process of creating and releasing versioned products is the same as it is for products that use other dimensions. When you create a versioned product, the first version (V1) will be created as a product dimension, and the variants will be released. As the product changes and a new version is needed, the new version value (V2) will be added, and the required variants will be released. There is no expectation that all the versions (V1, V2, and V3) will be created in advance for the product.

#### IMPORTANT

If you turn on and use the version dimension, some solutions that reference the inventory dimensions might stop working as expected. To confirm and fix these issues, contact the independent software vendor (ISV) for your affected solutions. For more information, see [Enable the version dimension](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a variant group

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a size, style, or color variant group for a product in Microsoft Dynamics 365 Commerce.

## Overview

Dynamics 365 Commerce supports multiple variants for products. It is ideal to set up variant groups for different product categories. For example, a size group can be created for t-shirts with sizes extra small, small, medium, large, and extra large, or a color group could be created to include all available colors of a product. Variant groups should be added before products are added.

In this topic, a size group will be created and configured. Similar procedures can be used for adding and configuring style groups and color groups.

## Create a size group

To create a size group, follow these steps.

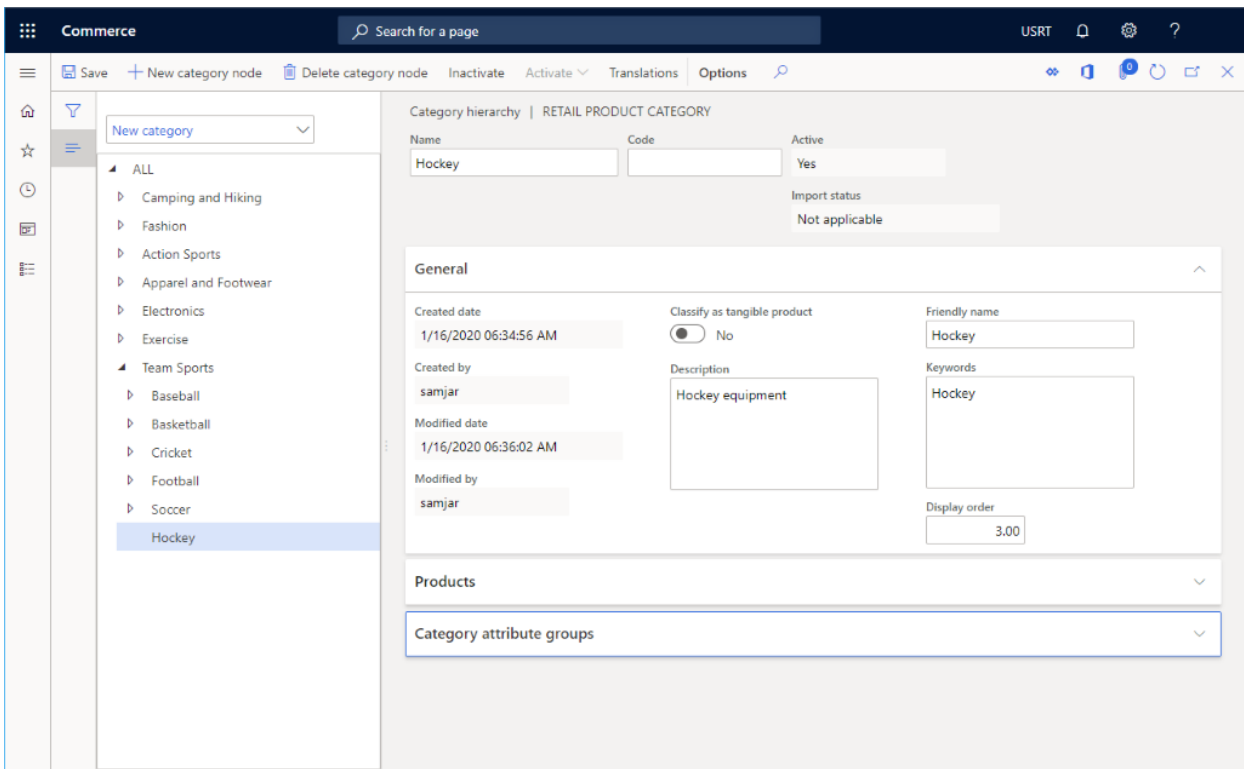
1. In the navigation pane, go to **Modules > Retail and commerce > Products and categories > Variant groups > Size groups**.
2. On the action pane, select **New**.
3. In the **Size group** box, enter a name for the size group.
4. In the **Description** box, enter an appropriate description.
5. On the action pane, select **Save**.

## Add attributes to the size group

To add attributes to a size group, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Products and categories > Variant groups > Size groups**
2. In the navigation pane, select a size group.
3. Under **Size group lines**, select **Add**.
4. In the **Size** box, enter a string representing the size (for example, "XL").
5. In the **Size name** box, enter a name for the size (for example, "Extra Large").
6. In the **Replenishment weight** box, enter a number representing the replenishment weight.
7. In the **Number in bar code** box, enter a number representing the bar code.
8. In the **Display order** box, enter a number representing the display order.
9. When finished adding sizes, select **Save** on the action pane.

The following image shows an example of a size group for "casual shirt sizes".



## Additional resources

[Product information overview](#)

[Set up retail products](#)

[Product dimensions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a new product in Commerce

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a new product in Microsoft Dynamics 365 Commerce.

## Overview

A product is primarily defined by a product number, name, and description. However, other data is also required in order to describe a product or service:

## Create a new product

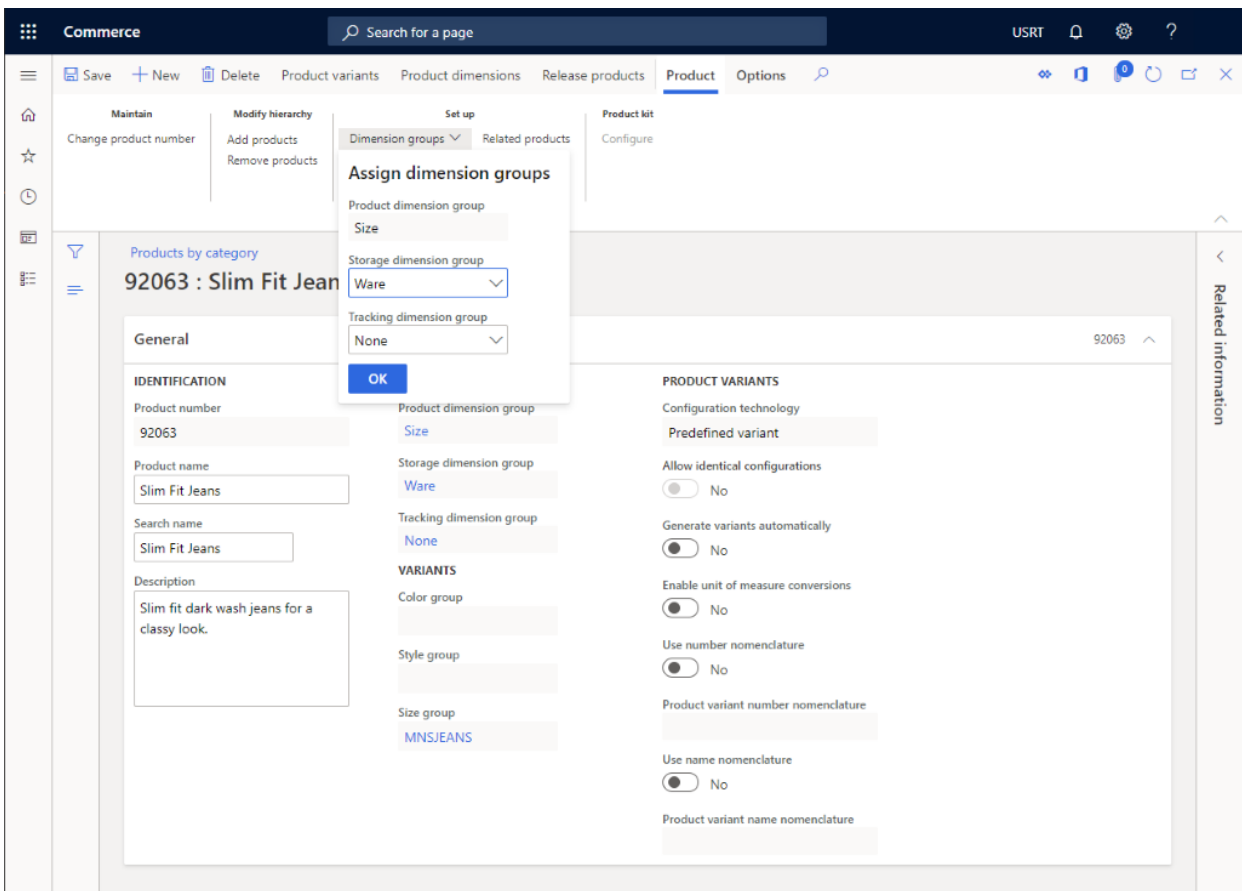
1. In the navigation pane, go to **Modules > Retail and commerce > Products and categories > Products by category**.
2. On the action pane, select **New**.
3. In the **Product type** drop-down list, select either **Item** or **Service**.
4. In the **Product subtype** drop-down list, select either **Product** (if the product will have no variants) or **Product master** (if the product will have variants).
5. In the **Product number** box, enter a product number if one is not already prepopulated.
6. In the **Product name** box, enter a product name.
7. In the **Search name** box, enter a search name.
8. In the **Retail category** drop-down list, select an appropriate category.
9. If the product is a kit, select **Yes** for **Product kit**.
10. If the product subtype is product master, set the **Product dimension group** to include the supported variants. Options include **Color**, **Size**, **Style**, and **Configuration**. You may need to create additional product dimension groups if needed.
11. In the **Configuration technology** drop-down list, select an appropriate option.
12. Select **OK**.

The following image shows an example product being added.

Product name	Product number	Search name
Dark wash straight Leg Jeans	92000	Straight Leg Jeans
Boot-Leg Vintage Wash Jeans	92064	Boot-Leg Jeans
Slim Fit Jeans	92063	Slim Fit Jeans
Relaxed Fit Straight-leg Jeans	92062	Straight-leg Jeans
Wrinkle-free Button-Down Shirt	92061	Button-Down Shirt
Easy-Fit Button-Down Shirt	92060	Button-Down Shirt
Casual Fit Button-Down Shirt	92059	Button-Down Shirt
Relaxed Fit Button-Down Shirt	92058	Button-Down Shirt
Slim Fit Button-Down Shirt	92057	Button-Down Shirt
Denim Button-Down Shirt	92056	Button-Down Shirt
Casual Drape Cardigan	92055	Casual Cardigan
Charcoal Trim Cardigan	92054	Charcoal Cardigan
Varsity Cardigan	92053	Varsity Cardigan
Blue Denim Jacket	92052	Blue Denim Jacket
All Season Windbreaker	92051	Windbreaker
Quick-Dry Performance Jacket	92050	Quick-Dry Jacket
Black Down Vest	92049	Black Down Vest

Once a product is added, additional data can be set for it, such as **Product description**, **Variant groups**, **Dimension groups**, **Product attributes**, and **Related products**.

The following image shows a product's additional details.



## Create product variants

If the product subtype is **Product master**, specific variants will need to be created.

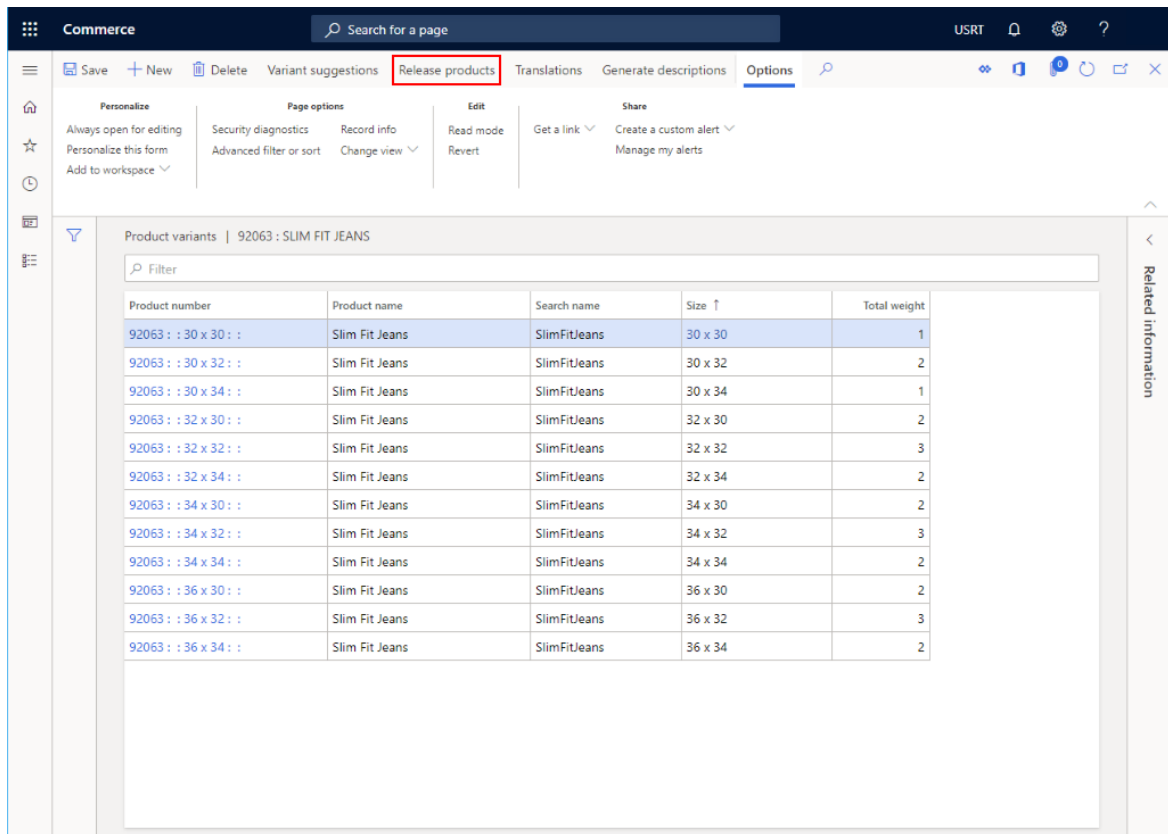
To create product variants, follow these steps.

1. On the action pane, select **Product variants**.
2. If variant groups have been selected on the action pane, select *\* Variant suggestions*.
3. Select the variants you would like to support for the product.
4. Select **Create**.

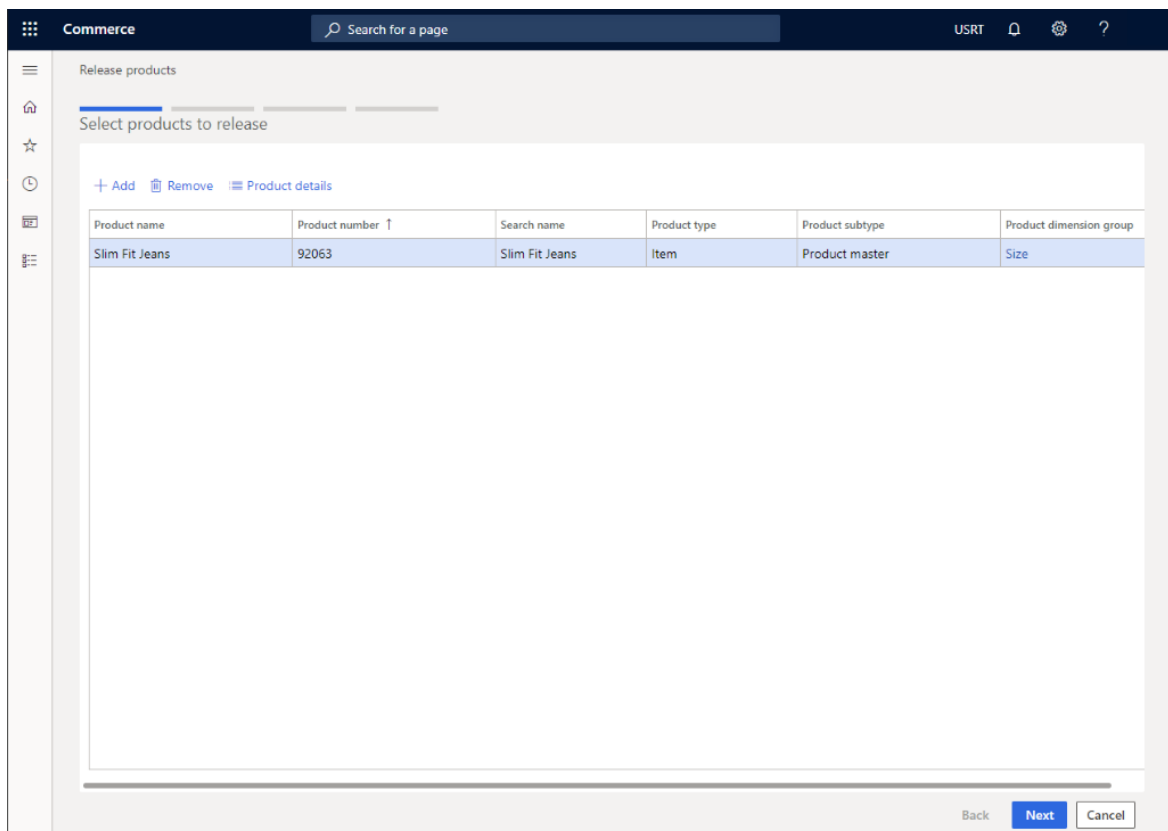
## Release a product

To sell a product it must first be released to a legal entity.

1. From the product page, select **Release products**.

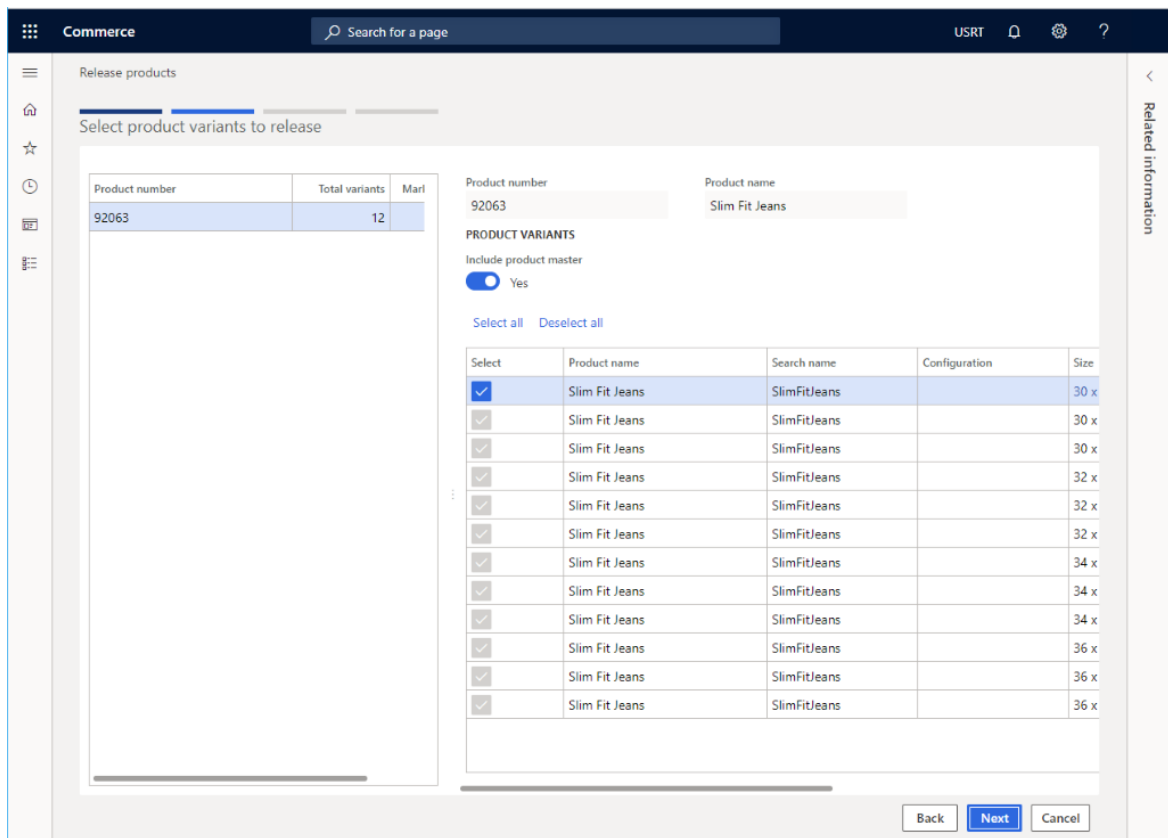


2. Select the product to release, and then select **Next**.

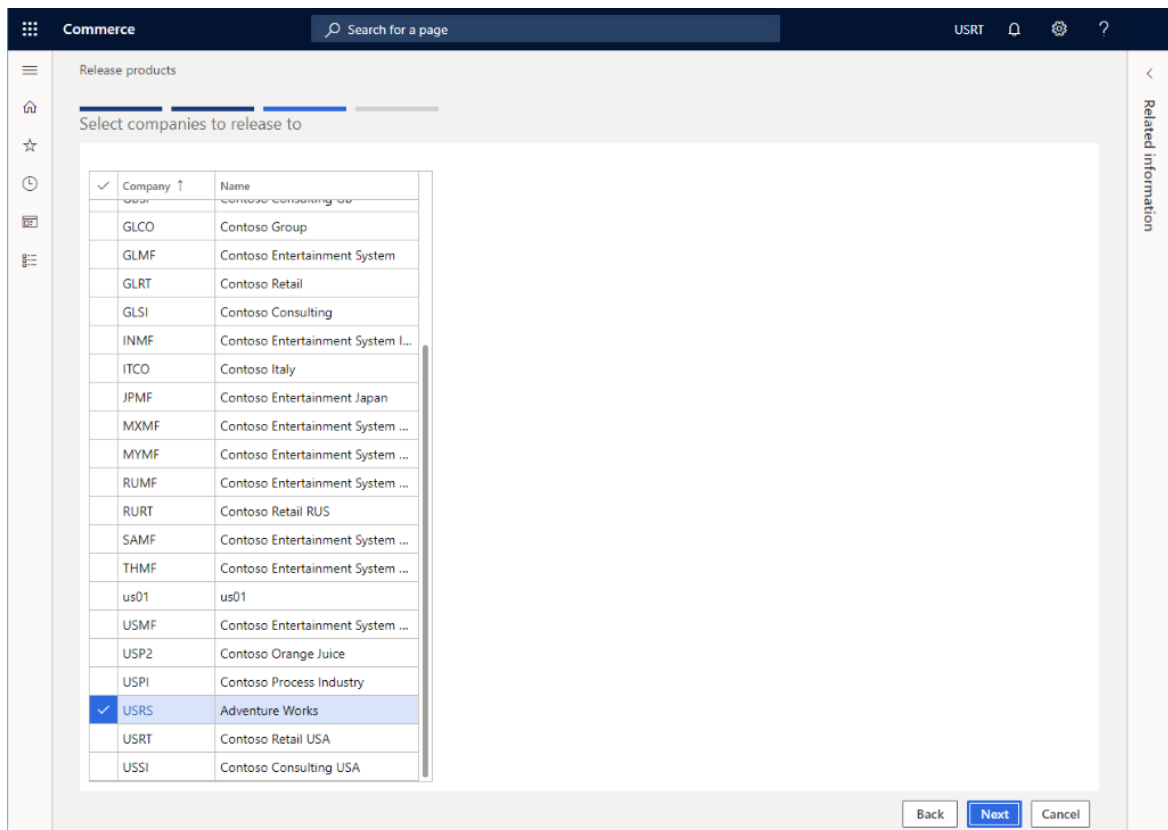


3. Select the set of product variants to release, and then select **Next**.

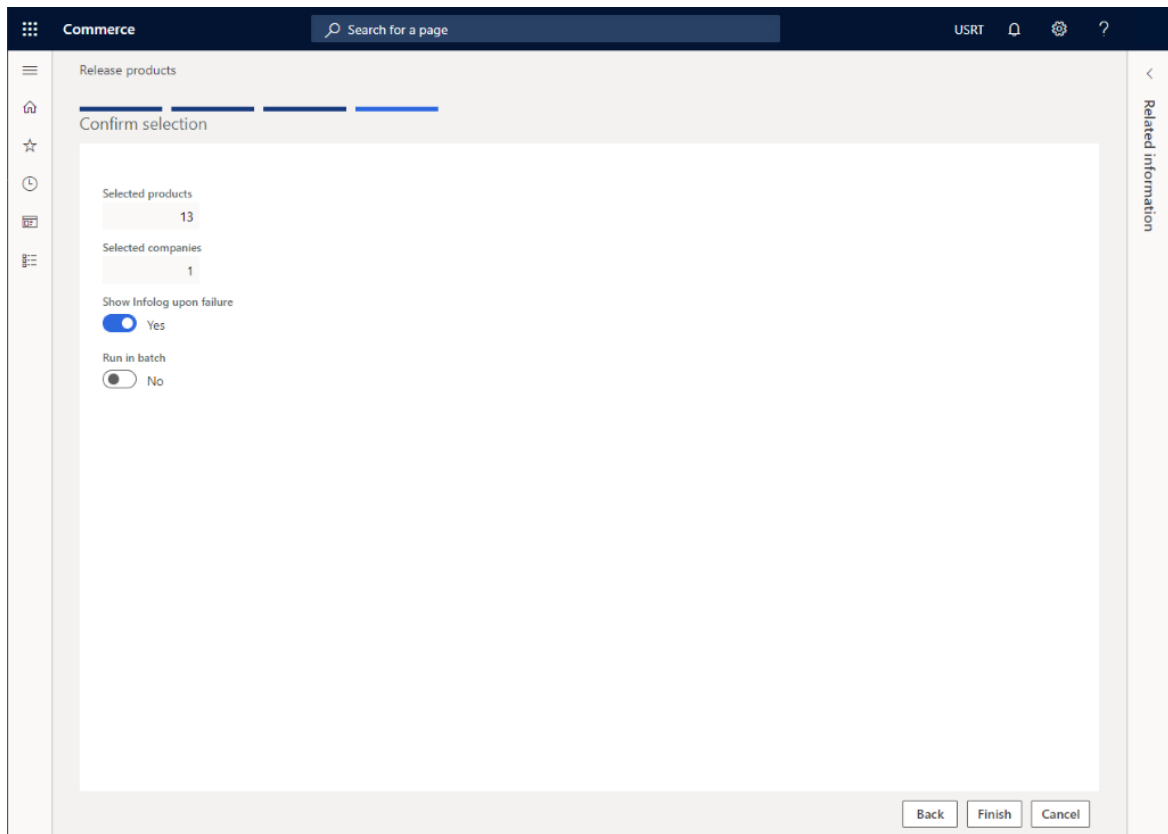




4. Select the legal entity, and then select **Next**.



5. Select **Finish**.



## Configure a released product

Once a product is released, it will then require further configuration that includes adding a price to the product.

1. In the navigation pane, go to **Modules > Retail and commerce > Products and categories > Released products by category**.
2. Select the product category node for the product that was released, and then select the product from the product list.
3. On the action pane, select **Edit**.
4. In the **Purchase** section, configure any required properties including **Unit**, **Price**, and **Quantity**.
5. On the action pane, select **Validate** to ensure that no errors are reported for missing fields.
6. On the action pane, select **Save**.

The following image shows an example configuration for a released product.

Commerce Search for a page USRT

[Edit](#) [New](#) [Delete](#) [Product](#) [Purchase](#) [Sell](#) [Manage inventory](#) [Plan](#) [Retail](#) [Setup](#) [Options](#)

[Maintain](#) [Modify hierarchy](#) [New](#) [Product master](#) [Languages](#) [Set up](#) [Product kit](#)  
[Validate](#) [Add products](#) [Product](#) [Released product variants](#) [Translations](#) [Dimension groups](#) [Product categories](#) [Configure](#)  
[Remove products](#) [Product dimensions](#) [Counting reason code policy](#) [Related products](#) [Unit conversions](#)

Released product details  
92063 : Slim Fit Jeans

General 92063

<b>IDENTIFICATION</b>	Item number 92063	Search name Slim Fit Jeans	<b>ADMINISTRATION</b>	Counting reason code policy
Product type	Item	Description Slim fit dark wash jeans for a classy look.	Storage dimension group Ware	Product lifecycle state
Product subtype	Product master		Tracking dimension group None	
	<b>FURTHER IDENTIFICATION</b>		Item model group MOV_AVG	
	Product name Slim Fit Jeans			

Purchase ea | RP | 60.00

<b>PURCHASE ORDER</b>	Underdelivery Unit ea 0.00	<b>ADMINISTRATION</b> Vendor	<b>PRICE UPDATE</b> Latest purchase price No	<b>PRICES</b> Price 60.00 Price quantity 1.00
Overdelivery 0.00	Intercompany stopped No	<b>TAXATION</b> Item sales tax group RP	Date of price 5/27/2019	

Sell ea | RP | 220.00

<b>SALES ORDER</b>	<b>TAXATION</b> Item sales tax group RP	<b>PRICE UPDATE</b> Sales price model None	<b>BASE SALES PRICE</b> Price 220.00 Price quantity 1.00	<b>CONTINUITY</b> Continuity schedule ID
Unit ea Overdelivery 0.00 Underdelivery 0.00		Base price Purchase price Contribution ratio 0.00 Date of price 5/27/2019	<b>PRICE ADJUST</b> Allow price adjust No	Event duration 0 <b>SELL DATES</b> Sell start date Sell end date

Manage inventory ea

## Additional resources

[Create legal entities](#)

[Create a variant group](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage product categories and products

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes an enhanced way to manage product categories and products in Dynamics 365 Commerce. The enhancements let merchandising managers view a structure of product properties that is shared between the product hierarchy and released product details.

To learn more about how to manage product categories, in the **Category and product management** workspace, select the **Commerce product hierarchy** tile.

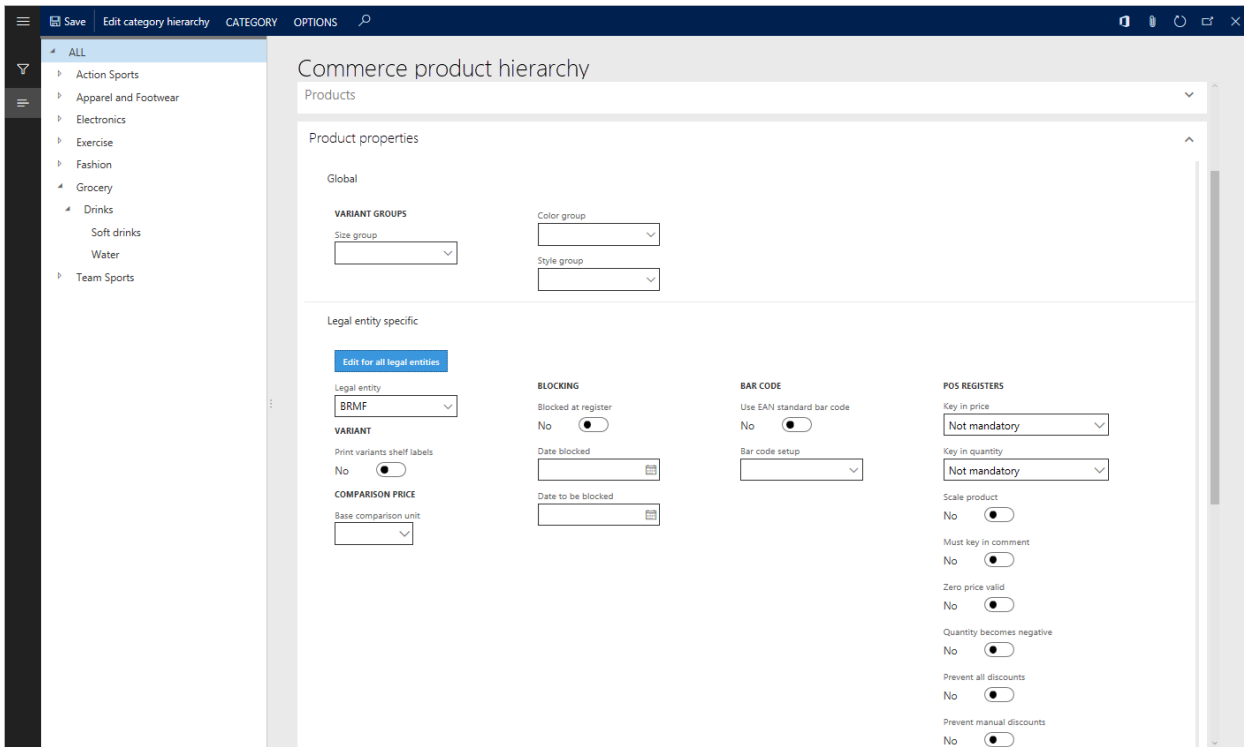
Notice the enhanced structure of the **Commerce product hierarchy** page that appears. In previous versions of the app, product properties were divided into *basic product properties* and *Retail product properties*, based on the scope of their applicability. Retail product properties are *global* in their scope of applicability. In other words, for a given product property, the same value is shared across all legal entities. By contrast, basic product properties are *legal entity–specific*. In other words, for a given basic product property, the value can differ across legal entities, depending on the individual business requirements of each legal entity.

In the enhanced product category structure, product properties are logically separated based on their applicability in a group, to reflect the structure of the released product details form structure.

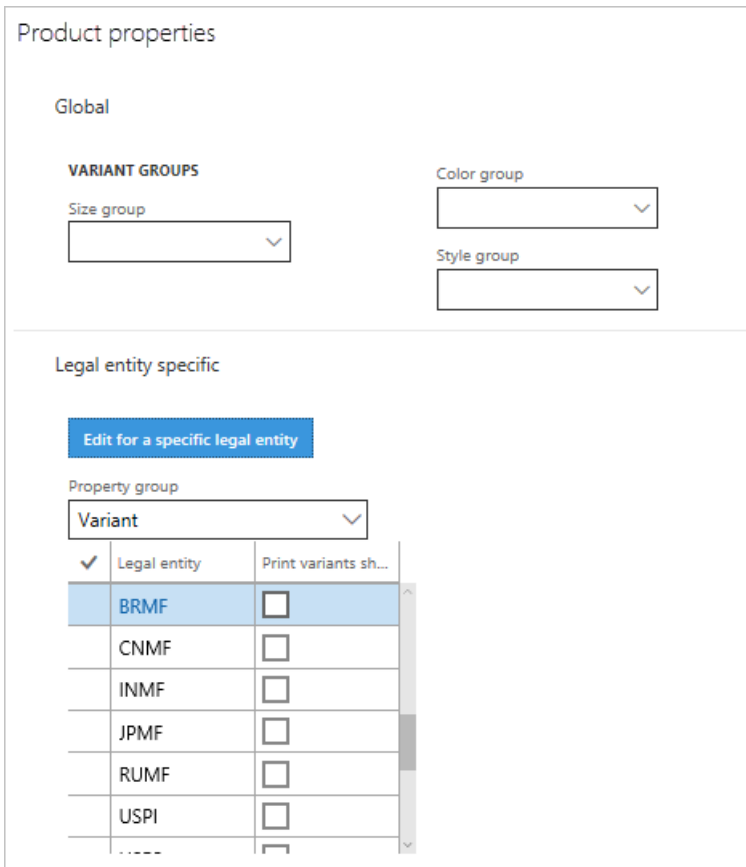
The screenshot displays the Dynamics 365 Commerce interface for managing product categories. The top navigation bar shows 'Dynamics 365' and 'Commerce'. The main content area is titled 'Commerce product hierarchy' and is divided into two main sections: 'Product properties' and 'Legal entity specific'. The 'Product properties' section includes 'Global' properties such as 'VARIANT GROUPS' (Size group, Color group, Style group) and 'Legal entity specific' properties (Legal entity, BAR CODE, POS REGISTERS, VARIANT, COMPARISON PRICE, BLOCKING). The 'Legal entity specific' section has a toggle for 'Edit for all legal entities' and a dropdown for 'Legal entity' set to 'BRMF'.

You can switch between managing legal entity–specific properties across all legal entities and managing them for a specific legal entity.

To manage properties across all legal entities, select **View for all legal entities** (or **Edit for all legal entities**).



To manage properties for a specific legal entity, select **View** for a specific legal entity (or **Edit** for a specific legal entity).

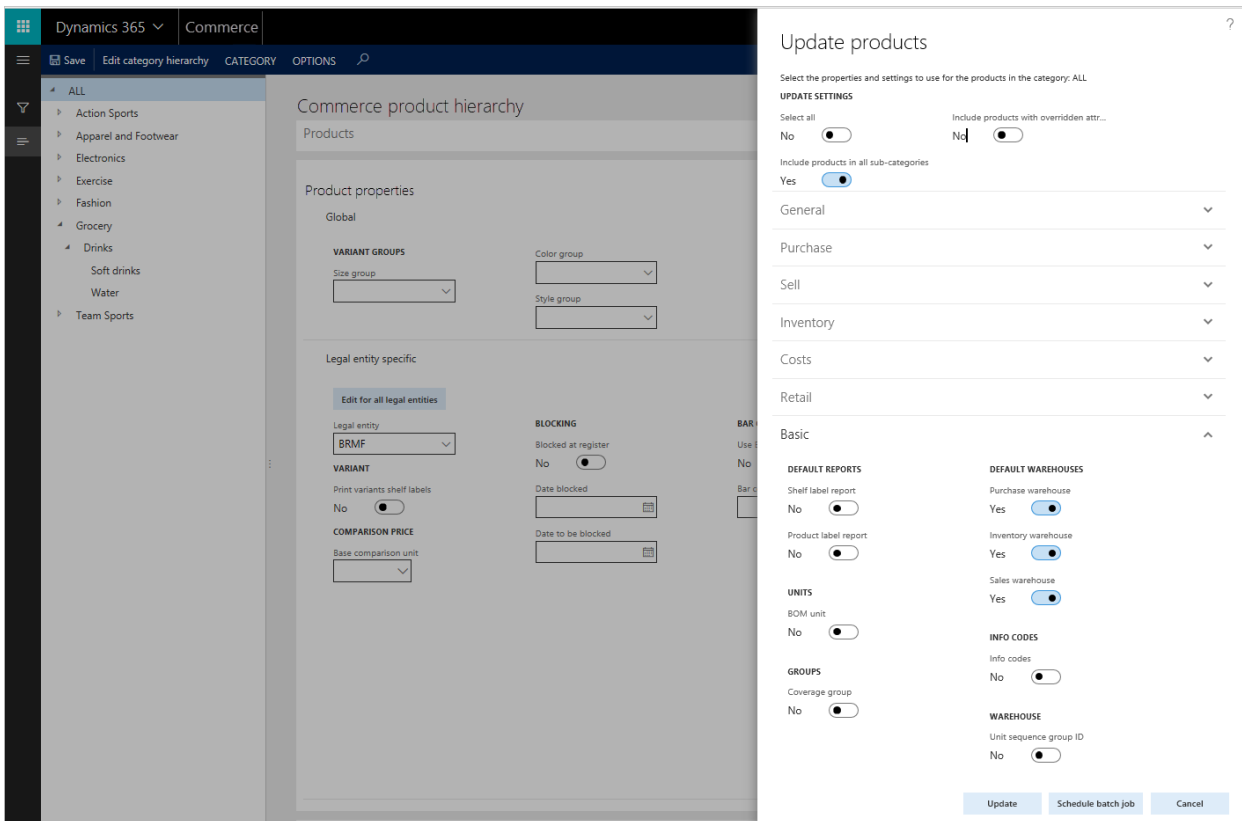


Additionally, in the enhanced product category structure, a merchandising manager can now define default values for an additional set of product properties at the level of the individual category. Then, when products are created, they inherit default values for their product properties, based on the association of those properties with an individual category in the product hierarchy. These inherited product properties can also be modified for each product to meet individual business requirements.

Selecting properties to update products on the Commerce product

# hierarchy page

You can use the new enhanced structure for product properties to select updated product properties that must be pushed to the associated products. On the **Commerce product hierarchy** page, on the Action Pane, select **Category**, and then select **Update products** to open the **Update products** dialog box.



## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage attributes and attribute groups

2/18/2021 • 10 minutes to read • [Edit Online](#)

*Attributes* provide a way to further describe a product and its characteristics through user-defined fields (such as **Memory size**, **Hard disk capacity**, **Is Energy star compliant**, and so on). Attributes can be associated with various Commerce entities, such as product categories and channels, and default values can be set for them. Products then inherit the attributes and the default values when they are associated with the product categories or channels. The default values can be overridden at the individual product level, at the channel level, or in a catalog.

For example, a typical television product might have the following attributes.

CATEGORY	ATTRIBUTE	PERMISSIBLE VALUES	DEFAULT VALUE
TV & Video	Brand	Any valid brand value	None
TV	Screen Size	20–80 inches	None
	Vertical Resolution	480i, 720p, 1080i, or 1080p	1080p
	Screen Refresh Rate	60hz, 120hz, or 240hz	60hz
	HDMI Inputs	0–10	3
	DVI Inputs	0–10	1
	Composite Inputs	0–10	2
	Component Inputs	0–10	1
LCD	3D Ready	Yes or No	Yes
	3D Enabled	Yes or No	No
Plasma	Operating Temp From	32–110 degrees	32
	Operating Temp To	32–110 degrees	100
Projection	Projection Tube Warranty	6, 12, or 18 months	12
	# of Projection Tubes	1–5	3

## Attributes and attribute types

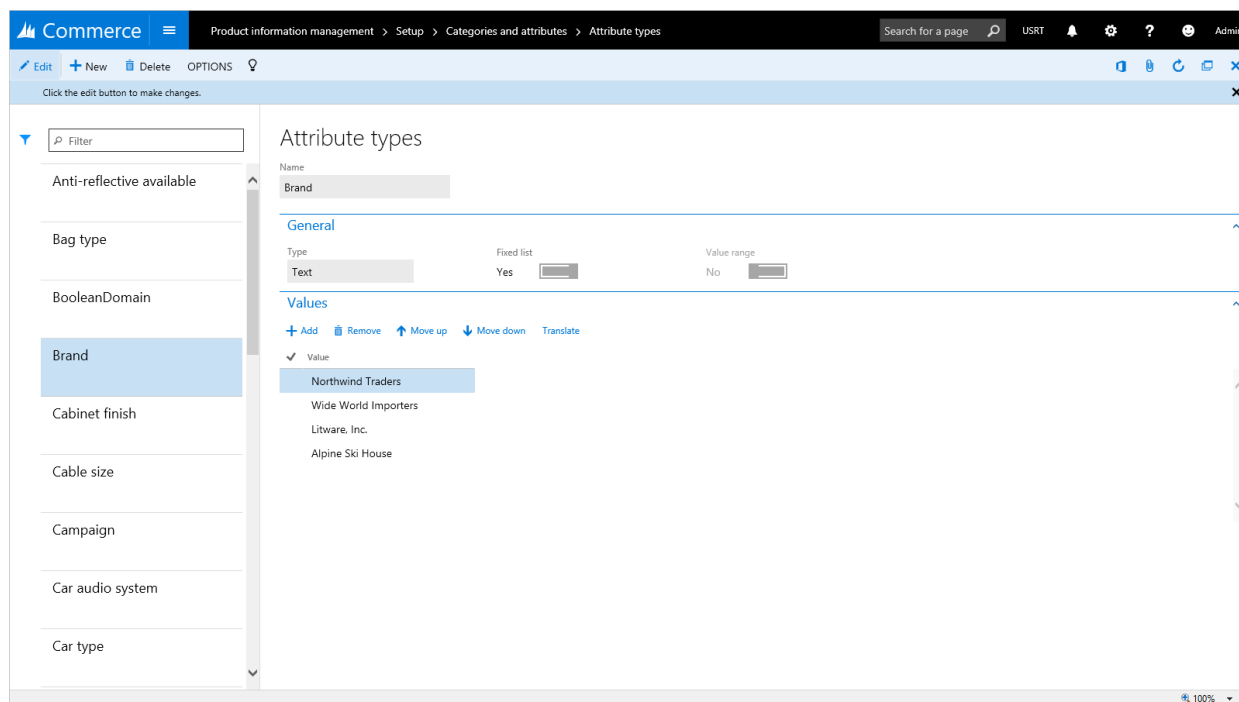
Attributes are based on *attribute types*. The attribute type identifies the type of data that can be entered for a specific attribute. The following attribute types are supported:

- **Currency** – This type supports a currency value. It can be bounded (that is, it can support a range of values), or it can be left open.
- **DateTime** – This type supports a date and time value. It can be bounded or left open.

- **Decimal** – This type supports a numerical value that includes decimal places. It also supports a unit of measure. It can be bounded or left open.
- **Integer** – This type supports a numerical value. It also supports a unit of measure. It can be bounded or left open.
- **Text** – This type supports a text value. It also supports a predefined set of possible values (that is, an *enumeration*).
- **Boolean** – This type supports a binary value (**true** or **false**).
- **Reference** – This type references other attributes.

## Set up attribute types

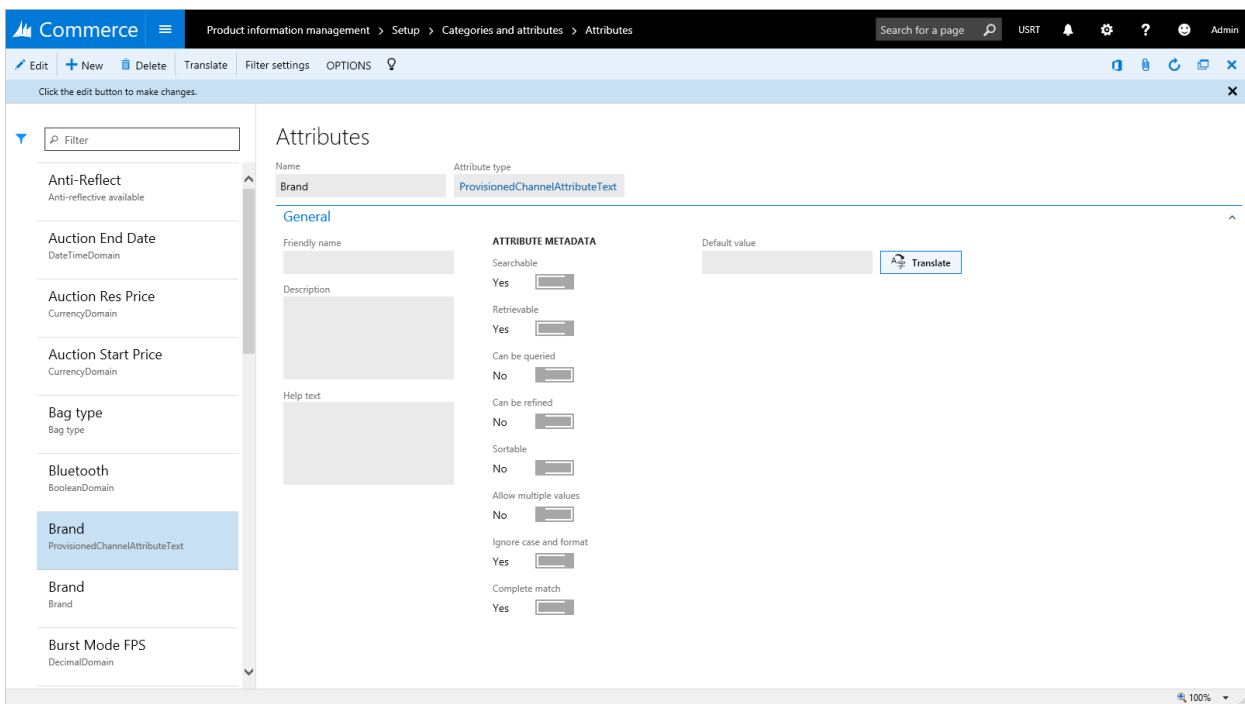
1. Sign in to the back-office client as a merchandising manager.
2. Go to **Product information management > Setup > Categories and attributes > Attribute types**.
3. Create two attribute types of the **Text** type, set the **Fixed list** option to **Yes**, and then add a list of values:
  - Name one attribute type **Lens shape**, and add the following values: **Oval**, **Square**, and **Rectangle**.
  - Name the other attribute type **Sunglass brand**, and add the following values: **Ray ban**, **Aviator**, and **Oakley**.



## Set up an attribute

1. Sign in to the back-office client as a merchandising manager.
2. Go to **Product information management > Setup > Categories and attributes > Attributes**.
3. Create an attribute that is named **Lens**.
4. Set the **Attribute type** field to **Lens shape**.





## Attribute metadata

*Attribute metadata* lets you select options to specify how the attributes for each product should behave. For example, you can specify whether attributes are required, whether they can be used for searches, and whether they can be used as a filter.

For products, the attribute metadata settings can be overridden at the channel level. This capability will be discussed later in this topic.

As you might notice, the **Attributes** page includes options that are related to attribute metadata. Under **Attribute metadata** for POS, one option that is named **Can be refined** affects the behavior of the attribute values in the point of sale (POS) or the way that the system handles those attribute values. Only attributes for which you may set the **Can be refined** option to **Yes**, will show up for refinement or filtering of products in the POS.

Here are the remaining attribute metadata options on the **Attributes** page:

- Searchable
- Retrievable
- Can be queried
- Sortable
- Allow multiple values
- Ignore case and format
- Complete match

These options were originally intended to improve the search functionality for the online storefront. Although Commerce doesn't include the online storefront out of the box, it does include the eCommerce Publishing Software Development Kit (SDK). Customers can use this SDK to put products into a search index of their choice. Although the product data is imported, customers should still be able to distinguish searchable data, data that can be queried, and so on. In that way, they can build an optimal index to make sure that they index only attributes that, *in their opinion*, should be indexed.

For information about the purpose of these remaining options, see [Overview of the search schema in SharePoint Server 2013](#).

# Filter settings for attributes

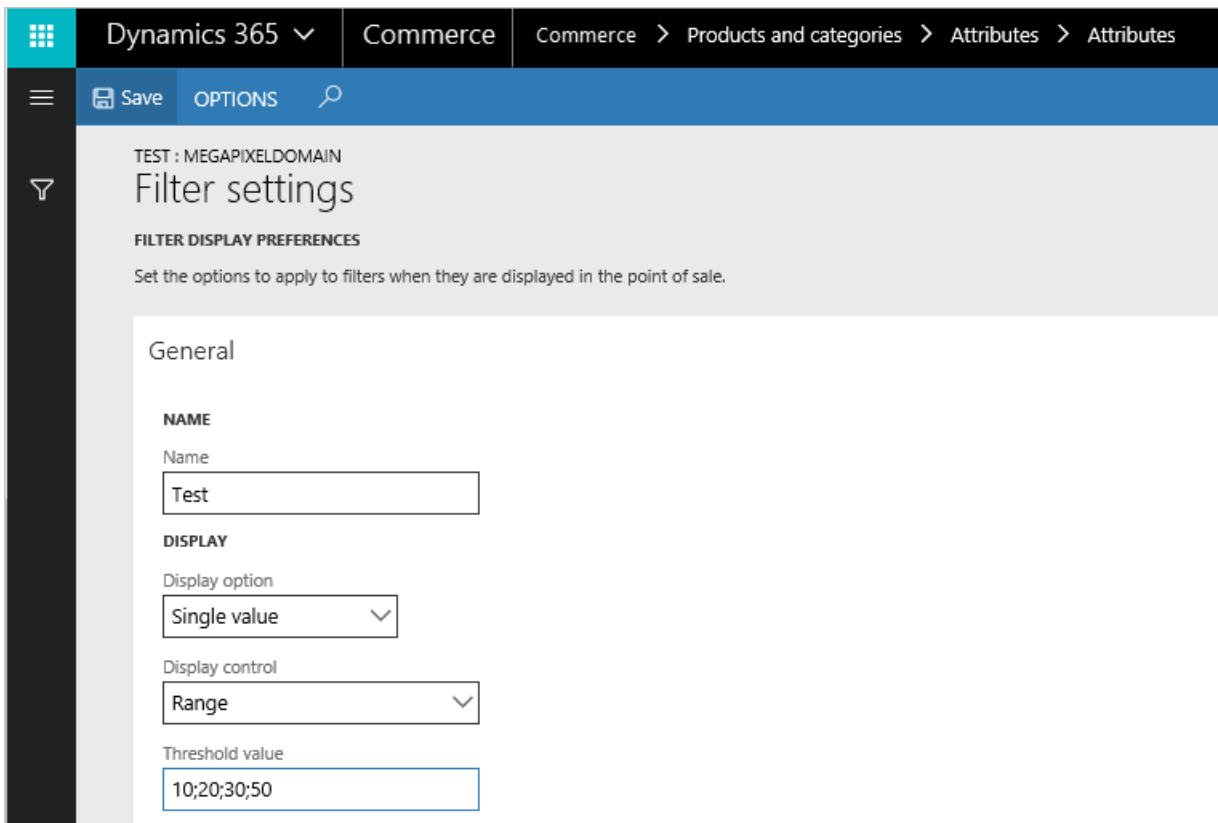
Filter settings for attributes let you define how the filters for attributes are shown in the POS. To access the filter settings for an attribute, on the **Attributes** page, select the attribute, and then, on the Action Pane, select **Filter settings**.

The **Filter display preferences** page includes the following fields:

- **Name** – By default, this field is set to the name of the attribute. However, you can change the value.
- **Display option** – The following options are available:
  - **Single value** – This option is available for the following attribute types: **Boolean, Currency, Decimal, Integer, and Text**. This option enables single value selection for these attributes in the client for refinement.
  - **Multi value** – This option is available for the following attribute types: **Currency, Decimal, Integer, and Text**. This option enables multi-value selection for this attribute in the client for refinement.
- **Display control** – The following options are available:
  - **List** – This option is available for the all attribute types.
  - **Range** – This option is available for the following attribute types: **Currency, Decimal, and Integer**.
  - **Slider** – This option is available for the following attribute types: **Currency, Decimal, and Integer**.
  - **Slider with bars** – This option is available for the following attribute types: **Currency, Decimal, and Integer**.
- **Threshold value** – This setting is required if you selected **Range** as the display control type. You can define values by using a semicolon (;) as a delimiter.

For example, for the filter like **Bag Volume**, a threshold value can be **10; 20; 50; 100; 200; 500; 1000; 5000**. In this case, the POS will show the following ranges. Any ranges that don't have any products in the result set will appear dimmed.

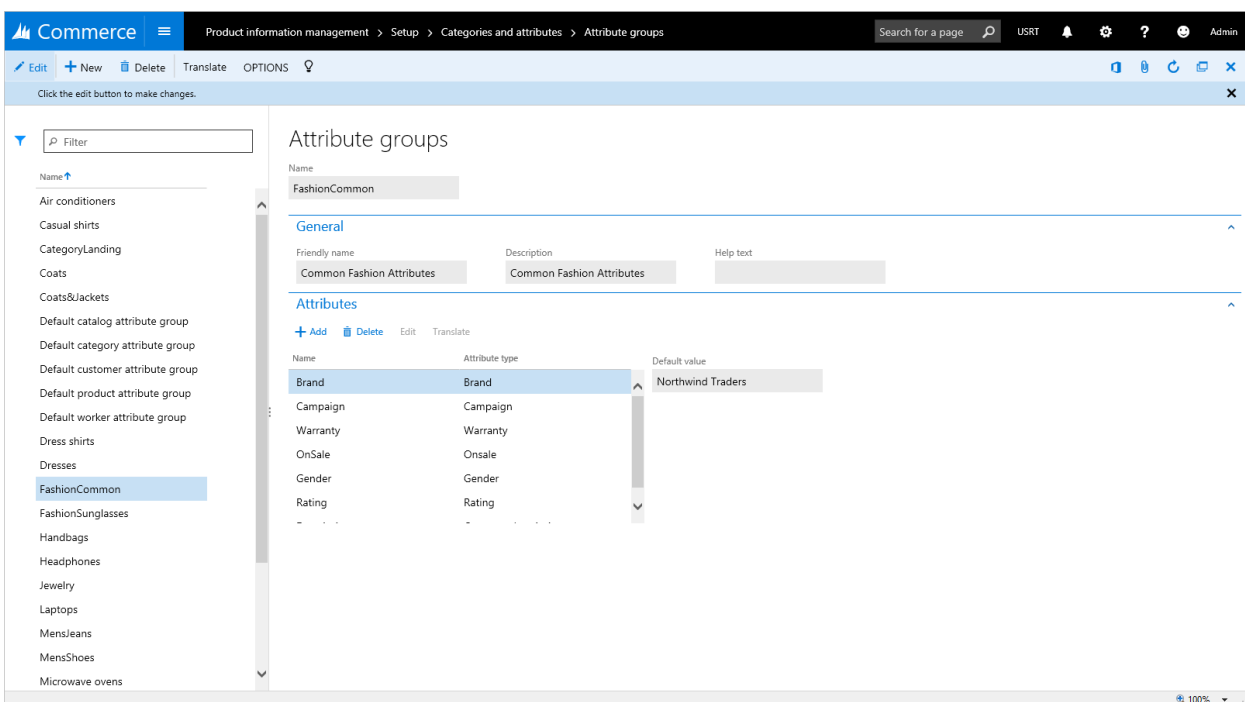
- Less than 10
- 10 – 20
- 20 – 50
- 50 – 100
- 100 – 200
- 200 – 500
- 500 or more



## Attribute groups

After attributes have been defined, they can be assigned to attribute groups. An *attribute group* is used to group the individual attributes for a component or subcomponent in a product configuration model. An attribute can be included in more than one attribute group. Attribute groups can help users configure products, because the various selections are arranged in a specific context. Attribute groups can be assigned to categories or channels.

You can also set default values for attributes that are included in an attribute group. For example, you add an attribute for color to an attribute group and select **Blue** as the default attribute value. In this case, when the attribute group is added to a product that includes color as one of its attributes, **Blue** appears as the default color for that product.

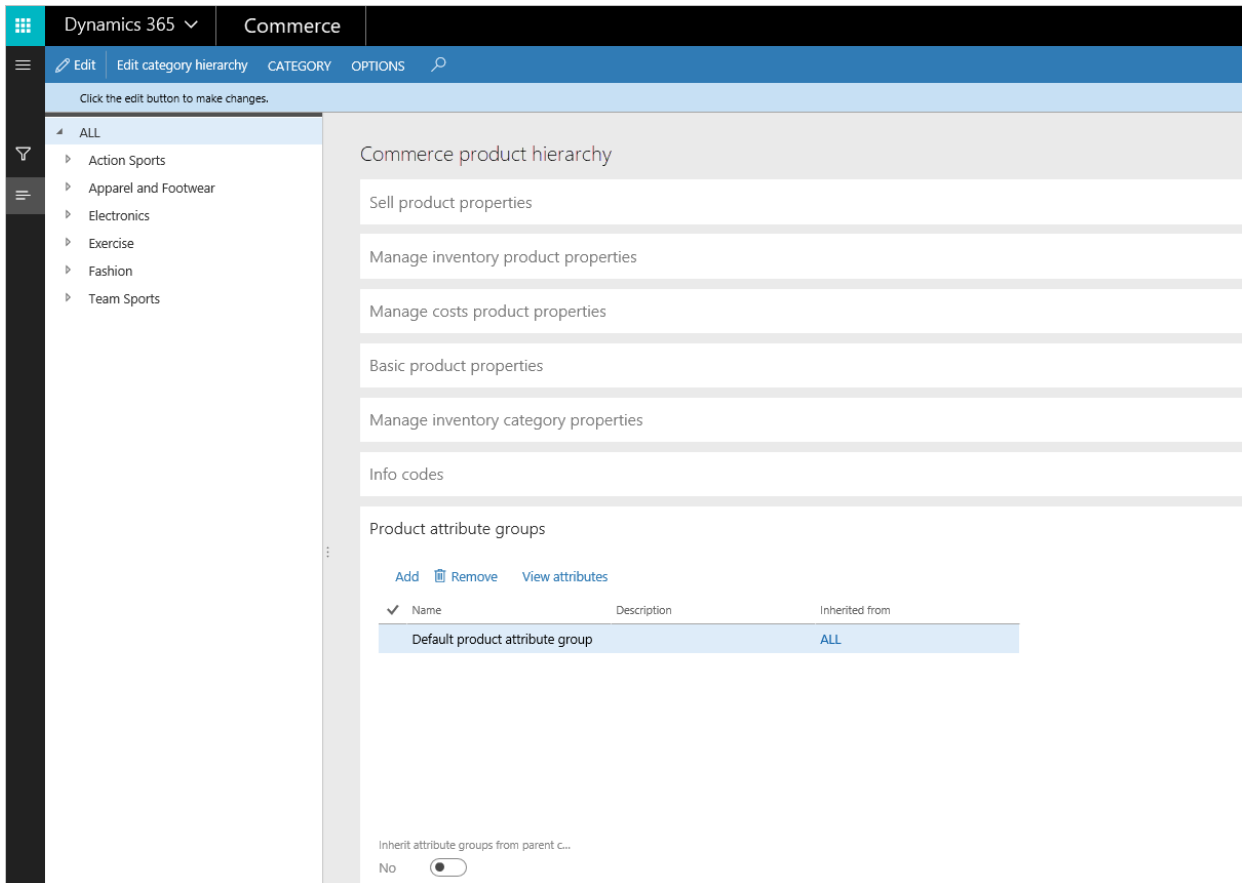


### Create an attribute group

1. Sign in to the back-office client as a merchandising manager.
2. Go to **Product information management > Setup > Categories and attributes > Attribute groups**.
3. Create an attribute group that is named **Fashion Sunglasses**.
4. Add the following attributes: **Lens shape** and **Sunglass brand**.

### Assign attribute groups to categories

One or more attribute groups can be associated with category nodes in the following types of category hierarchies: Commerce product hierarchy, Channel navigation category hierarchy, and Supplemental product category hierarchy. Then, when products are categorized, they inherit the attributes that are included in the attribute groups.



Follow these steps to assign attribute groups to categories in the Commerce product hierarchy.

1. Sign in to the back-office client as a merchandising manager.
2. Go to **Retail and Commerce > Category and product management > Commerce product hierarchy**.
3. Select **Fashion navigation hierarchy**.
4. Under **Menswear**, select the **Pants** category, and then, on the **Product attribute groups** FastTab, add an attribute group that is named **Men's belt**.
5. Select the **Fashion sunglasses** category, and verify the new attributes in the **Fashion Sunglasses** attribute group by selecting **View attributes**.

The attribute group should show the new **Lens shape** and **Sunglass brand** attributes.

6. Under **Menswear**, select the **Pants** category, and verify the attributes for the **Men's belt** attribute group by selecting **View attributes**.

The attribute group should show the **Men's belt brand**, **Belt fabric**, and **Belt size** attributes.

#### NOTE

This procedure can also be used to assign attribute groups to categories in the Channel navigation category hierarchy and the Supplemental product category hierarchy. In step 2, use the following navigation paths:

- Retail and Commerce > Category and product management > Channel navigation categories
- Retail and Commerce > Category and product management > Supplemental product categories

### Assign attribute groups to stores

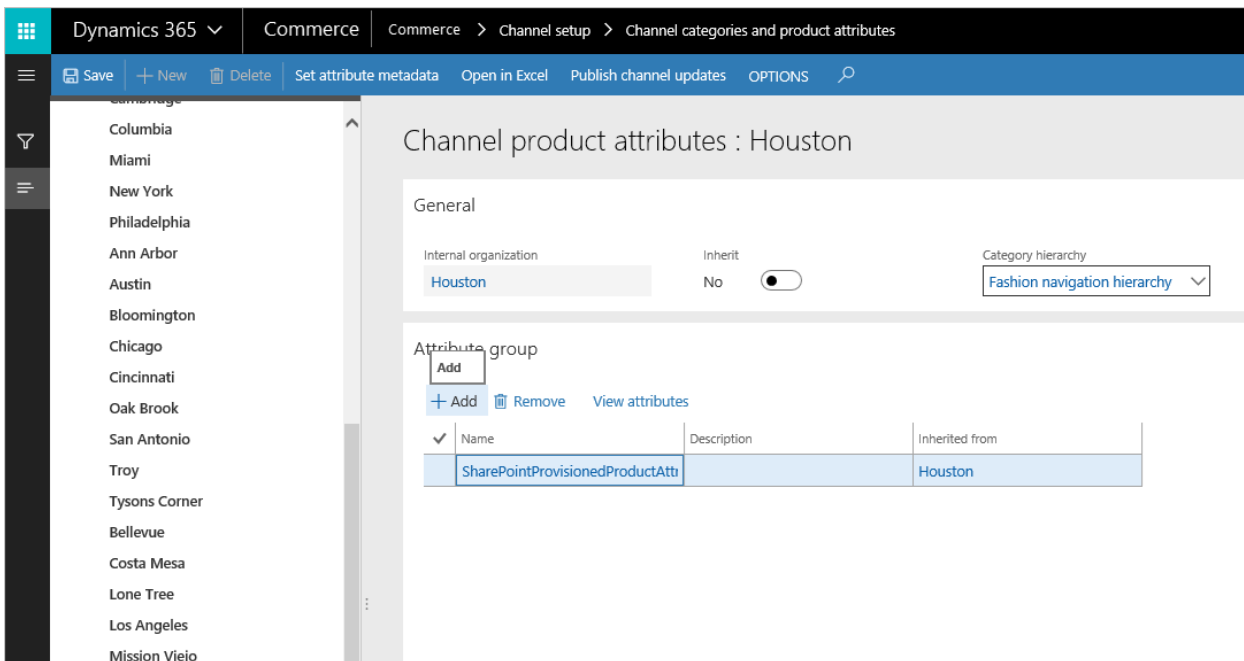
One or more attribute groups can be associated with one or more stores in the store hierarchy. Then, when products are enriched for specific stores, they inherit the attributes that are included in the attribute groups.

1. Sign in to the back-office client as a merchandising manager.
2. Go to **Retail and Commerce > Channel setup > Channel categories and product attributes**.
3. Assign attribute groups to the Houston channel:
  - a. Select the **Houston** channel.
  - b. On the **Attribute group** FastTab, select **Add**, and then, in the **Name** field, select **SharePointProvisionedProductAttributeGroup**.
  - c. Select **Add** again, and then, in the **Name** field, select **Men's belt**.
  - d. Select **Add** again, and then, in the **Name** field, select **Fashion Sunglasses**.

#### NOTE

An option lets you specify that this channel should inherit the attribute groups from its parent channel in the hierarchy. If you set the **Inherit** option to **Yes**, the child channel node inherits all the attribute groups and all the attributes in those attribute groups.

4. Enable the attributes so that they are available in the Houston channel:
  - a. On the Action Pane, select **Set attribute metadata**.
  - b. Select the **Fashion** category node, and then, on the **Channel product attributes** FastTab, select **Include attribute** for each attribute.
  - c. Select the **Fashion Accessories** category node, select the **Fashion Sunglasses** category, and then, on the **Channel product attributes** FastTab, select **Include attribute** for each attribute.
  - d. Select the **Menswear** category node, select the **Pants** category, and then, on the **Channel product attributes** FastTab, select **Include attribute** for each attribute.

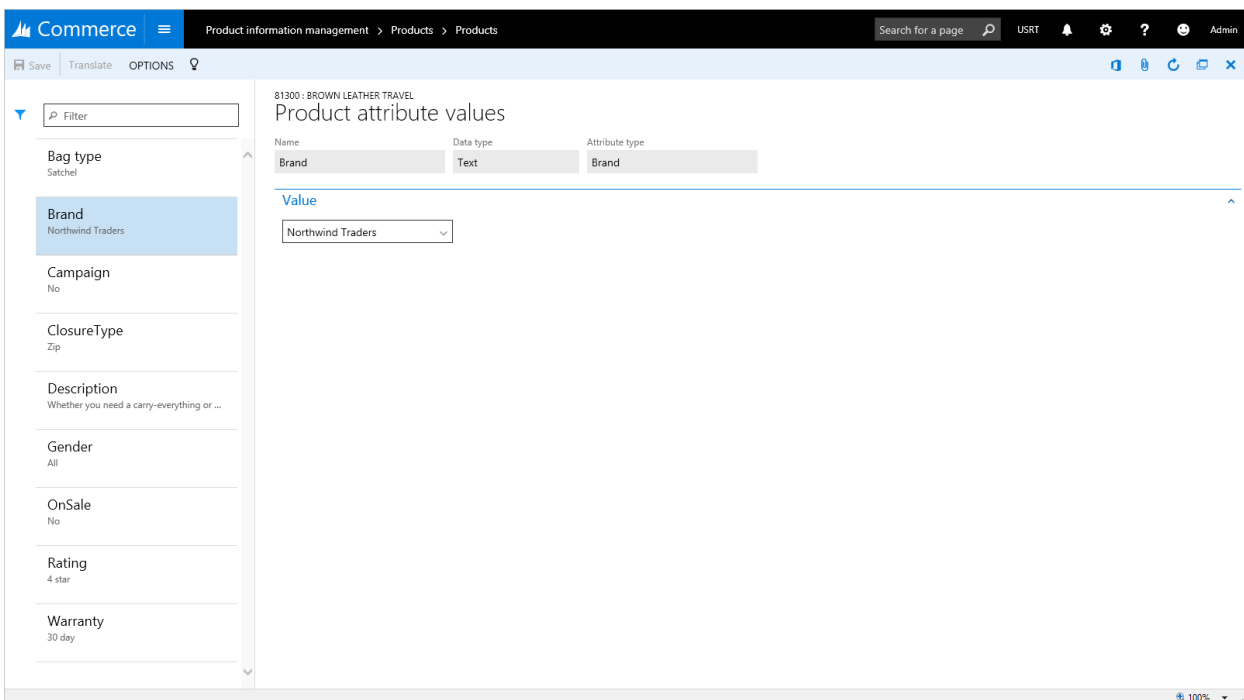


## Overriding attribute values

The default values of attributes can be overridden for individual products at the product level. Default values can also be overridden for individual products in specific catalogs that are targeted at specific channels.

### Override the attribute values of an individual product

1. Sign in to the back-office client as a merchandising manager.
2. Go to **Retail and Commerce > Category and product management > Released products by category**.
3. Select the **Fashion > Fashion Accessories > Fashion Sunglasses** category node.
4. Select the required product in the grid. Then, on the Action Pane, on the **Product** tab, in the **Set up** group, select **Product attributes**.
5. Select an attribute in the left pane, and then update its value in the right pane.



### Override the attribute values of products in a catalog

1. Sign in to the back-office client as a merchandising manager.

2. Go to **Retail and Commerce > Catalog management > All catalogs**.
3. Select the **Fabrikam Base Catalog** catalog.
4. Select the **Fashion > Fashion Accessories > Fashion Sunglasses** category node.
5. On the **Products** FastTab, select the required product, and then select **Attributes** above the product grid.
6. On the following FastTabs, update the values of the required attributes:
  - Shared product media
  - Shared product attributes
  - Channel media
  - Channel product attributes

#### NOTE

If shared product media and shared product attributes are created, they apply to all the products.

### Override the attribute values of products in a channel

1. Sign in to the back-office client as a merchandising manager.
2. Go to **Retail and Commerce > Channel setup > Channel categories and product attributes**.
3. Select the **Houston** channel.
4. On the **Products** FastTab, select the required product, and then select **Attributes** above the product grid.

#### NOTE

If no products are available, add products by selecting **Add** on the **Products** FastTab and then selecting the required products in the **Add products** dialog box.

5. On the following FastTabs, update the values of the required attributes:
  - Shared product media
  - Shared product attributes

- Channel media
- Channel product attributes

**NOTE**

If shared product media and shared product attributes are created, they apply to all the products.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Define channel attributes

2/18/2021 • 2 minutes to read • [Edit Online](#)

Enrich the channel by adding channel and product attributes. You must add the channel to the organization default hierarchy before you can define the attributes.

## Define channel product attribute.

1. Go to Retail and Commerce > Channel setup > Channel categories and product attributes.
2. In the tree, select 'Contoso Retail\Fashion\Houston'.
3. Click Edit.
4. In the Category hierarchy field, enter or select a value.
  - The navigation hierarchy will be used for navigating the product inside your channels.
5. Expand the Attribute group section.
6. Click Add.
  - The attribute group will be shown in the product details and also will help when using the product filter.
7. In the Name field, enter or select a value.
8. Click View attributes.
9. Expand the Attributes section.
10. Click Add.
11. In the list, mark the selected row.
12. Click AddBtn.
13. Click OK.
14. Click Save.
15. Close the page.
16. Expand the Products section.
17. Click Add.
18. In the list, mark the selected row.
19. Click Add.
  - Similarly you can add multiple products.
20. Click OK.
21. Click Attributes.
22. Expand the Channel media section.
23. Click Edit.
24. In the Channel field, select an option.
25. Select the Default check box.
26. Click OK.
27. Expand the Channel product attributes section.
28. In the Brand field, type a value.
  - Similarly you can set the other attributes.
29. Click Save.
30. Close the page.
31. Click Save.

32. Click Publish channel updates.

- After publish the data will be ready for sync with the channel tables. If you didn't publish the modified data will not be synced with the channel tables.

33. Click OK.

34. Click Save.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a channel navigation hierarchy

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a channel navigation hierarchy in Microsoft Dynamics 365 Commerce.

## Overview

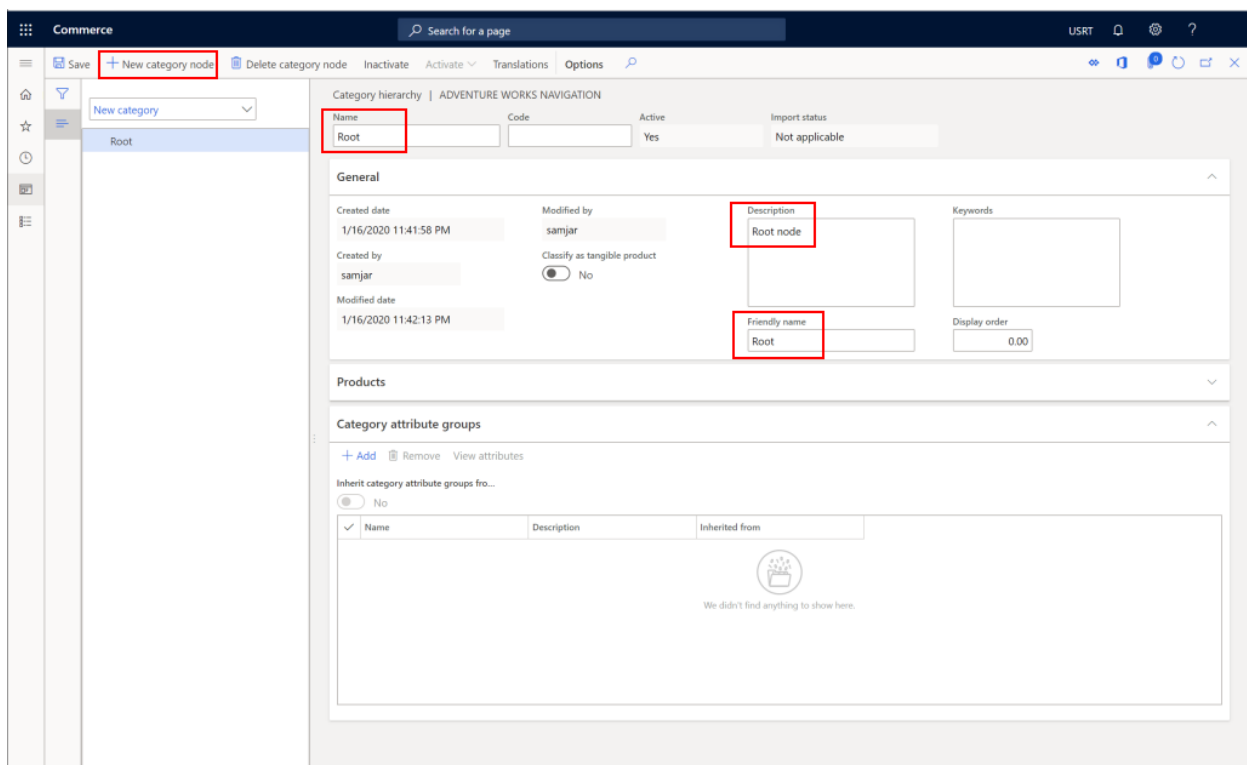
A channel navigation hierarchy is used to group and organize products into categories so that the products can be browsed online or in point of sale (POS).

## Create a channel navigation hierarchy

To create a channel navigation hierarchy, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Products and categories > Channel navigation categories**.
2. On the action pane, select **New**.
3. In the **Name** box, enter a name.
4. In the **Description** box, enter a description.
5. Select **Create**.
6. On the action pane, select **New category node** to create a root node.
7. In the **Name** box, enter a name.
8. In the **Description** box, enter a description.
9. In the **Friendly name** box, enter a friendly name.
10. On the action pane, select **Save**.

The following image shows an example root node.



The screenshot displays the Microsoft Dynamics 365 Commerce interface for creating a new category node. The main pane shows the 'New category node' form for a 'Root' node. The form is titled 'Category hierarchy | ADVENTURE WORKS NAVIGATION' and includes the following fields and sections:

- Name:** Root
- Code:** (empty)
- Active:** Yes
- Import status:** Not applicable
- General:**
  - Created date:** 1/16/2020 11:41:58 PM
  - Modified by:** samjar
  - Description:** Root node
  - Keywords:** (empty)
  - Created by:** samjar
  - Classify as tangible product:** No
  - Modified date:** 1/16/2020 11:42:13 PM
  - Friendly name:** Root
  - Display order:** 0.00
- Products:** (empty)
- Category attribute groups:**
  - Inherit category attribute groups from:** No
  - Table:**

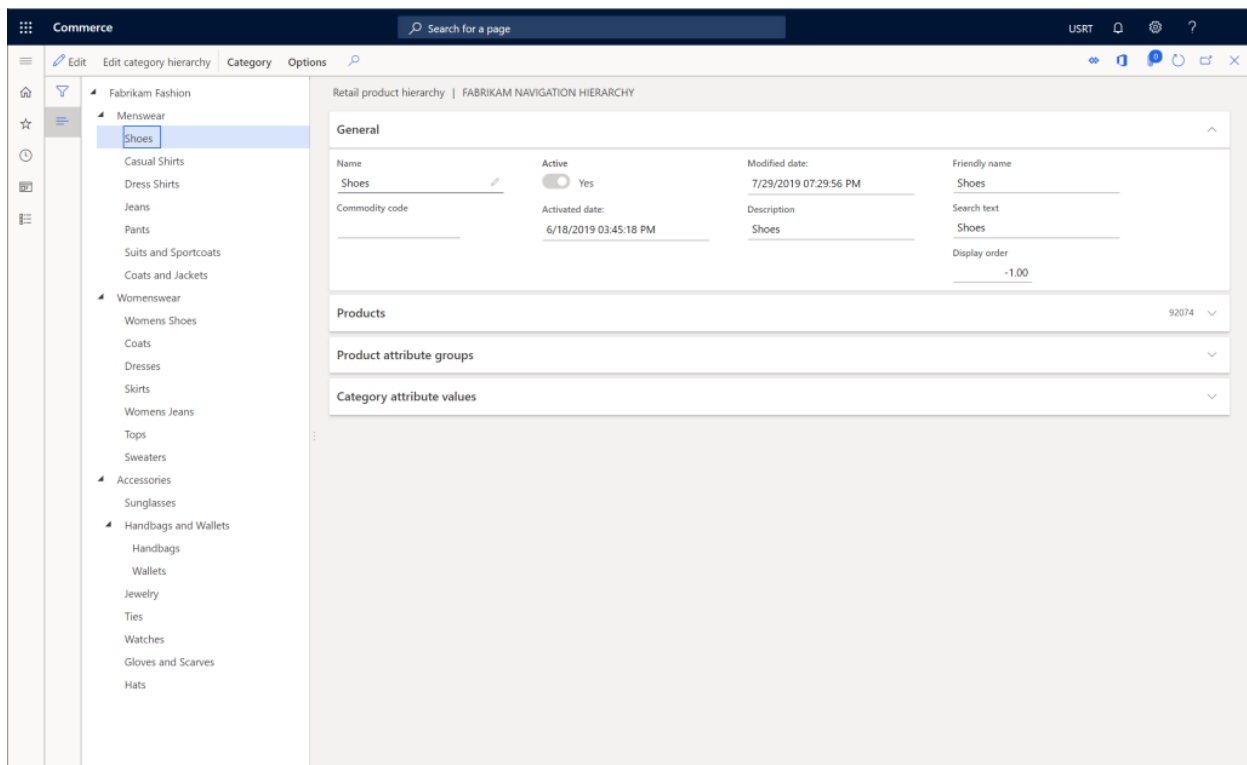
Name	Description	Inherited from
We didn't find anything to show here.		

# Create navigation category nodes

To create any additional navigation category nodes to represent the product categories on the channel, follow these steps.

1. In the navigation pane, select the parent node to add a category to.
2. On the action pane, select **New category node**.
3. In the **Name** box, enter a name.
4. In the **Description** box, enter a description.
5. In the **Friendly name** box, enter a friendly name.
6. In the **Display order** box, enter a display order (optional).
7. On the action pane, select **Save**.

The following image shows an example of a completed channel navigation hierarchy.



# Add products to category nodes

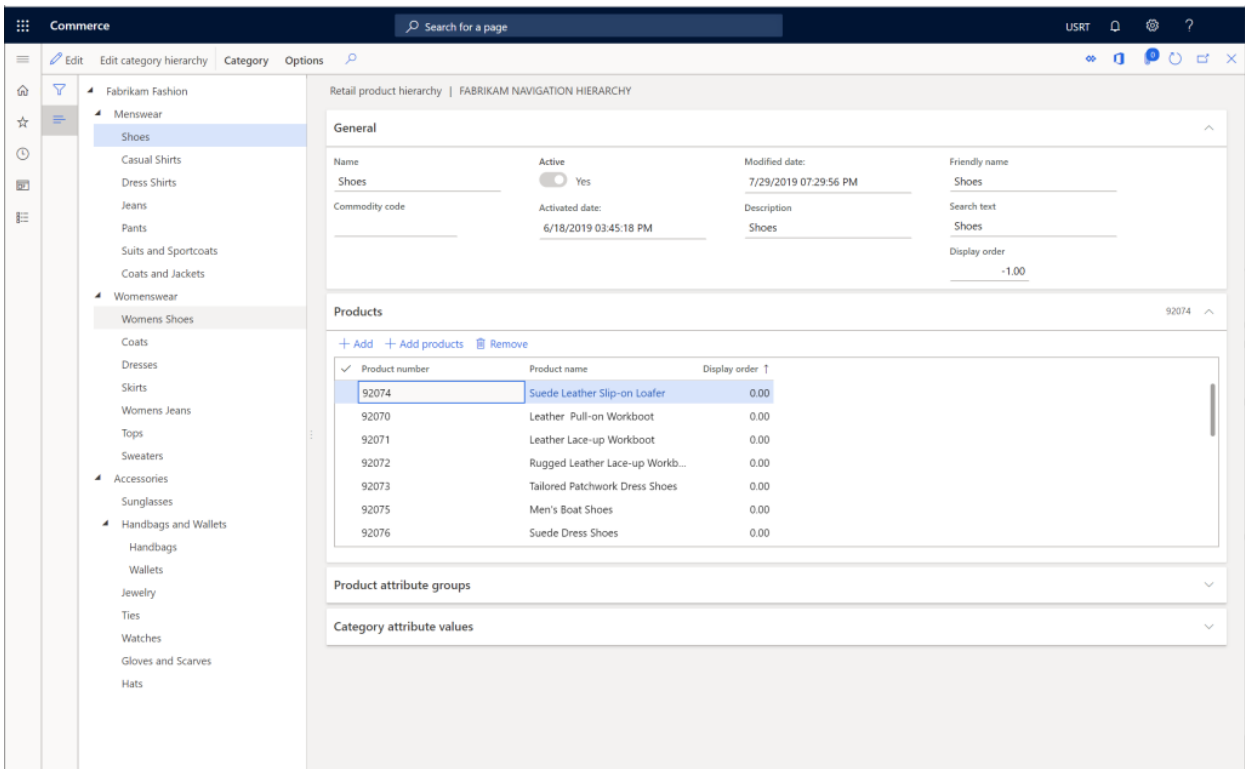
To add products to category nodes, follow these steps.

1. Select a category node.
2. Under **Products**, select **Add**.
3. Find the new product(s) you want to add using product number or product name, and then select **OK**.
4. On the action pane, select **Save**.

## NOTE

Adding products to a node inside the channel navigation hierarchy is not sufficient for the products to show up on a selected channel, the products must also be assorted to a product.

The following image shows an example node with products added.



## Add product attribute groups to category nodes

### NOTE

Attribute groups must be created before you can add them to a node inside the channel navigation hierarchy.

To add product an attribute group to a category node, follow these steps.

1. Select a category node.
2. Under **Product attribute group**, select **Add**.
3. Find the attribute group(s) you would like to add, and then select **OK**.
4. On the action pane, select **Save**.

The following image shows a sample node with product attribute groups added.

Commerce Search for a page USRT

Edit Edit category hierarchy Category Options

Retail product hierarchy | FABRIKAM NAVIGATION HIERARCHY

**General**

Name: Shoes Active: Yes Modified date: 7/29/2019 07:29:56 PM Friendly name: Shoes  
 Commodity code: Activated date: 6/18/2019 03:45:18 PM Description: Shoes Search text: Shoes  
 Display order: -1.00

**Products** 92074

**Product attribute groups**

Add Remove View attributes

Name	Retail attribute group type	Description	Inherited from
MensShoes	Default		Shoes
FabrikamCommon	Default	Fabrikam common attributes	Shoes

Inherit attribute groups from parent...  
 No

**Category attribute values**

Reset value Translate

Name	Type	Inherited from
ColumnIndex	IntegerDomain	Shoes
IndexInsideColumn	IntegerDomain	Shoes

**GENERAL**

Name: ColumnIndex  
 Type: IntegerDomain  
 Value: 1

## Additional resources

[Set up assortments](#)

[Manage attributes and attribute groups](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure a channel to use a channel navigation hierarchy

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to configure a channel to use a channel navigation hierarchy in Microsoft Dynamics 365 Commerce.

## Overview

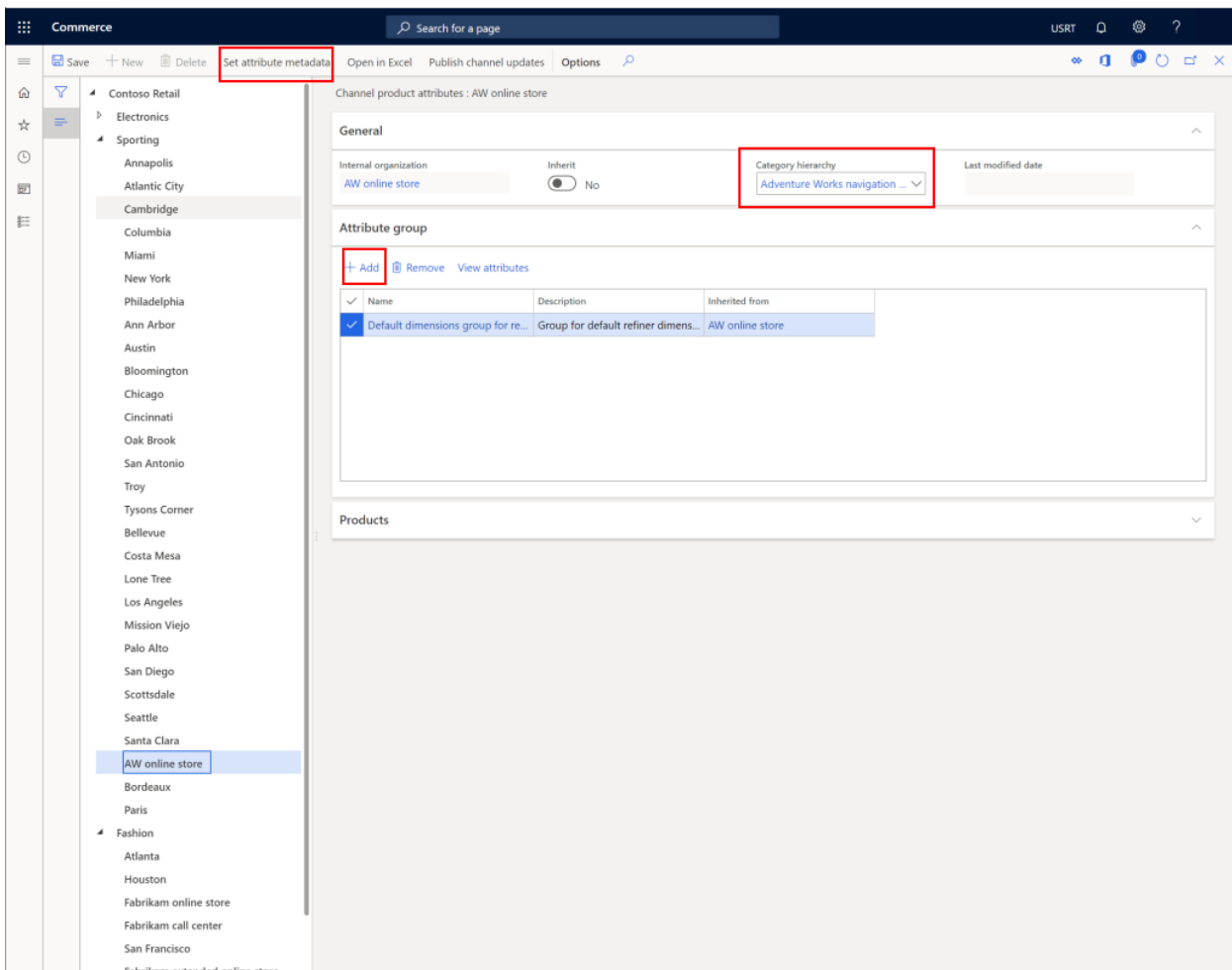
Channel navigation hierarchies organize products into categories so that the products can be browsed on an e-Commerce site or at points of sale (POS). Retail and online channels must be configured with channel navigation hierarchies.

## Configure the channel

To configure a channel to use a channel navigation hierarchy, follow these steps.

1. In the navigation pane, go to **Modules > Retail and commerce > Channel setup > Channel categories and product attributes**.
2. Select the channel to configure.
3. On the action pane, select **Set attribute metadata**.
4. In the **Category hierarchy** drop-down list, select the appropriate channel navigation hierarchy.
5. On the action pane, select **Save**.
6. Under **Attribute group**, add any attribute groups that will be global attributes for all nodes.

The following image shows how to configure a channel to use a channel navigation hierarchy.



## Set attribute metadata

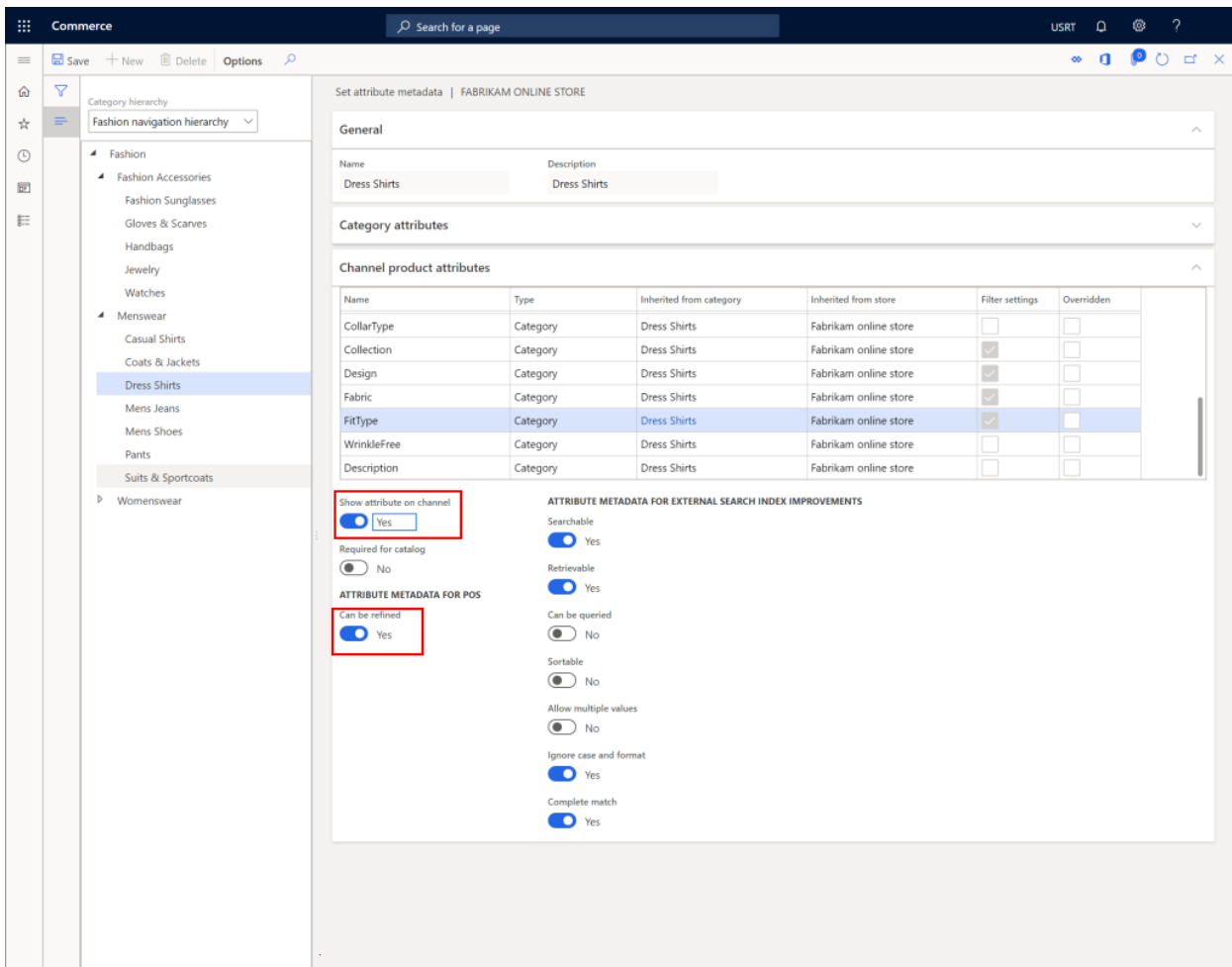
Setting the attribute metadata will allow configuration of attributes on each node.

To set attribute metadata, follow these steps.

1. On the action pane, select **Set attribute metadata**.
2. For each node select **Channel product attributes**.
3. Set **Show attribute on channel** to **Yes** and **Can be refined** to **Yes**, to enable refiners on that channel.
4. After configuring each node as desired, on the **Action pane**, select the **Save** button to save.
5. Select the **X** in the top right corner to exit this screen back to the **Channel categories and product attributes** page.

The following image shows an example set of channel product attributes configured on a channel category node.





## Publish changes

For changes to take effect, you will need to publish the changes.

To publish changes, follow these steps.

1. On the action pane, select **Publish channel updates**.
2. In the **Publish channel updates** pane, select **OK**.

The following image shows how to publish channel updates.

Commerce

Search for a page

Save New Delete Set attribute metadata Open in Excel Publish channel updates Options

Channel product attributes : AW online store

General

Internal organization: AW online store Inherit: No

Attribute group

Name	Description	Inherit
Default dimensions group for re...	Group for default refiner dimens...	AW c

Products

Run in the background

OK Cancel

## Additional resources

[Create a channel navigation hierarchy](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Assortment management

2/18/2021 • 5 minutes to read • [Edit Online](#)

## Overview

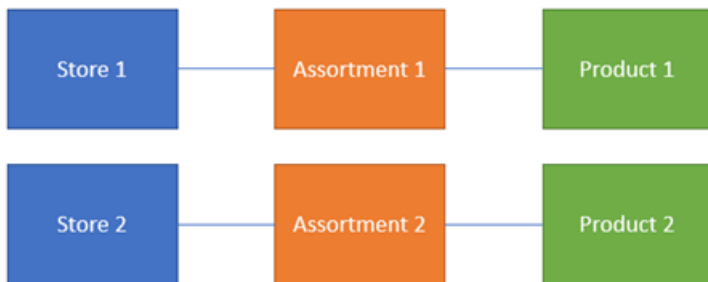
Dynamics 365 Commerce provides *assortments* that let you manage product availability across channels. Assortments determine which products are available at specific stores and during a specific period.

In Commerce, an assortment is a mapping of one or more channels (or groups of channels, when organization hierarchies are used) to one or more products (or groups of products, when category hierarchies are used).

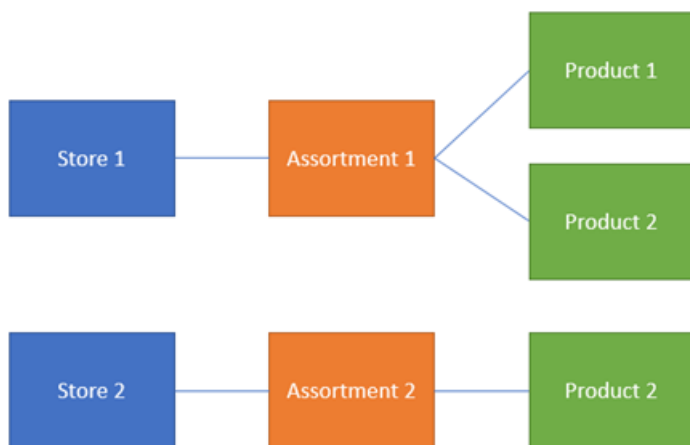
The overall product mix of a channel is determined by the published assortments that are assigned to the channel. Therefore, you can configure multiple active assortments per channel.

### Basic assortment setup

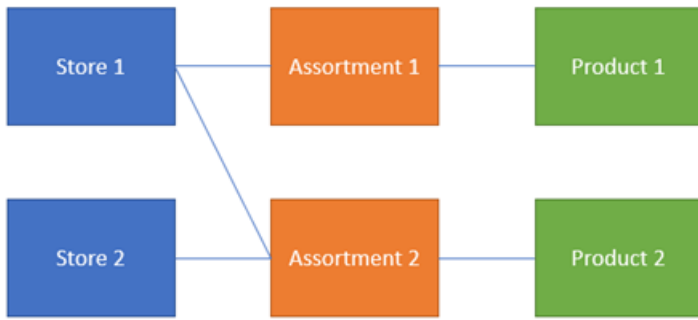
In the following example, a unique assortment is configured for each store. In this case, only product 1 is available at store 1, and only product 2 is available at store 2.



To make product 2 available at store 1, you can add the product to assortment 1.

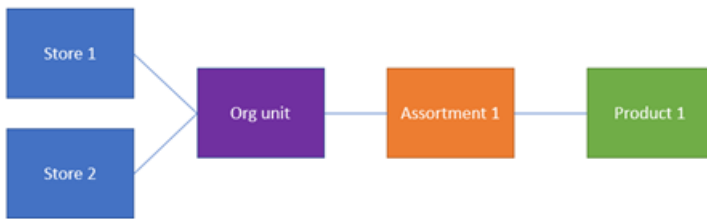


Alternatively, you can add store 1 to assortment 2.



### Organization hierarchies

In situations where multiple channels share the same product assortments, you can configure the assortments by using the Commerce assortment organization hierarchy. When nodes from this hierarchy are added, all channels in that node and its child nodes will be included.



### Product categories

Similarly, on the product side, you can include groups of products by using product category hierarchies. You can configure assortments by including one or more category hierarchy nodes. In this case, the assortment will include all products in that category node and its child nodes.

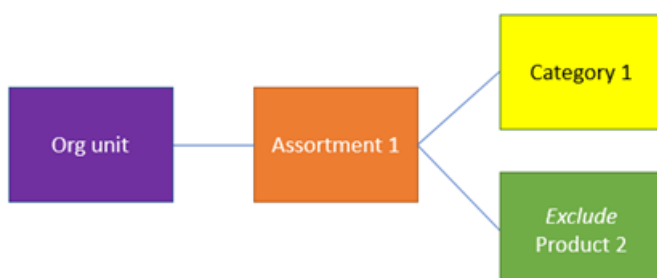


### Excluded products or categories

In addition to including products and categories in assortments, you can use the Exclude option to define specific products or categories that should be excluded from assortments. In the following example, you want to include all the products in a specific category, except product 2. In this case, you don't have to define the assortment product by product or create additional category nodes. Instead, you can just include the category but exclude the product.

#### NOTE

If a product is both included and excluded in one or more assortments by definition, the product will always be considered excluded.



## Global and released products

Assortments are defined at a global level and can contain channels from multiple legal entities. The products and categories that are included in assortments are also shared across legal entities. However, a product must be released before it can actually be sold, ordered, counted, or received in the channel (for example, in the point of sale [POS]). Therefore, although two stores in different legal entities can share an assortment that contains the same products, the products are available only if they have been released to those legal entities.

## Dynamic and static assortments

Assortments can be defined with specific channels and products or by including organization units and categories. Assortments including references to these groups are considered dynamic assortments. If the definition or contents of those groups change while the assortment is active, the definition of the assortment will also change.

For example, an assortment is originally defined and published so that it references a category of products. If additional products are later added to the category, those products are automatically included in the definition of the existing assortment. You don't have to manually add the products to the assortment. Similarly, if an organization unit is added to a different node, the organization unit's assortment is automatically adjusted based on that definition.

## Stopped products

You can "stop" released products for the sales process by turning on a setting in the **Default order** settings. This setting is most often used when a product is at the end of its life and should not be sold at any channel. Assortments respect this setting, and stopped products won't be assorted, regardless of the assortment configuration.

## Blocked products

In addition to stopping sales of a product, you can temporarily block sales of a product. You can configure this setting on the **Commerce** tab of a released product. Blocked products are still assorted, but you will receive a message in the POS that states that the product can't be sold.

## Date effectivity

Assortments are date-effective. Therefore, retailers can configure when products should or should not be available per channel. You can define and publish assortments ahead of time, and specify the start and end dates. The products will automatically become available or unavailable on the specified dates.

## Process assortments batch job

Assortments that are defined in Commerce must be processed before they take effect. This processing is done for the following reasons:

- Assortment definitions must be de-normalized so that channels can more easily consume them. A product mix for a channel can be defined through multiple assortments that span various date ranges. When some of this information is calculated ahead of time on the server, performance at the channel is improved.
- The products and channels in the assortment can change outside the assortment itself. Dynamic assortments that contain references to categories or organization units must be processed periodically so that they include or exclude records, based on their current assignment.

## Implementation considerations

Consider the following implementation requirements as you plan and manage assortments for your Commerce implementation:

- **Data replication and database size** – Although assortments help serve the business need to manage product availability, they are also an important tool for managing the size of channel and offline databases. Well-managed assortments help reduce the amount of data that must be processed and replicated to

channel and offline databases. They also help reduce the number of records that must be persisted. Fewer records in these databases will increase performance when you add items to a transaction, search, and browse for products.

- **Date-effective/expiring assortments** – One of the most effective tools for managing the number of products in channel and offline databases is the date effectivity of assortments. If you leave open-ended (non-expiring) assortments for seasonal products or products that are at the end of their life, these databases will grow indefinitely. You can use various approaches to help manage this situation. For example, you can maintain separate assortments for seasonal products and products that are always available.
- **Sales and returns outside assortments** – This capability helps retailers effectively manage their assortments by letting them limit the number of available products to products that belong to the core product mix for the store. This capability also helps retailers handle situations where a product was mistakenly omitted from an assortment, or where a product was returned outside the effective dates for the assortment.

If product data doesn't exist in the channel database, the POS makes real-time calls to headquarters to retrieve the required information, so that the product can be sold, returned, or put on a customer order. Product information that is retrieved in this manner is available only during the scope of that transaction. The product isn't added to the assortment definition. Therefore, subsequent real-time calls will be made as required.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

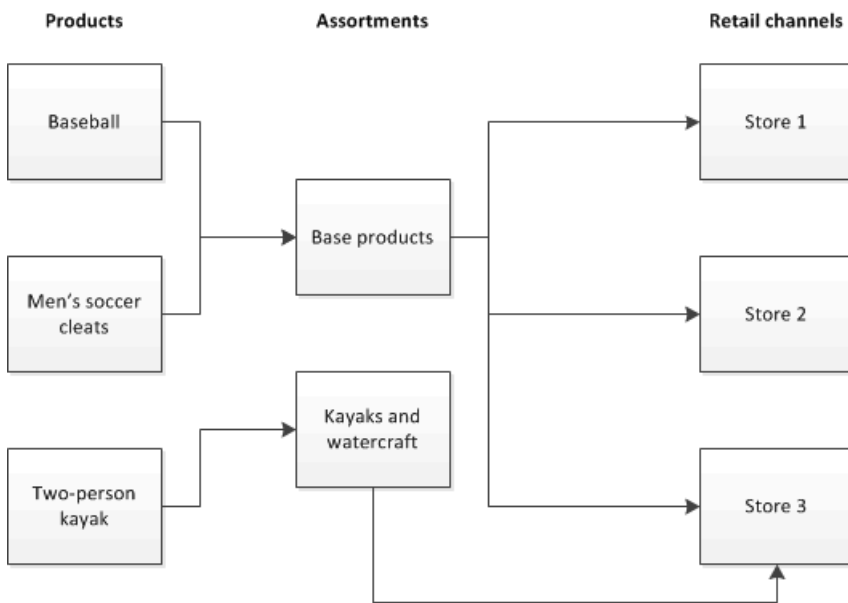
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up assortments

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article describes what an assortment is and explains how to set up assortments in Dynamics 365 Commerce.

An assortment is a collection of related products that you assign to a commerce channel, such as a brick-and-mortar store or an online store. You use assortments to identify the products that are available in each store. An assortment can include categories of products. Therefore, all products that are assigned to a specific category are included in the assortment. An assortment can also include specific products and specific variants of products. By setting up an assortment, you can assign thousands of products to your channels at that same time, in any combination that your stores require. You can set up as many product assortments as you require. Each product can be included in one or more assortments, and each assortment can be assigned to one or more channels. For example, you define one assortment that includes a base set of products. All stores receive this assortment. You then define another assortment that includes only large sporting equipment. Only your larger stores receive this assortment. The following diagram shows how products can be assigned to assortments, and how those assortments can be assigned to channels.



## Prerequisites

Before you can set up an assortment and assign it to a commerce channel, you must complete the following tasks.

TASK	DESCRIPTION
Set up a channel.	Channels represent a brick-and-mortar store, an online store, or an online marketplace. You must set up at least one channel and configure the options for the store. Assortments are assigned to stores to identify the products that a particular store carries.

TASK	DESCRIPTION
Create an organization hierarchy.	After you set up the commerce channels for your organization, you must configure an organization hierarchy that represents the organizational structure of your channels. An organization hierarchy can be used for assortments, replenishment, and reporting. By adding your channels to an organization hierarchy, you can assign assortments to groups of stores. Instead of assigning the assortment individually to each store, you assign the assortment to the high-level organization node. Then, whenever a new channel is added to the high-level organization node, that channel automatically inherits any assortments that were assigned to the higher-level organization node. You can assign assortments only to channels that are included in an organization hierarchy that is assigned the <b>Commerce assortment</b> purpose.
Define products.	Before you can add products to an assortment, you must add them in Commerce. You can add products manually, or you can import them from a vendor. After you add the products, you must release them to a legal entity. Only products that have been released to a legal entity can be made available to your channels. Products that haven't yet been released to a legal entity can be added to an assortment, and the assortment can be approved. However, until the products have been released to a legal entity, they can't be made available to the channels.
Set up a category hierarchy.	When you create your commerce products, you can group and categorize them by using the category hierarchy feature. You can create one core hierarchy to group and categorize all products that you distribute through your channels. You can also create separate, supplemental category hierarchies to group or categorize your products for special purposes, such as promotions or assortments. By using category hierarchies, you can assign all the products in a specific category to an assortment. Any products that are added to the category that is included in the assortment are automatically included in the assortment. Then, the next time that the commerce assortment scheduler is run, these products become available to the channels that the assortment is assigned to.

## Setting up an assortment

After you complete the prerequisites, you can create an assortment and assign it to your channels. To set up an assortment, you must complete the following tasks.

1. Create a new assortment, or copy an existing assortment.
2. Select the channels or the high-level groups of channels that the assortment applies to.
3. Add product categories, individual products, or product variants to the assortment. You can include all products in a specific category, or you can exclude selected products from a category that is included in the assortment.
4. Publish the assortment. When you publish an assortment, the assortment scheduler is automatically run. This process generates the list of products. When this process is completed, the products become available to the channels that the product assortment is assigned to. If changes are made to an assortment that has been published, or to the channels that the assortment is assigned to, the assortment must be updated. To update the assortment when changes are made, you can run the assortment scheduler as a batch job.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage assortments (November 2016)

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates how to create and publish a new product assortment and uses the demo data company USRT.

1. Click Category and product management.

## Create an assortment

1. Click the Assortments tab.
2. Click New.
3. Click Assortment.
  - The Assortment ID is required and must be a unique value.
4. In the Assortment name field, type a value.
5. In the Effective date field, enter a date.
6. In the Expiration date field, enter a date.
7. Expand the Commerce channels section.
8. Click Add line.
9. In the tree, select 'Contoso Retail\Electronics\Boston'.
10. Click Add.
11. Click OK.
12. Expand the Products section.
13. Click Add line.
14. In the Category field, enter or select a value.
15. Click Save.

## Publish an assortment

1. Click Publish.
2. Click Yes.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up bar codes

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes how to use bar codes in Dynamics 365 Commerce.

You can use bar codes to purchase and sell products, track product variants, and set up customers and employees. You can also use bar codes to issue and endorse coupons, gift cards, and credit memos. You can set up products so that they have standard bar codes or custom, in-house bar codes. Products can have more than one bar code. For example, a product might have multiple bar codes if it comes from various manufacturers, or if it has variants that are based on size, style, or color. Bar codes can include the weight or price of the product. Bar code masks are templates that are used to create bar codes.

## NOTE

If you assign a unique bar code to each variant combination, you can scan the bar code at the register and let the program determine which variant of the product is being sold. You can also collect and view statistics about sales by variant. Each size, color, and style group can be assigned a unique number that identifies that group in the bar code. Commerce uses the bar code mask to automatically generate bar codes for each variant combination. This functionality can be useful if there are many sizes, colors, and styles, because the number of combinations increases significantly as each variant code is added. If this functionality isn't used, bar codes must be manually assigned to each combination that represents a product variant.

You can create bar codes manually or automatically. To create bar codes, complete the following tasks in the order in which they are listed.

1. [Set up bar code mask characters.](#)
2. [Set up bar code masks.](#)
3. Configure bar code setups.
4. [Create bar codes for products.](#)

## Additional resources

[Set up bar code masks](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up bar code masks

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to set up bar code mask characters, bar code masks, and how to assign bar code masks to bar codes.

## Set up bar code mask characters

Bar code masks are used to create bar codes and to quickly identify bar codes that are scanned into the point of sale (POS). Masks are comprised of characters which act as placeholders that indicate the format for the bar codes that will be created. To configure a bar code mask, you need to set up bar code mask characters. Go to **Retail and Commerce > Inventory management > Barcodes and labels > Mask characters**. Click **New** to create bar code mask characters. Mask characters can be created to indicate the following bar code data.

FIELD	DESCRIPTION
Product	Placeholder for product ID.
Any number	Used to specify a number that will be hard coded in bar codes.
Check digit	Indicates that the bar code format in a bar code mask uses a check digit to confirm the validity of a bar code.
Size digit	Indicates size in a bar code created for a product variant which includes size.
Color digit	Indicates color in a bar code created for a product variant which includes color.
Style digit	Indicates style in a bar code created for a product variant which includes a style.
EAN license code	Placeholder for EAN license issued for EAN license codes.
Price	Indicates price for price embedded bar codes.
Quantity	Indicates quantity in quantity/random weight embedded bar codes.
Employee	Indicates bar code segment for employee ID number used for bar code POS login.
Customer	Indicates customer ID segment.
Data entry	<i>Not yet implemented.</i>
Discount code	<i>Deprecated</i> as of Dynamics 365 for Retail Spring 2017 release. Previously: Indicates discount code for a bar code that's used to add a discount to a point of sale transaction.

FIELD	DESCRIPTION
Coupon code	Indicates coupon code for a bar code used to add a discount to an order. This replaced discount code.
Gift card	Indicates a gift card number when issuing or paying by gift card.
Loyalty card	Adds a loyalty customer to the transaction, and can be used when paying by loyalty.

## Define bar code masks

After bar code mask characters are specified for the necessary bar code masks, go to **Retail and Commerce > Inventory management > Barcodes and labels > Barcode mask setup**. On this page, you can define bar code masks that use the previously specified characters. These bar code masks will be used when generating bar codes and will also help to identify bar codes scanned at the POS.

1. Click **New** to create a new bar code mask.
2. Enter values in the **Mask ID** and **Description** fields, and then select a bar code mask type in the **Type** field.
3. In the **General** section, select a value in the **Bar code standard** field, and then specify the bar code prefix, if one is required.
4. In the **Bar code mask segment** section, add bar code segments that will be used in the bar code to be created.

As an example, to create a bar code mask with mask ID 'Product', you'd do the following:

1. Create a new bar code mask and select type 'Product'.
2. Select a bar code standard, for example, 'Code 39'.
3. Provide a prefix to be used to easily identify the bar code. For example, '22'.
4. Add a mask segment. The 'Product' mask segment will be selected.
5. Provide a length for the product segment, for example, '10'. The length should match the length of a product ID commonly used in the store. The mask will be displayed as a preview in the **General** section under **Mask**.

## Assign bar code masks to bar codes

Bar codes masks must be assigned to bar codes before they can be used. Continuing with the previous example, to assign the bar code mask to a bar code, do the following:

1. Go to **Organization administration > Setup > Bar codes**. Click **New** to create a new bar code.
2. Enter values in the **Barcode setup** and **Setup** fields.
3. In the **General** section, in the **Bar code type** field, select 'Code 39'. In the **Mask ID** field, select the 'Product' mask previously created.
4. Under **Size**, enter '12'.
5. Click **Save**.

The bar code mask can now be used to create bar codes for products. The above steps are examples of how to create bar code masks for products, but they also illustrate how to create bar code masks for any of the other supported bar code types. Bar code masks, types, and lengths should be adjusted for use in your specific environment.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create and configure extended warranties

2/18/2021 • 12 minutes to read • [Edit Online](#)

This topic covers extended warranties and describes how to create and configure them in Microsoft Dynamics 365 Commerce.

## Overview

Customers are increasingly choosing extended support and services when they buy products, especially consumer products that are sold at a premium price point, such as phones and computers. By providing extended warranties for purchase, retailers can help build customer loyalty. Extended warranties let customers know where they can go for service and support. Therefore, they can have confidence that their issues will be handled effectively.

Extended warranties can be sold to customers in a retail channel during the initial product purchase. They can also be sold for a limited time after the initial purchase.

### Warranty item setup

Dynamics 365 Commerce provides functionality that lets you create a warranty item and set attributes for it. These attributes include the association between a product and a warranty item, the price of the warranty, and the duration of the warranty. After a warranty item is configured and released to the organizational unit, a retailer can sell warranties through Modern Point of Sale (MPOS), online stores, and other retail channels.

### Warranty item sales

Extended warranties are sold in a retail channel during the initial product purchase. They can also be sold for a limited time after the initial purchase.

At the point of sale (POS), sales associates are prompted to add an extended warranty when a related product is added to a customer's cart. Therefore, an upsell or cross-sell opportunity is presented to sales associates as part of the sales flow.

Customers can also return later and buy an extended warranty for a product that they previously purchased. In these cases, a sales associate can look up the original transaction and sell the customer the related extended warranty item.

### Warranty terminology

The following table defines some warranty-related terms.

TERM	DESCRIPTION
Extended warranty/Warranty	An <i>extended warranty</i> refers to a service agreement or contract that provides a prolonged warranty to customers. The extended warranty includes the additional service of replacing or repairing goods during the extended warranty's coverage period.

TERM	DESCRIPTION
Manufacturer's warranty	<p>A <i>manufacturer's warranty</i> (often referred to as a <i>limited warranty</i>) is the warranty that customers receive when they purchase a product. Here are some features of a manufacturer's warranty:</p> <ul style="list-style-type: none"> <li>• The warranty cost is included in the cost of the product. Customers don't have to pay any additional amount for a manufacturer's warranty.</li> <li>• Depending on the product category, a manufacturer's warranty typically lasts 30 days, six months, or one year. (For most consumer electronics, the warranty lasts one year).</li> <li>• The warranty covers any defects that are caused by mechanical or electrical failures. Coverage is limited, and it doesn't include any accidental damage to the purchased product. Customers who want to protect the products that they purchase from everyday damages should invest in an extended warranty. Extended warranties last two to ten years, depending on the product category. They also have wider coverage and cover everyday mishaps such as drops, spills, and stains.</li> </ul>
Warranty item	<p>A <i>warranty item</i> is an extended warranty item that is sold for a warrantable item. An example is a two-year accidental protection plan for laptops.</p>
Warrantable item	<p>A <i>warrantable item</i> is a serialized product that a warranty is sold for. For example, a laptop is a warrantable item that two-year and three-year extended warranties are sold for.</p>
Warranty group	<p>A <i>warranty group</i> is a relationship between warranty items and warrantable items. The POS uses warranty groups to determine which warranty items sales associates should be prompted to add when a warrantable item is added to a customer's cart.</p>
Warranty policy	<p>A <i>warranty policy</i> is an entity that is created in Commerce when a warranty policy is sold. A warranty policy includes information such as the start and end dates of the purchased warranty item, terms and conditions, and the serial number of the warranted product. Warranty policy numbers can be shared with customers, so that they have a reference for the extended warranty item that they purchased.</p>

## Create a warranty item

To create a warranty item in Commerce, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Products**.
2. Select **New** to create a warranty item.
3. In the **New product** dialog box, in the **Product type** field, select **Service**.
4. In the **Product subtype** field, select **Product**.
5. In the **Product service type** field, select **Service**.



6. In the **Product name** field, enter the product name.
7. In the **Retail category** field, select a value in the drop-down dialog box, and then select **OK**.
8. In the **Product number**, enter the product number.
9. Select **OK**.
10. On the **Product details** page, on the **Warranty** FastTab, set the **Unit of time** and **Length of time** fields.

FIELD NAME	VALUE	DESCRIPTION
Unit of time	<b>Day(s), Week(s), Month(s), or Year(s)</b>	This field specifies the unit of time that is used for the warranty.
Length of time	A positive integer value	This field specifies the duration of the warranty in the selected unit of time.

For example, for a two-year warranty, set the **Unit of time** field to **Year(s)** and the **Length of time** field to **2**. Alternatively, set the set the **Unit of time** field to **Month(s)** and the **Length of time** field to **24**, as shown in the following illustration.

## WAR-00001 : 2 Year Extended Protection Plan for Laptops

### General

#### IDENTIFICATION

Product number

WAR-00001

Product name

2 Year Extended Protection Plan for

Search name

2YearExtendedProtect

Description

Extended coverage for your device including accidental damage coverage for two years.

#### ADMINISTRATION

Product dimension group

Storage dimension group

SiteWH

Tracking dimension group

#### VARIANTS

Color group

Style group

Size group

### Warehouse

### Warranty

Unit of time

Month(s)

Length of time

24

11. Select **Save** to save the warranty item.
12. Release the warranty product to the company so that it can be sold. For more information, see [Set up retail products](#).
13. On the **Released product details** page, on the **Warranty** FastTab, set the **Price range base**, **Lower limit**, and **Upper limit** fields.

FIELD NAME	VALUE	DESCRIPTION
------------	-------	-------------

FIELD NAME	VALUE	DESCRIPTION
Price range base	None, Base price, or Selling price	<ul style="list-style-type: none"> <li>• <b>None</b> – The <b>Lower limit</b> and <b>Upper limit</b> values of price ranges aren't applicable.</li> <li>• <b>Base price</b> – A given warranty will be applicable if the base price (that is, the price without discounts) of the warrantable item is between the <b>Lower limit</b> and <b>Upper limit</b> values that are specified here, based on the price of the warrantable item.</li> <li>• <b>Selling price</b> – This value is reserved for future use.</li> </ul>
Lower limit, Upper limit	A positive integer value	These fields define the upper and lower price limits of the warrantable item, and how the current warranty item is applicable to the warrantable item. These limits can be based on the warrantable item's base price (also known as the manufacturer's suggested retail price [MSRP]). If the <b>Price range base</b> field is set to <b>Base price</b> , only a warrantable item (product) that has a base price between the <b>Lower limit</b> and <b>Upper limit</b> values will trigger a prompt to add the warranty item at the POS.

For example, the following illustration shows the **Price range base** field set to **Base price**, the **Lower limit** field set to \$500, and the **Upper limit** field set to \$1000.

[Released product details](#)

### WAR-00001 : 2 Year Extended Protection Plan for Laptops

Warranty  Product name  FIFO

---

**Purchase**

---

**Promote**

---

**Deliver**

---

**Sell**

---

**Warranty**

WARRANTABLE PRICE RANGE	Lower limit	WARRANTY DURATION
Price range base <input type="text" value="Base price"/>	<input type="text" value="500.00"/>	Length of time <input type="text" value="24"/>
	Upper limit <input type="text" value="1,000.00"/>	Unit of time <input type="text" value="Month(s)"/>

14. Assort the warranty item to the channel where it will be sold. For more information, see [Set up assortments](#).

## Example

A laptop warrantable item (product) has a base price \$999, and there are two laptop warranty items:

- Warranty\_1 has a lower limit of \$500 and an upper limit of \$1,000, and the **Price range base** field is set to **Base price**.
- Warranty\_2 has a lower limit of \$1,001 and upper limit of \$2,000, and the **Price range base** field is set to **Base price**.

In this case, when the laptop warrantable item is added to a customer's cart, a prompt to add Warranty\_1 will be shown at the POS, because the price of the laptop is between the lower and upper limits for Warranty\_1.

### NOTE

For this example, if you want prompts to be shown for both Warranty\_1 and Warranty\_2, regardless of the price of the warrantable item, set the **Price range base** field to **None**.

## Configure channel-specific settings

Channel-specific settings let you specify whether a prompt to add a warranty item should be shown at the POS when a warrantable item is added to a customer's cart.

To configure channel-specific setting in Commerce, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Warranty > Warranty settings**.
2. On the **Channel specific** tab, in the **Prompt for warranty** column for your channel, follow one of these steps:
  - Select the check box if a prompt for the warranty item should be shown at the POS when the warrantable item is added to the cart.
  - Clear the check box if no prompt for the warranty item should be shown at the POS when the warrantable item is added to the cart.
3. Run the 1070 job to sync the data to the channel.

## Configure a number sequence for warranty policies

Each warranty policy is uniquely identified by a warranty policy number that is generated by a number sequence. For more information about number sequences, see [Number sequences overview](#).

To configure a number sequence for warranty policies in Commerce, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Warranty > Warranty settings**.
2. On the **Number sequences** tab, in the row for the **Warranty policy** reference, enter or select a value in the **Number sequence code** field.

## Set up a warranty group

A warranty group is a relationship between warranty items and warrantable items. The POS uses warranty groups to determine which warranty items sales associates should be prompted to add when a warrantable item is added to a customer's cart.

To set up a warranty group in Commerce, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Warranty > Warranty groups**.
2. Select **New** to create a warranty group.
3. In the **Name** field, enter a name for the new group.

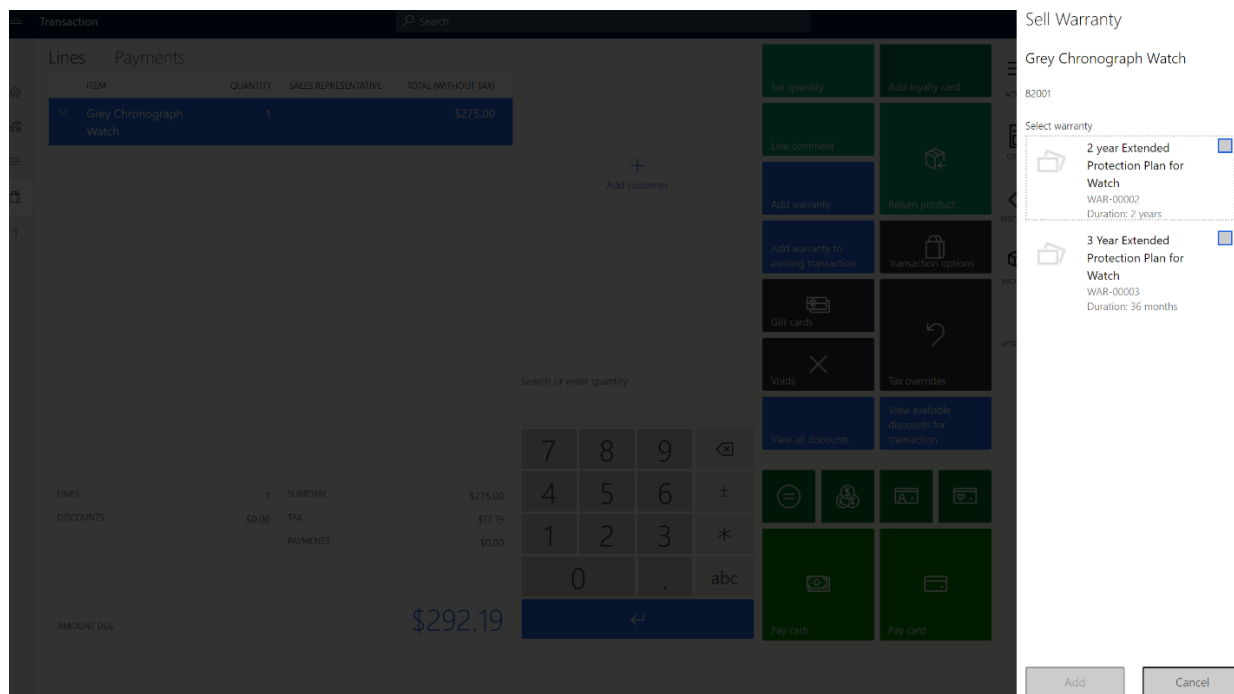
4. On the **General** FastTab, in the **Description** field, enter a description of the group.
5. On the **Warranty products** FastTab, select **Add line** to add a warranty item.
6. In the **Display order** field, enter a number to rank the warranty group at the POS. The POS will show warranty items in order of ascending rank in the warranty prompt.
7. On the **Warrantable products** FastTab, select **Add line** to add warrantable products.
8. If the warranty item is applicable to a whole category of warrantable items (products), select the category in the **Category** field. If the warranty item is applicable to a specific warrantable item (product), select the product in the **Product** field.
9. On the **Applicable channels** FastTab, select **Add line** to add the channel where you want to sell the warranty item.
10. Select **Save** to save the configuration.
11. Select **Publish** to publish the warranty group.
12. Run the **1040** job to sync the data to channel.

## Sell warranty items at the POS

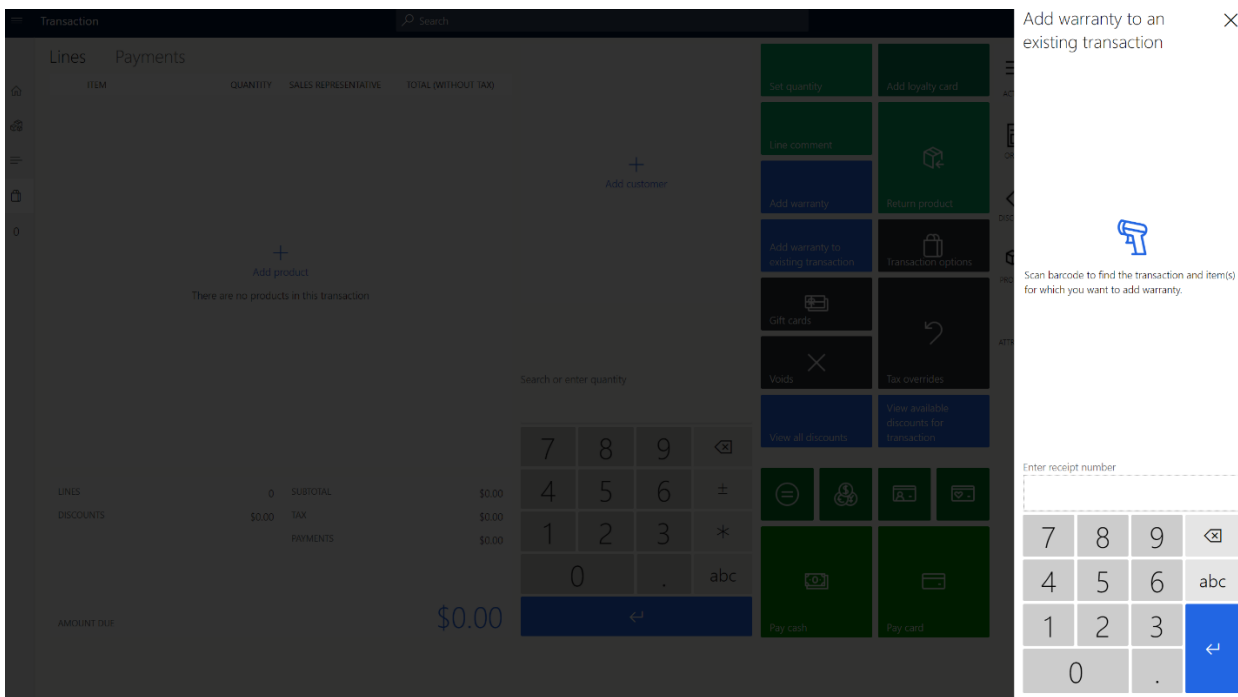
Two POS operations let sales associates sell warranty items during the workflow for customer purchases:

- **Add warranty** – This operation triggers a prompt that shows applicable warranties for a warrantable item that is selected in the cart.
- **Add warranty to existing transaction** – This operation lets sales associates sell warranties for warrantable items that were previously sold. Sales associates can find the original transaction for a warrantable item by entering the receipt number of the transaction.

The following illustration shows an example of a POS terminal page with a prompt to add a warranty item for the current purchase of a warrantable item.



The following illustration shows an example of the feature for adding a warranty item for a warrantable item that was previously sold.



## Process warranty transactions

When warranties are sold in cash-and-carry transactions, after the transactions are posted in Commerce headquarters, Commerce users can run the **Process warranty transactions** job to process the warranty transactions and create warranty policies.

To process warranty transactions in Commerce headquarters, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Warranty > Process warranty transactions**.
2. In the **Choose organization nodes** dialog box, in the **Organizational hierarchy** field, select a value.
3. In the **Available organization nodes** list, select either an individual store or, if you want to create the batch job for a group of stores, a node.
4. Select the right arrow button to add your selection to the **Selected organization nodes** list.
5. Select the **Run in the background** tab.
6. Set the **Batch processing** option to **Yes**, and then select **Recurrence**.
7. In the **Define recurrence** dialog box, in the **Start date** field, select or enter a start date for the recurrence.
8. In the **Start time** field, select or enter a start time for the recurrence.
9. Follow one of these steps:
  - Select the **No end date** option if the recurrence should never end.
  - Select the **End after** option if the recurrence should end after a specific number of runs. If you select this option, enter the number of runs.
  - Select the **End by** option if the recurrence should end by a specific date. If you select this option, select or enter the date.
10. Select **OK**.
11. Select **OK**.

# Warranty policies

When an extended warranty is sold, a warranty policy entity is automatically created. Warranty policy numbers can be shared with customers, so that they have a reference for the warranty item that they purchased. The properties of warranty policies include the effective start date and expiration date of the warranty, terms and conditions, and the serial number of the warrantable item that the warranty was sold for.

## NOTE

Warranty policy properties are automatically generated when warranty policy entities are created. Currently, they can't be manually configured or edited.

The following table describes the warranty policy properties and their values. In Commerce headquarters, the database table is named WARRANTYPOLICY.

PROPERTY NAME	VALUE	DESCRIPTION
PolicyNumber	A character string (maximum of 20 characters)	The warranty policy number
WarrantiedItemId	A character string (maximum of 20 characters)	The ID of the warrantable item
WarrantiedInventoryLotId	A character string (maximum of 20 characters)	The inventory lot ID of the warrantable item
WarrantiedSerialNumber	A character string (maximum of 20 characters)	The serial number of the warrantable item
WarrantiedFulfilledDate	A date	The fulfillment date of the warrantable item
WarrantyItemId	A character string (maximum of 20 characters)	The ID of the warranty item
WarrantyInventoryLotId	A character string (maximum of 20 characters)	The inventory lot ID of the warranty item
WarrantySalesDate	A date	The sale date of the warranty item
WarrantyEffectiveDate	A date	The effective date of the warranty policy
WarrantyExpirationDate	A date	The expiration date of the warranty policy
CustAccount	A character string (maximum of 20 characters)	The customer account number
Status	<b>Created, Voided, Effective, or Expired</b>	The status of the warranty policy
Notes	A character string (a maximum of 255 characters)	Notes about the warranty policy, such as terms and conditions

# Frequently asked questions (FAQ)

## Why don't I see a warranty prompt in the POS?

Make sure that the warranty item is assorted to the channel. Also make sure that the warranty group is configured so that it includes the relevant channel.

## When I try to add a warranty to an existing transaction and enter the customer order receipt number, why don't I see any transaction line items?

Receipts can be found only if a pull job (P-job) is run to upload the receipts to Commerce headquarters. To run the P-job, go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**, select the **P-0001** job, and then select **Run now**.

## Why is the warranty feature applicable only to serialized products?

A warranty is a service that is provided for a specific, unique product. In Dynamics 365, a product can be uniquely identified only by a serial number.

## Additional resources

[Set up retail products](#)

[Set up assortments](#)

[Number sequences overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Retail discounts

2/18/2021 • 15 minutes to read • [Edit Online](#)

## Overview

This topic provides an overview of the discounts functionality in the Dynamics 365 Commerce. It explains the properties found on the various discount forms, and best practices for discount management. However, this topic does not cover the various discounts types in detail, for example, simple, quantity, mix and match, and threshold discounts. These details will be covered in separate topics created for each of these discount types.

Because retailers require flexible discounting, and discount styles and types vary by industry, there are many ways to define discounts in Commerce. The discounting functionality was added on top of the existing discount functionality in the core product (Supply Chain Management), resulting in some duplication of functionality. As a result, the discount types can be configured for five different entities: customer, loyalty program, channel, catalog, and affiliations. Because of the number of discounting options, it's especially important that you plan and document your discounting strategy.

## Creating discounts

Each discount type has a dedicated page that you use to create and manage the discount. Commerce also has an **All discounts** page and a **Pricing and discounts management** workspace, both of which you can use to create a new discount of any type.

### Discount headers and discount lines

All discounts have a header and one or more lines. All discount types have properties defined on the header and some discount types have additional properties defined per line. For example, quantity discounts have quantity tiers. People often think about discounts in Commerce in terms of the discount header only and assume that all the lines on the discount are related to each other because they share a discount header. However, this view of discounts is too simplistic. For simple discounts and quantity discounts, it is more accurate to think of each discount line as an independent discount that shares some properties with the other discount lines. In fact, the pricing engine evaluates simple and quantity discounts in just this manner. Each discount line for simple discounts and quantity discounts is independent. For simple discounts, it is easy to understand that each discount line is independent of all other discount lines on the same discount, because there isn't a quantity or amount criterion required to qualify for the discount. For quantity discounts, you might think that the lines can be combined to reach the quantity criterion for a discount, but they don't. The quantity tiers must be reached independently for each line of a quantity discount.

When you create discounts, we recommend that you always avoid or minimize overlapping discount lines. Overlapping discount lines occur when two or more discount lines in the same discount can be applied to the same product. In this case, the pricing engine must treat the discount as two or more independent discounts that must then be evaluated against each other to find the best discount amount. In addition, it can be difficult for a user to know what the discount will be just by looking at the discount definition.

## Managing discounts

### Settings and options that are common to all discounts

This section describes the properties that are common to all types of discounts.

When you manage discounts, it's important that you understand each discount option individually, but it is equally important that you understand which options affect each other and how. The common settings for

discounts fall into two categories. In the first category are settings that filter discounts for consideration. Examples include **Status**, **Currency**, and **Unit of measure**. Settings in the second category control the order in which multiple discounts are considered and applied. Examples include **Discount concurrency mode** and **Pricing priority**. The following image shows the various properties of a discount.

The screenshot shows the configuration page for a discount. At the top, there are fields for 'Discount' (ST100000), 'Name' (Grand Opening), and 'Discount type' (Discount). Below this is a 'General' section with fields for 'Status' (Enabled), 'Currency' (USD), 'Discount concurrency mode' (Best price), 'Discount account', 'Coupon code required' (No), and 'PRICING PRIORITY' (Override priority: No, Pricing priority: 0). The 'Details' section includes 'Description', 'Disclaimer', and 'Text for fiscal receipt'. The 'Price/discount' section is collapsed. The 'Validation period' section has 'Date validation type' (Standard), 'Effective date', 'Expiration date' (Never), 'VALIDATION PERIOD' (Discount period number), 'Description', 'Start date', and 'End date'. The 'Lines' section shows a table with columns for Category, Product, Product variant, Configuration, Size, Color, Style, Name, Unit, and Line.

Category	Product	Product variant	Configuration	Size	Color	Style	Name	Unit	Line
<input checked="" type="checkbox"/>	BMX Tires & Tubes	0005					Basic Inner Tube	ea	Incl
	BMX Gloves	0008					Mesh BMX Gloves	ea	Incl
	Fielding Gloves	0015					Youth Utility Baseball Glove	ea	Incl

## Discount ID

This field is labeled **Discount** and holds a unique ID for each discount. It's set when you first create a discount and can't be changed later. In **Commerce parameters**, you can set up independent number sequences for each type of discount. In this case, make sure that the number sequences won't collide. For example, you can use a unique prefix for each discount type. For example, D for **d**iscount, Q for **q**uantity, MM for **m**ix and **m**atch, and T for **t**hreshold.

## Discount name

This field is a short free text field that is used to describe the discount. The string value in this field is shown in the MPOS and CPOS cart line and printed on Modern Point of Sale (MPOS) and Cloud POS (CPOS) customer receipts. Therefore, your cashiers and end customers will see this description. It is the primary means for MPOS/CPOS users and customers to know which discount was applied.

## Discount type

There are four types of discounts in Commerce: **Discount**, **Quantity**, **Mix and match**, and **Threshold**. The discount type is set when you first create a discount and can't be changed later. The discount type determines whether there is a quantity or amount criterion that must be met to qualify for the discount.

## Status

The status of a discount can be either **Enabled** or **Disabled**. When you first create a discount, the status is **Disabled**. Discounts can only be edited when they are disabled. When discount data is pushed to a channel, disabled discounts are not pushed. If a discount was previously enabled and pushed to the channel, then this new push will also remove the discount from the channel. When you change the status to **Enabled**, various validation checks are performed on the discount, depending on the type of discount. The list of validation checks has increased in recent updates of the product to prevent incomplete or poorly defined discounts from being pushed to commerce channels. Here is a partial list of the validations that are performed when you enable a discount:

- A discount must have at least one discount line.
- The percentage value for a percentage discount must be more than 0 (zero) and less than or equal to 100.
- The amount value for an amount discount must be more than 0 (zero). Zero and negative amounts aren't

valid.

- A discount must have at least one price group. A discount that doesn't have a price group will never be applied to a transaction.
- A Unit of measure (UoM) is required for quantity and mix-and-match discount lines.
- For quantity discounts that have two or more quantity tiers, the discount value is validated to increase as quantities increase.
- For threshold discounts that have two or more threshold tiers, the discount value for each tier must be equal to or more than the largest discount of the previous tier.
- For mix-and-match least-expensive discounts, the number of least-expensive products must be more than 1 and less than the number of products that are required to trigger the discount.

## Currency

The currency of a discount defines the currency of all amount and price fields on the discount. Different discount types have different field options. The currency also acts as a filter during discount calculation. In Commerce, all sales order and MPOS/CPOS transactions have a currency, and the pricing engine will consider only discounts that have the same currency.

## Discount concurrency mode

This determines which discounts compete on a transaction, and which discounts are compounded together. The three values for this option are **Exclusive**, **Best price**, and **Compound**.

When the value is **Exclusive** or **Best price**, only one discount can be applied to a transaction line. The only difference between **Exclusive** and **Best price** is the order that the discounts are considered and applied in. **Exclusive** discounts are always evaluated and applied before **Best price** and **Compound** discounts, if all other settings are the same. Therefore, **Exclusive** and **Best price** discount never compete for the best price. Two or more **Exclusive** discounts will compete for the best price, as will two or more **Best price** discounts.

When the value is **Compound**, the discount can be compounded with any other discount that is also set to **Compound**. Therefore, two or more **Compound** discounts will all be applied to a transaction line. When multiple **Compound** discounts are applied to a transaction line, they are applied in the following order:

1. Discount price discounts
2. Amount-off discounts
3. Percentage-off discounts

**Compound** discounts compete with **Best price** discounts when both types apply to a transaction line.

Therefore, the **Compound** setting is used to determine which discounts are combined. Depending on the discount concurrency control mode used, two or more **Compound** discounts can be combined and compete with the **Best price** discounts that apply to the same products. The discount or discounts that have the largest total discount amount are applied.

## Discount account

Commerce lets you post discount amounts for a transaction to a separate general ledger (GL) account. The discount GL account is set by the product or customer. Commerce offers a unique way to separate the discount amounts during posting. You can post each type of discount to a specific GL account. Both options can make it easier for you to determine which discounts or discount types are being used in your general ledger.

### NOTE

When the discount account posting feature is enabled, then an additional debit entry and credit entry are made to reclassify the discount posting out of the Commerce discount GL account and into the discount GL account.

## Coupon code required

Starting with version 7.2 of the app, the call center coupons are now merged with discounts. For a discount, when **Coupon code required** is set to **Yes**, the **Status** field and the standard date fields, **Effective date** and **Expiration date**, are not available. These properties are controlled by equivalent properties that are on the **Coupons** page.

When **Coupon code required** is set to **Yes** on a discount, the discount is applied to a transaction only if the coupon code or bar code is provided by MPOS/CPOS. The values of the coupon codes and bar codes are defined and configured in a separate page named **Coupons**. The **Coupons** page is where the coupon is linked to the discount. When **Coupon code required** is set to **No**, a coupon code isn't required, and the discount will always be applied through its price groups.

### **Override priority and Pricing priority**

These two fields work together. When **Override priority** is set to **Yes**, the **Pricing priority** field becomes available for editing. You can then select a pricing priority to set directly on the discount. When **Override priority** is set to **No**, the priority is inherited from the priority of the price group associated with the discount. In the case of multiple price groups association, the priority number is determined by selecting the highest pricing priority of all the price groups associated with the discount.

### **Match all associated price groups**

In Commerce version 10.0.16 and later, a configuration called **Match all associated price groups** is available on all discount forms. If the configuration is enabled, the discount will be considered only if all the price groups associated to the discount are applicable to the transaction. For example, if the two price groups named "PG-Student" (price group for student affiliation) and "RP-Houston" (price group for the Houston store) are associated to a discount, and **Match all associated price groups** is enabled, the discount will be considered only for students who are shopping in the Houston store. This configuration provides a way to restrict affiliation and loyalty-based discounts to limited stores.

#### **NOTE**

If two or more channel price groups are associated to a discount, and **Match all associated price groups** is enabled, the discount won't apply because a transaction can be associated to only one store. Therefore, all the price groups associated to the discount don't match.

### **Description**

This field is a free-form text field. It isn't used in the MPOS/CPOS system or in transactions.

### **Disclaimer**

This is a free-form text field. It isn't used in the MPOS/CPOS system or in transactions.

### **Line type**

This field is on all discount lines. The possible values are **Include** and **Exclude**. This field is used in combination with the **Category**, **Product**, and **Variant** fields to define the set of products that the discount is applied to. Exclude discount lines always override include discount lines. When **Line type** is **Exclude** many of the other fields on the discount line are grayed out, as they do not apply.

### **Unit of measure**

**Unit of measure** (UoM) is a field on all discount lines except threshold discount lines. This field is label **Unit** in Commerce. The **Unit of measure** field acts as a *filter* to determine whether a discount should be applied to a transaction line. The UoM on the transaction line must match the UoM on the discount line. Otherwise, the discount line isn't considered during discount calculation. No UoM conversion is done during discount calculation.

### **Category, product, variant, and dimensions**

**Category**, **Product**, **Variant**, and **dimensions** are the last discount settings that are common to all discounts.

These fields are set on each discount line and specify what is being discounted. They act as a filter when the pricing engine searches for discounts that can be applied to a transaction. These fields are related to each other according to these rules – categories contain products, and products can come in different variations of size, color, style, and configuration.

The pricing engine does not use the parent/child relationships of categories, products, and variants to order discounts during discount calculation. This behavior differs from the way that the pricing engine handles sales price trade agreements. For example, both a discount for 10 percent on a category and a discount for 5 percent on a product in the same category will be considered. The larger of the two discount amounts will then be used, provided that all other properties are the same and the discounts aren't set to **Compound**, in which they both will be combined. If you want to force a product discount to be used over a category discount you can use pricing priority or the discount's concurrence mode to cause one discount to be applied before another.

When you edit discounts, the **Category**, **Product**, **Variant**, and **Dimensions** settings act as filters for each other. The **Category** and **Product** fields are automatically set from the *Commerce Category Hierarchy* if a product or variant is entered directly. The following sections provide detailed descriptions of each of these fields.

### **Category**

At a minimum, you must set the **Category** field. You can select any category from the product category hierarchy or any category from a supplemental category hierarchy. However, you can't select categories from channel navigation hierarchies or other non-commerce hierarchies. If only a category is specified on a discount line, the discount will be applied to any product in that category, even products that are added to the category after the discount has been created, provided that all other discount criteria are met, such as currency and UoM.

#### **NOTE**

The category that you select on a discount line is hierarchy specific. Therefore, you can't specify a value by typing a partial value in the field, as you can in most Commerce fields. If you type in a full category name, the drop-down list will expand, and that category will be selected. In addition, you can press Alt+Down arrow to expand the selection dialog box and then press Tab to move between the hierarchy selection and hierarchy tree within the drop-down list, so that you can use the field without using a mouse.

The capability to work with categories is a key differentiator between discounts and trade agreement discounts, and the main reason that we discourage you from using trade agreement discounts. Categories are organized in a multi-level hierarchy. By contrast, the item discount groups that are used by trade agreements are only a single level of grouping, and each group is specific to one of the three trade agreement discount types, such as Line discount, Multiline discount, and Total discount. Therefore, for trade agreements, if you want to use the same set of products in all three trade agreement discount types, you must create and manage three independent discount groups. However, for discounts, you must maintain only one category. You can then use that category in all four discount types. You can also use the same category in price adjustments, assortment management, and loyalty management.

### **Product**

The product can be a released product or a released product master. All discounts are company specific. Therefore, they work only with released products. If you select a product master, the discount will be applied to all variants of the product, even variants that are released after the discount is created, provided that all other discount criteria are met, such as currency and UoM.

### **Variant**

When you select a variant on a discount line, the discount will be applied to just that variant, provided that all other discount criteria are met, such as currency and UoM.

### **Dimensions**

Starting with the Retail 8.1.1 release, we have added the capability to set up discounts at a dimension level for a product. This provides the flexibility to choose one or more dimensions of a product as discount lines. This saves

the merchandizing manager from individually adding the variants on which the discounts apply. For example, you can specify a discount on all variants with a specific style or you can specify a discount on all variants that are of a specific color and style.

#### NOTE

The capability to set up promotions based on dimensions is not supported for price adjustments. The specific interface for defining the dimensions are removed in Retail versions 10.0.4 and later.

## Best practices

- Before you create discounts, document your discounting strategy and procedure. Keep your documentation up to date as your use of the product evolves.
- Use independent number sequences for each discount type and configure the number sequences so that the discount ID by itself indicates the discount type. For example, prefix the ID of each discount type with a different alphanumeric constant: **Q** for quantity, **MM** for mix and match, and so on.
- Test your discount configuration using the price simulator before you enable discounts. The price simulator has an option that lets you treat disabled discounts as enabled. This option was designed specifically for testing discounts before they are enabled.
- Expire discounts when they are no longer valid. In this way, you prevent the total number of discounts that the pricing engine considers during a transaction from growing unbounded. Otherwise, the performance of discount calculation can be affected over time.
- Leverage the supplemental categories to group the products, for example clearance products or last season products.
- Always avoid or minimize overlapping discount lines.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail sales price management

2/18/2021 • 23 minutes to read • [Edit Online](#)

This topic provides information about the process of creating and managing sales prices in Dynamics 365 Commerce. It focuses on the concepts that are involved in this process, and on the effects of the various configuration options for sales prices.

## Terminology

The following terms are used in this topic.

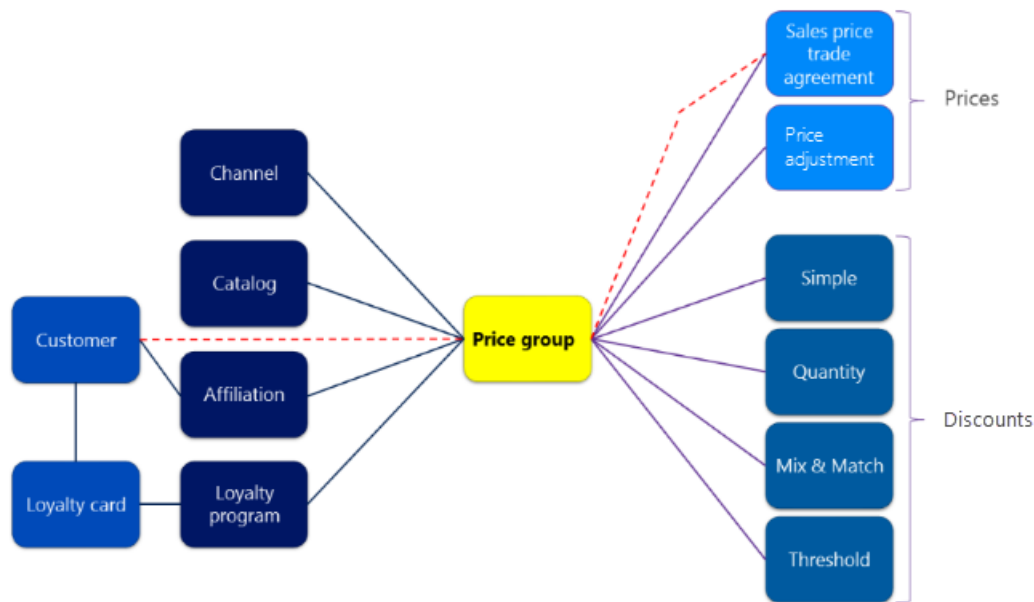
TERM	DEFINITION, USAGE, AND NOTES
Price	The single unit amount that a product sells for in a point of sale (POS) client or on a sales order. In this topic, the term <i>price</i> always refers to the sales price, not the inventory price or cost price.
Base price	The price that is set in the <b>Price</b> field on a released product.
Trade agreement price	The price that is set on a product or variant by using a trade agreement of the <b>Price (sales)</b> type.
Best price	When more than one price or discount can be applied to a product, the smallest price amount and/or the largest discount amount that produces the lowest possible net amount that the customer must pay. In this topic, the concept of best price is always referred to as "the best price." This best price differs from and should not be confused with the <b>Best price</b> enumeration value for a discount's concurrency mode.

## Price groups

Price groups are at the heart of price and discount management in Commerce. Price groups are used to assign prices and discounts to commerce entities (that is, channels, catalogs, affiliations, and loyalty programs). Because price groups are used for all pricing and discounts, it's very important that you plan how you will use them before you start.

By itself, a price group is just a name, a description, and, optionally, a pricing priority. The main point to remember about price groups is that they are used to manage the many-to-many relationships that discounts and prices have with commerce entities.

The following illustration shows how price groups are used. In this illustration, notice that "Price group" is literally at the center of pricing and discount management. The commerce entities that you can use to manage differential prices and discounts are on the left, and the actual price and discount records are on the right.



When you create price groups, you should not use a single price group for multiple types of commerce entities. Otherwise, it can be difficult to determine why a specific price or discount is being applied to a transaction.

As the red dashed line in the illustration shows, Commerce does support the core Microsoft Dynamics 365 functionality of a price group that is set directly on a customer. However, in this case, you get only sales price trade agreements. If you want to apply customer-specific prices, we recommend that you not set price groups directly on the customer. Instead, you should use affiliations.

Note that if the price group is set on the customer, then this price group gets associated to the sales order header of the orders created for this customer. If the user changes the price group on the order header, then the old price group gets replaced with the new price group only for the current order. For example, the old price group will not affect the current order, but it will still be associated to the customer for future orders.

The following sections provide more information about the commerce entities that you can use to set distinct prices when the price groups are used. The configuration of prices and discounts for all these entities is a two-step process. These steps can be done in either order. However, the logical order is to set the price groups on the entities first, because this step is likely to be a one-time setup that is done during implementation. Then, as prices and discounts are created, you can set the price groups on those prices and discounts individually.

## Channels

In the commerce industry, it's very typical to have different prices in different channels. The two primary factors that affect channel-specific prices are costs and local market conditions.

- **Costs** – The farther away a channel is from the product source, the more it costs to stock a product. For example, fresh produce has a limited shelf life and specific production requirements (for example, a growing season). During the winter, fresh lettuce likely costs more in northern climates than in southern climates. If you're setting prices for channels over a large geographical area, you will probably want to set different prices in different channels.
- **Local market conditions** – A store that has a direct competitor across the street will be much more price-sensitive than a store that doesn't have a direct competitor nearby.

## Affiliations

The general definition of an affiliation is a link to or association with a group. In Commerce, affiliations are groups of customers. Affiliations are a much more flexible tool for customer pricing and discounts than the core Microsoft Dynamics 365 concept of customer groups and discount groups. First, an affiliation can be used for both prices and discounts, whereas non-retail pricing has a different group for each type of discount and price. Next, a customer can belong to multiple affiliations but can belong to only one non-retail pricing group of each



type. Finally, although affiliations can be set up so that they are linked to a customer, they don't have to be. An ad-hoc affiliation can be used for anonymous customers at the POS. A typical example of an anonymous affiliation discount is a senior or student discount, where a customer can receive a discount just by showing a group membership card.

Although affiliations are most often associated with discounts, you can also use them to set differential pricing. For example, when a retailer sells to an employee, it might want to change the selling price instead of applying a discount on top of the regular price. As another example, a retailer that sells to both consumer customers and business customers might offer business customers better prices, based on their purchasing volume. Affiliations enable both these scenarios.

### **Loyalty programs**

In relation to prices and discounts, loyalty programs are basically just an affiliation that has a special name. Both prices and discounts can be set for a loyalty program, just as they can be set for an affiliation. However, the way that customers get loyalty pricing during a transaction or order differs from the way that they get affiliation pricing. Customers can get loyalty pricing only if a loyalty card is added to a transaction. When a loyalty card is added to a transaction, the loyalty program is also added. The loyalty program then enables special prices and discounts.

Loyalty programs can have multiple tiers, and the discounts can differ for different tiers. In this way, retailers can give frequent customers larger rewards without having to manually put those customers into a special group.

Loyalty programs have additional functionality besides prices and discounts. However, from the perspective of pricing and discounts, they are the same as affiliations.

### **Catalogs**

Some retailers use physical or virtual catalogs to market products to, and price them for, focused groups of customers. As part of their business model to target marketing via a catalog, these retailers can set differential prices on their various catalogs. Microsoft Dynamics 365 supports this capability by letting you define catalog-specific discounts and prices, just as you can define channel-specific or affiliation-specific discounts. When you edit a catalog, you can associate price groups with the catalog, just as you can associate them with a channel, affiliation, or loyalty program.

### **Best practices for price groups**

Don't use a price group for multiple entity types. Instead, use one set of price groups for channels, a different set of price groups for affiliations or loyalty programs, and so on. You can use a prefix or suffix in the name of the price group to visually group the various types of price groups that you're using.

Avoid setting price groups directly on a customer. Instead, use an affiliation. In this way, you can assign all types of prices and discounts to customers, not just sales price trade agreements.

## **Pricing priority**

By itself, a pricing priority is just a number and a description. Pricing priorities can be applied to price groups, or they can be applied directly to discounts. When pricing priorities are used, they let a retailer override the principle of the best price by controlling the order in which prices and discounts are applied to products. A larger pricing priority number is evaluated before a lower pricing priority number. Additionally, if a price or discount is found at any priority number, all prices or discounts that have lower priority numbers are ignored.

The price and a discount can come from two different pricing priorities, because pricing priorities apply to prices and discounts independently.

To use pricing priority for prices, you must assign a pricing priority to a price group and then create a sales price trade agreement for that price group.

The pricing priority feature was introduced to support the scenario where a retailer wants to apply higher prices

in a specific set of stores. For example, a retailer has defined regional prices for the east coast of the United States but wants higher prices for some products in New York City stores, because it costs more to sell some products in the city, and/or because the local market will bear a higher price.

As was described in the "Best price" section of this topic, the pricing engine typically selects the lower of two prices. Therefore, the retailer is usually prevented from using the higher of two prices in a store that has both the East coast and New York price groups. To resolve this issue before the pricing priority feature was introduced, the retailer had to define prices for every product two times and not assign both price groups. Alternatively, the retailer had to create extra price groups to isolate the products that have higher prices from products that have the usual, lower prices.

However, the pricing priority feature lets the retailer create a pricing priority for store prices that is higher than the pricing priority for regional prices. Alternatively, the retailer can create a pricing priority just for store prices and leave regional prices at the default pricing priority, which is 0 (zero). Both setups help guarantee that store prices will always be used before regional prices.

### Pricing priority example

Let's look at an example where store prices override other prices.

A national retailer sets most prices per region, and it has four regions: North east, South east, Mid-west and West. It has identified several high-cost markets that can support higher prices. These markets are in New York City, Chicago, and the San Francisco Bay area.

For this example, we will drill into the North east region. Store 1 is in Boston, and store 2 is in Manhattan. For the Boston store, two price groups are linked to the channel: North East and Store 1. For the Manhattan store, three price groups are linked to the channel: North East, NYC, and Store 2.

The retailer sets up two pricing priorities: High cost has a priority number of 5, and Store prices has a priority number of 10. (Remember that, by default, the pricing priority is 0 [zero], and a price or discount that has a higher priority number is used before a price or discount that has a lower priority number.) For the North East price group, the pricing priority is left at the default value of 0 (zero). For the NYC price group, the pricing priority is set to 5, because New York City is a high-cost market. For the Store 1 and Store 2 price groups, the pricing priority is set to 10.

Two products that the retailer sells are product 1, a commodity T-shirt, and product 2, brand-specific fashion jeans.

PRODUCT	NORTH EAST PRICE	NYC PRICE	STORE PRICE
T-shirt	\$15	Not set	Not set
Fashion jeans	\$50	\$70	Not set

The T-shirt sells for the same price (that is, \$15) at both the Boston and Manhattan stores, because only one price is set in the North East price group that is linked to both channels. The fashion jeans sell for \$50 in the Boston store, because that price is the only price that is available in that store. However, in the Manhattan store, two prices are available: \$50 and \$70. Because the pricing priority of 5 for the NYC price group is higher than the pricing priority of 0 (zero) for the North East price group, the price will be rung up as \$70 in the POS system.

#### NOTE

For each pricing priority, a full pass through the logic for the retail pricing engine is required. Therefore, to help maintain the performance of the price and discount calculation, you should use pricing priorities sparingly.

# Types of prices

In Microsoft Dynamics 365, you can set the price of a product in three places:

- Directly on the product (base price)
- In a sales price trade agreement
- In a price adjustment

The base price and trade agreement price are part of core Dynamics 365, and are available even if you don't use Commerce. The price adjustment functionality is available only in Commerce. The next section provides more information about each of these options for setting prices and explains how the options work together.

## Setting prices

### Base price

The easiest place to set the price for a product is directly on the product. The value that you set directly on a product is often referred to as the base price for the product. You set the base price in the **Price** field on the **Sell** tab of the **Released product details** page. The value that you enter is in the company currency. By default, the price is for a quantity of 1 of the unit of measure (UoM) that is set in the **Unit** field on the **Sell** tab. The actual price per unit of a product is based on the UoM, the price quantity, and the currency.

If a product has one price for everyone, the base price offers the most efficient way to manage the price of that product. Even if you use trade agreements to set prices, you might also set the base price on a product. Then, if you don't use an **All** trade agreement, you have a fallback price that is used when no trade agreement applies.

If a channel's currency differs from the company currency, the base price in that channel is determined by using currency conversion on the price that is set on the product.

Although the price unit isn't a common scenario, the pricing engine supports it. If the price unit is set to a value other than 0 (zero), the price per unit equals  $\text{Price} \div \text{Price unit}$ . For example, if a product's price is \$10.00, and the price unit is 50, the price for a quantity of 1 is \$0.20 ( $= \$10.00 \div 50$ ).

### Sales price trade agreement

By using the trade agreement journal, you can create sales price trade agreements for each product. In Microsoft Dynamics 365, there are three customer scopes for sales price trade agreements: **Table**, **Group**, and **All**. The customer scope determines the customers that a given sales price trade agreement applies to.

A **Table** sales price trade agreement is for a single customer that is set directly on the trade agreement. This scenario isn't a typical business-to-consumer (B2C) scenario. However, if it occurs, the pricing engine uses **Table** trade agreements when it determines price.

A **Group** sales price trade agreement is the type that is most often used with. Outside Commerce, **Group** sales price trade agreements are for a simple customer group. However, in Commerce, the concept of a customer group has been extended so that it's a more generic price group. A price group can be linked to a channel, affiliation, loyalty program, or catalog. For detailed information about price groups, see the "Price groups" section earlier in this topic.

#### NOTE

A trade agreement price is always used before the base price.

### Price adjustment

As the name implies, a price adjustment is used to modify the price that was either set directly on the product or set by using a trade agreement. A price adjustment can be used only to lower the price, not raise it. A price adjustment is the recommended way for retailers to create, track, and manage price markdowns for their

products over time.

There are three types of price adjustments: percentage off, amount off, and price. A price adjustment of the percentage off or amount off type is always applied to a sale transaction. However, a price adjustment of the price type is applied only if the adjusted price is less than the price that was set by using the base price or trade agreement price. Therefore, if the price that is set in a price adjustment is more than the unadjusted price, the price adjustment isn't used.

## Determining price for a product in a transaction

The calculation of the price and discount on a transaction uses the principle of finding the best price for the customer. According to this principle, if more than one price is found, the lowest price is used. Additionally, the combination of discounts that produces the largest discount amount for the whole transaction is used. In some cases, a smaller discount must be used on a single product, so that additional discounts can be applied to other products in the transaction.

The only exception to the principle of finding the best price for the customer is an option for mix-and-match least-expensive discounts. This option enables least-expensive discounts that favor the retailer when products are selected and grouped. Therefore, when a transaction includes more products than are required to qualify for the least-expensive discount, the pricing engine selects the products that produce the smallest possible discount amount for the customer.

The pricing engine returns three prices for every product: the base price, the trade agreement price, and the active price.

The base price is just the property on the product and is the same for everyone everywhere.

On the sales price trade agreement, if the **Find next** option is set to **Yes**, the lowest price that is found for applicable sales price trade agreements is used as the trade agreement price. Trade agreements can be found by using price groups or the **ALL** account code. Alternatively, trade agreements can be assigned directly to a customer. If the **Find next** option is set to **No**, the first trade agreement price that is found is used. If no sales price trade agreements are found, then the trade agreement price is set equal to the base price.

The active price is calculated by taking the trade agreement price and applying the largest price adjustment that applies to the product. If no price adjustments are found, or if the calculated active price is more than the trade agreement price, the active price is set equal to the trade agreement price. Remember that you can't raise the price of a product by using a price adjustment. The applicable price adjustments can be found only by using price groups that are assigned to a channel, catalog, affiliation, or loyalty program.

## Category price rules

The category price rules feature in Commerce gives you an easy way to create new trade agreements for all the products in a category. This feature also lets you automatically find existing trade agreements for the products in the category and expire them.

When you select the option to expire existing trade agreements, the system creates a new trade agreement journal for the products in the category that have an active trade agreement. However, the journal must be manually posted. Additionally, the category price rules can find existing trade agreements only if you're using the same price rule (that is, if you create a new price rule that uses the same category that was before). If you aren't using the same price rule, the existing trade agreements won't be expired.

The prices can be increased or decreased by using the **Price rule** and **Price basis** fields of the category price rules.

- In the **Price rule** field, select the type of price change to use:
  - **Markup** – A percentage of the price basis is used to calculate the sales price. For example, a product

that costs 10.00 and sells for 15.00 has a markup of 50 percent.

- **Margin** – A percentage of the sales price is used to calculate the amount of profit. For example, a product that costs 10.00 and sells for 15.00 has a margin of 33.3 percent.
- **Fixed amount** – An amount that is added to the price basis is used to calculate the sales price. For example, a product that costs 10.00 and sells for 15.00 has a fixed amount of 5.00.
- In the **Price basis** field, select the type of price to modify:
  - **Base cost** – The amount that the retailer paid to the supplier.
  - **Base price** – The sales price before trade agreements and price adjustments are applied.
  - **Current price** – The sales price after trade agreements and price adjustments are applied.

To easily update the prices of various products from different product categories, you can use the supplemental product categories together with the category price rules.

## Best practices

Microsoft SQL Server Express is often used for channel databases because of the cost (free). Keep in mind that SQL Server Express has hardware limitations and limits on data size. If you don't plan correctly, you can quickly reach the data size limits of SQL Server Express. This consideration applies not only to pricing but also to other areas of the product. Here are a few best practices that can help you reduce the size of your data:

- If you're using trade agreements, and your prices change, you should expire the old trade agreements by setting an end date. Over time, this approach helps reduce the number of trade agreements that are kept in channel databases. It also helps reduce the amount of data that the price calculation algorithm must work with.
- If your prices vary by product variant, consider using the product base price as the price of the most common variant. Then use trade agreements only for the variant prices that are exceptions. This approach helps reduce the number of trade agreement records. Because it's so easy to import data into Microsoft Dynamics 365, you might be tempted to import a trade agreement for every variant of every product. However, that approach can produce many trade agreements that have the same value. Therefore, it can needlessly increase the size of your data.
- Commerce processes variant-specific prices in order from most specific to least specific. If a product dimension doesn't affect the price, you don't have to define trade agreements for it. For example, a product is available in three colors and four sizes, but the price varies only by size. If you define a trade agreement for every variant, you create 12 records. Instead, you can define a trade agreement just for each size and leave the Color dimension blank. In this case, you produce only four records.

Alternatively, if not every value of a dimension produces a different price, you can define one trade agreement for the product master and leave all product dimensions blank. Then define a separate trade agreement just for each dimension value that produces a different price. For example, if the XXL size has a higher price, but all other sizes have the same price, you require only two trade agreements: one for the product master and one for the XXL size.

## Prices that include tax vs. prices that exclude tax

When you set sales prices in Dynamics 365, you don't specify whether the price value that you're setting includes or excludes tax. The value is just the price. However, the **Price includes sales tax** setting on channels lets you configure channels so that they either include or exclude tax from prices. This setting is set on the channel and can change even in a single company.

If you work with both inclusive and exclusive types of tax, it's very important that you set prices correctly, because the total amount that the customer pays will change if the **Price includes sales tax** setting on the channel is changed.

# Differences between retail pricing and non-retail pricing

A single pricing engine is used to calculate prices across all channels: Call center, Retail store, and Online stores. This helps in enabling the unified commerce scenarios.

Pricing is designed to work with retail entities instead of non-retail entities. Specifically, it's designed to set prices by store, not by warehouse.

The pricing engine **does not support** the following pricing features:

- Setting prices by Site or Site and Warehouse storage dimensions is not supported. If you only specify Site dimension on the trade agreements, then the pricing engine will ignore the Site and apply the trade agreement to all sites. If you specify both Site and Warehouse, then the behavior is undefined/untested because it's expected that retailers use the store price groups to control the prices for each store/warehouse.
- Attribute-based pricing is not supported.
- Vendor discount pass-through is not supported.
- The standard Supply Chain Management pricing engine supports the pricing calculation based on the "Requested ship date" and "Requested receipt date" along with the current date. However, retail pricing currently does not support these values. The reason is that for B2C scenarios customers do not expect the requested delivery date to affect the item price. In some cases, retailers have both B2B and B2C operations. For B2B operations it is common to change prices based on the delivery dates. These retailers can use Supply Chain Management pricing for their B2B business and retail pricing for their B2C business. Retail pricing kicks in only if the application user is added as a call center user, so the retailers can assign certain users who will work with the Supply Chain Management pricing and assign a few that will work with the Retail pricing, that is, these users should be added as a call center users. Additionally, the **Use today's date for calculating prices** property in the **Commerce parameters > pricing and discounts > Miscellaneous** section must be turned on. This way they can keep the using accounts receivable parameter value for Requested ship date or Requested receipt date for Supply Chain Management pricing, but the retail pricing will keep using the today's date for pricing calculation.

In addition, **only** the pricing engine supports the following pricing features:

- The price is based on product dimensions, in order from the most-specific variant price to the least-specific variant price to the product master price. A price that is set by using two product dimensions (for example, Color and Size) is used before a price that is set by using only one product dimension (for example, Size).
- The same price group can be used to control pricing and discounts.

## Pricing API enhancements

Price is one of the most important factors that controls the buying decisions of many customers, and many customers compare prices on various sites before they make a purchase. To help ensure that they provide competitive prices, retailers carefully watch their competitors and often run promotions. To help these retailers attract customers, it's very important that product search, the browse feature, lists, and the product details page show the most accurate prices.

In an upcoming release of Commerce, the **GetActivePrices** application programming interface (API) will return prices that include simple discounts (for example, single-line discounts that don't depend on other items in the cart). In this way, the prices that are shown are close to the actual amount that customers will pay for items. This API will include all the types of simple discounts: affiliation-based, loyalty-based, catalog-based, and channel-based discounts. Additionally, the API will return the names and validity information for the applied discounts, so that retailers can provide a more detailed description of the price and create a sense of urgency if the discount's validity will expire soon.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Price adjustments and discounts

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides information about price adjustments and discounts in Dynamics 365 Commerce.

In Commerce, you can make price adjustments to products, and can also set up discounts that are applied to a line item or a transaction at the point of sale (POS), in a call center sales order, or in an online order. Both price adjustments and discounts can be linked to price groups. For both price adjustments and discounts, you can specify a single start date and end date or a reoccurring period, a discount code, and a few additional attributes.

Price adjustments and discounts can be applied to products, variants, or categories. If more than one discount applies to a product, a customer might receive either one of the discounts or a combined discount, depending on the configuration of the discount. Commerce automatically applies the discount or combination of discounts that gives the best price to the customer. When you set up a price adjustment or a discount, be sure to confirm that price groups are assigned to the correct channels, catalogs, affiliations, or loyalty programs that you want the discount to apply to. Additionally, if you want to automatically generate the discount ID, set up number sequences on the **Commerce parameters** page before you define a new price adjustment or discount.

## NOTE

You can delete a price adjustment or a discount. However, statistical information will be lost.

## Types of discounts

There are many types of discounts:

- **Simple discount** – A single percentage or amount.
- **Quantity discount** – A discount that is applied when two or more products are purchased.
- **Mix and match discount** – A discount that is applied when a specific combination of products is purchased.
- **Threshold discount** – A discount that is applied when the transaction total is more than a specified amount.
- **Tender-based discount** – A discount that is applied when the transaction total is more than a specified amount and a specific payment type (for example, cash, credit, or debit card) is used for payment.
- **Shipping discount** – A discount that is applied when the transaction total is more than a specified amount and a specific mode of delivery (for example, two day shipping or overnight shipping) is used on the order.

Both price adjustments and discounts can be associated with price groups. Price groups can then be associated with channels, catalogs, affiliations, and loyalty programs.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Apply multiple Retail discounts to a product

2/18/2021 • 18 minutes to read • [Edit Online](#)

## Overview

This topic reviews all the factors considered when multiple discounts can be applied to a product. In this scenario, the commerce pricing engine applies as many discounts as it can, to maximize the total discount amount on a product. Multiple options affect the order in which the discounts are applied. Throughout this topic it's noted when a setting affects the order of discount application and exclusivity of a discount. The following settings affect how multiple discounts, applicable on a product, are processed.

- **Discount concurrency control model**
- **Pricing Priority**
- **Discount type (Simple, Mix and Match, Quantity, and Threshold)**
- **Discount concurrency mode (Exclusive, Best price, and Compound)**
- **Multiple occurrences mode**, when it's set to **Favor retailer** (for mix-and-match least-expensive discounts only)

The **Discount concurrency control model** is described below in detail; however, the rest of the properties are covered in [Retail discounts](#).

## Discount concurrency control model

The discount concurrency control model changes when and how multiple discounts are applied to products in a transaction. The **Best price and compound concurrency control model** option on the **Discounts** tab on the **Commerce parameters** page is different from the **Discount concurrency mode** property on each discount.

In earlier versions of the app, there was only one way to apply multiple discounts based on the **discount type**, **discount concurrency mode**, and **pricing priority** (if used) properties of discounts. Now, the discount concurrency control model setting affects how all discounts compete and compound together.

### Background on why this change was made

In previous versions of the app, you could directly customize the price engine by overlaying their custom business logic in the price engine. With the transition to an online service and to improve overall application lifecycle management, the Dynamics 365 application has been sealed and overlaying customizations are no longer allowed. New extensibility points have been added to enable the same types of customizations that were the most common. Most discount customizations are in one of the following categories.

- **Minor changes to existing discounts:** For example, moving the start date and end date from the discount header to the discount lines.
- **New discount types:** In some cases, companies need to introduce a new type of discount. For example, capping the total discount amount for a simple discount.
- **Changing the when and how (the flow) of multiple discounts being applied:** For example, having all mix and match discounts applied on top of quantity or simple discounts while still having quantity and simple discounts compete for best price or having store-specific and customer-specific discounts compete for best price and then compound the winning discount with loyalty program discounts.

The first two types of customizations are handled by providing a new extensibility model within the price engine that enables these scenarios. However, to address the third type of customization we expanded the functional

capabilities of the system by introducing this setting. A discount's concurrency mode and pricing priority already gave the user significant flexibility over the order of discount application. By introducing a new configuration setting that affects how a discount's concurrency mode and pricing priority interact, all discount ordering customization is covered, which results in the concurrency model option.

### **Best price and compound within priority, never compound across priorities**

This is the default and is the legacy way in which multiple discounts are processed. When this option is selected, all compound discounts within the same pricing priority are combined, and the combined result competes with any best price discounts in the same pricing priority. After a discount is applied to a product, all discounts at lower pricing priorities are ignored.

### **Best price only within priority, always compound across priority**

This is the new way multiple discounts can be processed. When this option is selected, discounts with **Discount concurrency mode** set to **Best price** and **Compound** are all treated as "best price" within a single pricing priority. When applied, the best price discount within a priority, is compounded with the best price and compound discounts at lower pricing priorities. In this concurrency control model, only a single discount can be applied to a product per pricing priority, and if that single discount is a best price or compound discount, then it will compound with all additional best price or compound discounts at lower pricing priorities.

### **Examples**

The following examples show how the pricing engine processes a pool of discounts for different concurrency control models.

#### **Example 1**

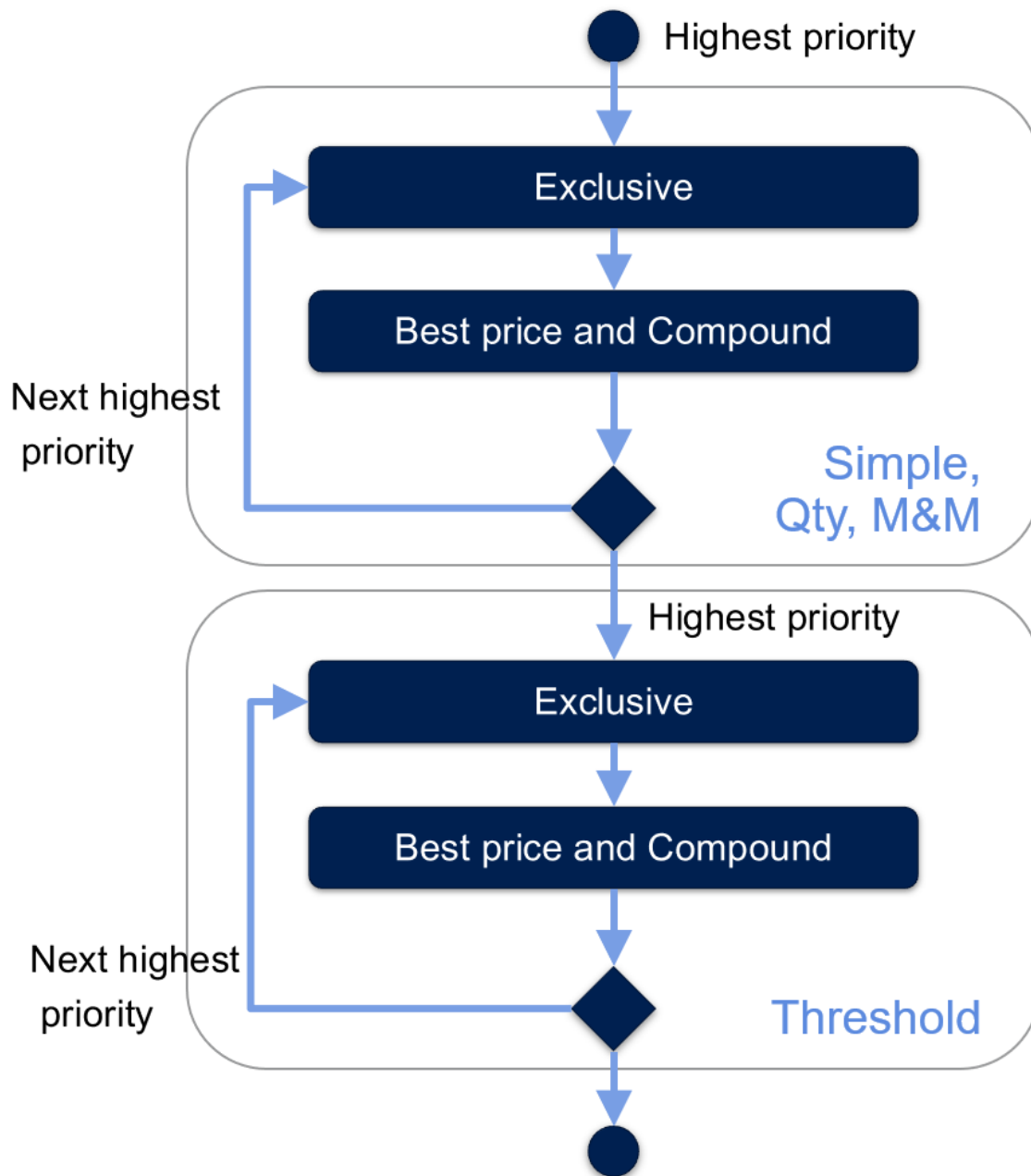
In the first scenario, **Best price and compound within priority, never compound across priorities** is selected as the discount concurrency control model. There are two pricing priorities, and for each pricing priority, there is one discount of each discount type, for example **Simple**, **Mix and Match**, **Quantity**, and **Threshold**. Let's assume there are discounts at two priorities 5 and 10 and all products have multiple discounts at both these priorities. The pool of possible pricing priorities is determined by the price groups and discounts that can be applied to the product.

1. Given that a discount concurrency control model is selected, for each product, the pricing engine next considers the highest pricing priority of the discounts that are applicable on the product. Thus, the pricing engine evaluates and applies the simple, quantity, and mix-and-match discounts with priority 10.

#### **NOTE**

Threshold discounts are not evaluated yet because, as indicated by their name, they will be evaluated against the transaction amount, after all the other discounts have been applied.

The following image shows a concise view of how the pricing algorithm loops through the discounts across various priorities. Note that this diagram applies for both the discount concurrency control models, but the difference is in the way in which the pricing algorithm treats discounts at different priorities. This difference is elaborated using the following example.



2. Within priority 10, the pricing engine first considers the discounts that have the concurrency mode set to **Exclusive**. If there is more than one exclusive discount applicable to the product, then the best exclusive discount is applied. When a product gets an exclusive discount, no other discounts can be applied to this product at any priority.

**NOTE**

Mix-and-match, least-expensive discounts that have the **Multiple occurrences mode** property set to **Favor retailer** are skipped in this step. After all the **Exclusive** discounts (**Simple, Quantity, Mix and Match**) at pricing priority 10 have been applied, then the exclusive mix-and-match **Favor retailer** discounts, at pricing priority 10, are applied to any undiscounted products.

3. Within priority 10, the pricing engine then considers the discounts that have the discount concurrency mode set to **Best price and Compound**. If multiple **Compound** discounts apply to a product, then they are compounded, and the resulting total discount amount competes against the other **Best price** discounts. Either one of the **Best price** discounts or the combination of **Compound** discounts gets applied to the product, depending on which discount gives the most benefit to the customer. Like the previous step, mix-and-match least-expensive discounts that have the **Multiple occurrences mode**

property set to **Favor retailer** are skipped in this step. Once all the **Best price** and **Compound** discounts at pricing priority 10 have been applied, then **Best price** and **Compound** mix-and-match **Favor retailer** discounts are evaluated against each other and gets applied. A **Best price** discount applies only to undiscounted products, but a **Compound** discount applies to undiscounted products and products that are discounted with another **Compound** discounts at the same pricing priority.

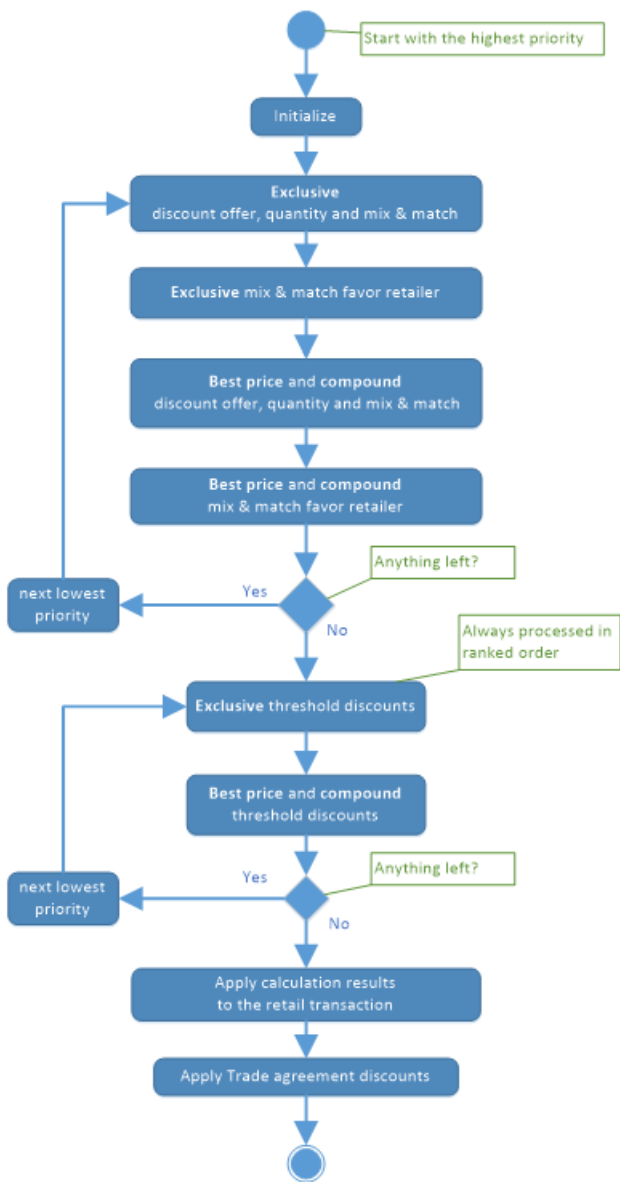
4. Because the discount concurrency control mode is set to **Best price and compound within priority, never compound across priorities** so the simple, quantity, and mix-and-match discounts applicable to the product, at pricing priority 5 do not compete with the applied discounts. At this point, for a product, all the simple, quantity, and mix-and-match discounts at the highest priority have been evaluated.
5. Next, within priority 10, the pricing engine evaluates threshold discounts that have the concurrency mode set to **Exclusive**. An **Exclusive** threshold discount can't be applied to a product that already has a discount applied, so a threshold amount is applied and evaluated only on the undiscounted products. If more than one of these discounts apply to the transaction, the discounts compete, and the largest discount is applied.
6. Next, within priority 10, the pricing engine evaluates threshold discounts that have the concurrency mode set to **Best price** and **Compound**. The pricing engine evaluates and applies threshold discounts at pricing priority 10. **Compound** threshold discounts are compounded with other **Compound** threshold discounts and compete against the other **Best price** discounts within the same pricing priority. A **Best price** threshold discount applies only to undiscounted products, but a **Compound** threshold discount applies to undiscounted products and products that are discounted with another **Compound** (**Simple, Quantity, and Mix and Match**) discount.

#### **NOTE**

If there were threshold discounts set at a higher priority, for example if it's set to 11, and all the other discount types were at priority 10 and 5, then the threshold discounts would have evaluated at priority 11 and then compounded with the simple, quantity, and mix and match discounts at priority 10. This is important because simple, quantity, and mix and match discounts are evaluated within their highest priority and the threshold discounts are evaluated within their highest priority and then compounded. Any threshold discounts at the lower priority are ignored.

At this point, all the discounts at the highest priorities have been evaluated.

The above logic is showcased in the following image, which shows the detailed view of how the pricing algorithm loops through the discounts across various priorities. Note that this diagram applies for both the discount concurrency control models, but the difference is in the way in which the pricing algorithm treats discounts at different priorities.



In this example, let's assume the following setup.

### Product information

PRODUCT #	PRODUCT PRICE
Prod1	\$10
Prod2	\$20
Prod3	\$10

### Discount setup

DISCOUNT #	DISCOUNT CONCURRENCY	PRIORITY	DISCOUNT AMOUNT	DISCOUNT TYPE	APPLICABLE ON PRODUCTS
BP1	Best price	10	15%	Simple, Quantity, or Mix and Match	Prod1, Prod2

DISCOUNT #	DISCOUNT CONCURRENCY	PRIORITY	DISCOUNT AMOUNT	DISCOUNT TYPE	APPLICABLE ON PRODUCTS
BP2	Best price	5	20%	Simple, Quantity, or Mix and Match	All products
C1	Compound	10	\$1	Simple, Quantity, or Mix and Match	Prod1, Prod2
C2	Compound	10	10%	Simple, Quantity, or Mix and Match	Prod1, Prod2
C3	Compound	5	25%	Simple, Quantity, or Mix and Match	All products
C4	Compound	5	10%	Threshold only	All products

**Step 1:** For each product, determine the highest priority where a Simple, Quantity or Mix and Match discount exists. In this case, for prod1 it is priority 10, for prod2 it is priority 10 and for prod3 it is priority 5.

**Step 2:** For each product, find **Simple, Quantity, or Mix and Match** discounts, with discount concurrency as **Exclusive**, at the highest priority applicable to individual products. In this case, there are none for prod1 and prod2 at priority 10 and similarly, there are none for prod3 at priority 5.

**Step 3:** For each product, evaluate **Simple, Quantity, or Mix and Match** discounts, with discount concurrency as **Best price** and **Compound**, at the highest priority applicable to individual products. The following table illustrates this.

**NOTE**

Two asterisks (\*\*) indicate the discount that gets applied to a product.

TRANSACTION QUANTITY	PRODUCT	PRICE	PRIORITY 10 (C1 + C2)	PRIORITY 10 (BP1)	PRIORITY 5 (C3)	PRIORITY 5 (BP2)	TOTAL	EXPLANATION
----------------------	---------	-------	-----------------------	-------------------	-----------------	------------------	-------	-------------

TRANSACTION QUANTITY	PRODUCT	PRICE	PRIORITY 10 (C1 + C2)	PRIORITY 10 (BP1)	PRIORITY 5 (C3)	PRIORITY 5 (BP2)	TOTAL	EXPLANATION
1	Prod1	\$10	\$1.90**	\$1.50	(NA)	(NA)	\$10 – 1.90 = \$8.1	Because the combination of compound discounts is more than the best price discount, C1 and C2 are applied on the product. The discounts at lower priority, such as 5, are ignored.
1	Prod2	\$20	\$2.90	\$3**	(NA)	(NA)	\$20 – 3 = \$17.00	Because the best price discount is more than the combination of compound discounts, BP1 is applied on the product. The discounts at lower priority, such as 5, are ignored.

TRANSACTION QUANTITY	PRODUCT	PRICE	PRIORITY 10 (C1 + C2)	PRIORITY 10 (BP1)	PRIORITY 5 (C3)	PRIORITY 5 (BP2)	TOTAL	EXPLANATION
1	Prod3	\$10			\$2.50**	\$2.0	\$10 – 2.50 = \$7.5	Priority 5 is highest applicable priority for this product. The compounded discount is more than the best price discount, so C3 is applied on the product.

**Step 4:** Evaluate **Threshold** discounts applicable to the individual products at the highest priority. For this example, it is priority 5 for all the products.

TRANSACTION QUANTITY	PRODUCT	DISCOUNT APPLIED	DISCOUNTED PRICE	PRIORITY 5 (C4)	AMOUNT DUE	EXPLANATION
1	Prod1	C1, C2	\$8.1	\$0.81**	8.1 – 0.81 = \$7.29	For Threshold discounts, Priority 5 is the highest applicable priority for this product, so any applicable threshold discounts at priority 5 will be compounded with the applied discounts, if the applied discounts are of discount concurrency mode <b>Compound</b> . Because Prod1 has compound discounts only, the compound threshold discounts can be compounded.



TRANSACTION QUANTITY	PRODUCT	DISCOUNT APPLIED	DISCOUNTED PRICE	PRIORITY 5 (C4)	AMOUNT DUE	EXPLANATION
1	Prod2	BP1	\$17	(NA)	\$17	<p>For Threshold discounts, Priority 5 is the highest applicable priority for this product, so any applicable threshold discounts at priority 5 will be compounded with the applied discounts if the applied discounts are of discount concurrency mode <b>Compound</b>. Because Prod2 has a "Best price" discount, other discounts CANNOT be applied to this product.</p>

TRANSACTION QUANTITY	PRODUCT	DISCOUNT APPLIED	DISCOUNTED PRICE	PRIORITY 5 (C4)	AMOUNT DUE	EXPLANATION
1	Prod3	C3	\$7.5	\$0.75**	7.5 – 0.75 = \$6.75	For Threshold discounts, Priority 5 is highest applicable priority for this product, so any applicable threshold discounts at priority 5 will be compounded with the applied discounts, if the applied discounts are of discount concurrency mode <b>Compound</b> . Because Prod3 has compound discounts only, the compound threshold discounts can be compounded.

The final amount due for Prod1 is 7.29, Prod 2 is 17, and Prod 3 is 6.75.

**Example 2**

In the second scenario, **Best price only within priority, always compound across priorities** is selected as the discount concurrency control model while rest of the discounts remain as is.

1. Given that **Discount concurrency control model** is selected, for each product the pricing engine next considers the highest pricing priority of the discounts that are applicable on a product. If, for a product, discounts from more than one priority are applicable, then each is evaluated independently, and in descending order. Thus, the pricing engine first evaluates and applies the simple, quantity, and mix-and-match discounts with priority 10 followed by the discounts at priority 5.

**NOTE**

Like the previous discount concurrency control model, the **Threshold** discounts are not evaluated yet.

2. Within priority 10, the pricing engine first considers the discounts that have the concurrency mode set to **Exclusive**. If there is more than one exclusive discount applicable to the product, then the best exclusive discount is applied. When a product gets an exclusive discount, no other discounts can be applied to this product at any priority.

#### NOTE

Mix-and-match, least-expensive discounts that have the **Multiple occurrences mode** property set to **Favor retailer** are skipped in this step. After all the **Exclusive** discounts (**Simple, Quantity** and **Mix and Match**) at pricing priority 10 have been applied, then the **Exclusive** mix-and-match **Favor retailer** discounts, at pricing priority 10, are applied to any undiscounted products.

3. Within priority 10, the pricing engine then considers the discounts that have the discount concurrency mode set to **Best price** and **Compound**. As stated before, for this discount concurrency control model, discounts with **Discount concurrency mode** set to **Best price** and **Compound** are all treated as "best price"; within a single pricing priority. So, if multiple **Compound** and **Best price** discounts apply to a product, then all these discounts compete for best price and only the best discount wins within a priority. Like the previous step, mix-and-match least-expensive discounts that have the **Multiple occurrences mode** property set to **Favor retailer** are skipped in this step. When all the **Best price** and **Compound** discounts at pricing priority 10 have been applied, then **Best price** and **Compound** mix-and-match **Favor retailer** discounts are evaluated against each other and applied. Because both **Best price** and **Compound** are treated as best price, only one discount can be applied per product at a given priority.
4. The pricing engine repeats the steps 1 through 3 for any simple, quantity, and mix-and-match discounts at pricing priority 5.

#### NOTE

The pricing engine completes steps 1 through 3, one time, for every pricing priority that applies to the transaction. Therefore, we recommend that you keep the number of pricing priorities to a minimum, based on your business requirements.

At this point, all the simple, quantity, and mix-and-match discounts at all priorities have been evaluated and applied.

5. Next, within priority 10, the pricing engine evaluates threshold discounts that have the concurrency mode set to **Exclusive**. An **Exclusive** threshold discount can't be applied to a product that already has a discount applied, so a threshold amount is applied and evaluated only on undiscounted products. If more than one of these discounts apply to the transaction, the discounts compete, and the largest discount is applied.
6. Next, within priority 10, the pricing engine evaluates threshold discounts that have the concurrency mode set to **Best price** and **Compound**. Because **Best price** and **Compound** are all treated as "best price", these discounts compete for the best discount. The selected threshold discount gets applied to those products which do not have any other types of discounts already applied at priority 10. If there are other discounts, then the threshold discount is not applied because both **Best price** and **Compound** discounts are treated as **Best price**, and only one discount per priority is allowed with this discount concurrency control.

#### NOTE

If there were threshold discounts set at a higher priority, for example, it's set to 11 and all the other discount types were at priority 10 and 5, then the threshold discounts would have evaluated at priority 11 and the best threshold discount would have been applied at priority 11 (assuming there is no **Exclusive** discount (**Simple, Quantity, or Mix and Match**) applied at a lower priority).

7. The pricing engine repeats the steps 5 and 6 for threshold discounts at pricing priority 5.

Let's use the same example as before.

### Product information

PRODUCT #	PRODUCT PRICE
Prod1	\$10
Prod2	\$20
Prod3	\$10

### Discount setup

DISCOUNT #	DISCOUNT CONCURRENCY	PRIORITY	DISCOUNT AMOUNT	DISCOUNT TYPE	APPLICABLE ON PRODUCTS
BP1	Best price	10	15%	Simple, Quantity, or Mix and Match	Prod1, Prod2
BP2	Best price	5	20%	Simple, Quantity, or Mix and Match	All products
C1	Compound	10	\$1	Simple, Quantity, or Mix and Match	Prod1, Prod2
C2	Compound	10	10%	Simple, Quantity, or Mix and Match	Prod1, Prod2
C3	Compound	5	25%	Simple, Quantity, or Mix and Match	All products
C4	Compound	5	10%	Threshold only	All products

**Step 1:** For each product, determine the highest priority where a **Simple, Quantity, or Mix and Match** discount exists. In this case, for prod1 it is priority 10, for prod2 it is priority 10, and for prod3 it is priority 5.

**Step 2:** For each product, find **Simple, Quantity, or Mix and Match** discounts, with discount concurrency as **Exclusive**, at the highest priority applicable to individual products. In this case, there are none for prod1 and prod2 at priority 10 and there are none for prod3 at priority 5.

**Step 3:** For each product, evaluate **Simple, Quantity, or Mix and Match** discounts, with discount concurrency as **Best price** and **Compound**, at the highest priority applicable to individual products. See the table below.

#### NOTE

Two asterisks (\*\*) indicate the discount that gets applied to a product.

TRANSACTION QUANTITY	PRODUCT	PRICE	PRIORITY 10 (C1)	PRIORITY 10 (C2)	PRIORITY 10 (BP1)	TOTAL	EXPLANATION
1	Prod1	\$10	\$1	\$1	\$1.50**	\$10 – 1.50 = \$8.50	Because the compound discounts are treated as the "Best price" for discounts for this discount concurrency control model, the compound discounts will not combine. Rather they all compete for the best discount. Because BP1 gives the highest discount, BP1 gets applied at priority 10.
1	Prod2	\$20	\$1	\$2	\$3**	\$20 – 3 = \$17.00	Same as above. Because BP1 gives the highest discount, BP1 gets applied at priority 10.
1	Prod3	\$10				\$10	No discounts applicable at priority 10.

**Step 4:** For each product, determine the next highest priority where a **Simple, Quantity, or Mix and Match** discount exists. In this case, it is priority 5 for all three products.

**Step 5:** At priority 5, find **Simple, Quantity, or Mix and Match** discounts with discount concurrency mode as **Exclusive**. In this case, none.

**NOTE**

If an exclusive discount existed at priority 5, then it would have been ignored as exclusive discounts cannot co-exist with other discounts which have been applied at a higher priority

**Step 6:** At priority 5, evaluate **Simple, Quantity, or Mix and Match** discounts. The following table illustrates

this.

TRANSACTION QUANTITY	PRODUCT	DISCOUNTED PRICE	PRIORITY 5 (C3)	PRIORITY 5 (BP2)	TOTAL	EXPLANATION
1	Prod1	\$8.50	\$2.13**	\$1.7	$\$8.50 - 2.13 = \$6.37$	Because the discounts across priorities are compounded, the discounts at priority 5 are compounded on the discounts applied at priority 10. C3 gives the highest discount so C3 gets applied at priority 5.
1	Prod2	\$17	\$4.25**	\$3.40	$\$17.00 - 4.25 = \$12.75$	Same as above. Because C3 gives the highest discount, C3 gets applied at priority 5.
1	Prod3	\$10	\$2.5**	\$2.0	$\$10 - 2.5 = \$7.5$	Same as above. Because C3 gives the highest discount, C3 gets applied at priority 5.

**Step 7:** Evaluate **Threshold** discounts.

TRANSACTION QUANTITY	PRODUCT	DISCOUNT APPLIED AT PRIORITY 10	DISCOUNT APPLIED AT PRIORITY 5	DISCOUNTED PRICE	PRIORITY 5 (C4)	AMOUNT DUE	EXPLANATION
----------------------	---------	---------------------------------	--------------------------------	------------------	-----------------	------------	-------------

TRANSACTION QUANTITY	PRODUCT	DISCOUNT APPLIED AT PRIORITY 10	DISCOUNT APPLIED AT PRIORITY 5	DISCOUNTED PRICE	PRIORITY 5 (C4)	AMOUNT DUE	EXPLANATION
1	Prod1	BP1	C3	\$6.37	(NA)	\$6.37	For Threshold discounts, Priority 5 is highest applicable priority for this product. But the threshold discount at priority 5 will only get applied if there is no other discount applied at the priority 5. This is because both best price and compound discounts are treated as "Best price" and only one discount per priority is allowed with this discount concurrency control. So, the threshold discount is ignored.
1	Prod2	BP1	C3	\$12.75	(NA)	\$12.75	Same as above
1	Prod3		C3	\$7.5	(NA)	\$7.5	Same as above

The final amount due for prod1 is 6.37, Prod 2 is 12.75, and Prod 3 is 7.5.

**NOTE**

For the same discount setting, the results vastly differ depending on which discount concurrency control model is selected.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Determine the optimal combination of overlapping discounts

2/18/2021 • 7 minutes to read • [Edit Online](#)

When discounts overlap, you must determine the combination of overlapping discounts that will produce the lowest transaction total or the highest total discount. When the discount amount varies according to the price of the products that are purchased, such as in the common "Buy 1, get 1 X percent off" (BOGO) retail discount, this process becomes an issue of combinatorial optimization.

This article applies to Microsoft Dynamics AX 2012 R3 with KB 3105973 (released November 2, 2015) or later, and to Dynamics 365 Commerce. To determine the combination overlapping discounts to apply in a timely manner, we have introduced a method for applying overlapping discounts. We call this new method **marginal value ranking**. Marginal value ranking is used when the time that is required in order to evaluate the possible combinations of overlapping discounts exceeds a threshold that is configurable on the **Commerce parameters** page. In the marginal value ranking method, a value is calculated for each overlapping discount by using the discount's value on the shared products. The overlapped discounts are then applied from the highest relative value to the lowest relative value. For details about the new method, see the "Marginal value" section, later in this article. Marginal value ranking isn't used when the discount amounts for a product aren't affected by another product in the transaction. For example, this method isn't used for two simple discounts, or for a simple discount and a single product quantity discount.

## Discount examples

You can create an unlimited number of discounts on a common set of products. However, because there is no limit, performance issues can occur when you try to calculate the discounts that should be used on the various products. The following examples illustrate this issue in more detail. In example 1, we start with two products and two overlapping discounts. Then, in example 2, we show how the issue evolves as more products are added.

### Example 1: Two products and two discounts

In this example, two products are required in order to qualify for each discount, and the discounts can't be combined. The discounts in this example are **Best price** discounts. Both products qualify for both discounts. Here are the two discounts.

Discount	Details
Discount 1 - BOGO50%OFF	Buy 1 product get a 2 <sup>nd</sup> product of equal or lesser value at 50% off.
Discount 2 - B2G20%OFF	Buy 2 products and receive 20% off both.

For any two products, the better of these two discounts depends on the prices of the two products. When the price of both products is equal or almost equal, discount 1 is better. When the price of one product is significantly less than the price of the other product, discount 2 is better. Here is the mathematical rule for evaluating these two discounts against each other.

$$\text{if } \frac{\text{Price}_{P1}}{2} > \frac{\text{Price}_{P2}}{3}$$

**NOTE**

When the price of product 1 is equal to two-thirds of the price of product 2, the two discounts are equal. In this example, the effective discount percentage for discount 1 varies from a few percent (when the prices of the two products are far apart) to a maximum of 25 percent (when the two products have the same price). The effective discount percentage for discount 2 is fixed. It's always 20 percent. Because the effective discount percentage for discount 1 has a range that can be more than or less than discount 2, the best discount depends on the prices of the two products that must be discounted. In this example, the calculation is completed quickly, because only two discounts are applied on only two products. There are only two possible combinations: one application of discount 1 or one application of discount 2. There are no permutations to calculate. The value of each discount is calculated by using both products, and the best discount is used.

**Example 2: Four products and two discounts**

Next, we will use four products and the same two discounts. All four products qualify for both discounts. There are twelve possible combinations. In the end, two discounts will be applied to the transaction in one of three combinations: two applications of discount 1, two applications of discount 2, or one application of discount 1 and one application of discount 2. To illustrate the possible combinations, we will look at two different sets of four products that have different prices:

- All four products have the same price, \$15.00. In this case, the best discount combination is two applications of discount 1. Two products will be full price, and two will be 50 percent off. The discounted total for the transaction is \$45 (15 + 15 + 7.50 + 7.50), which is \$15 (25 percent) off the undiscounted total of \$60. Discount 2 is only \$12 (20 percent).
- Two products are \$20 each, one product is \$15, and one product is \$5. In this case, the best discount combination is one application of discount 2 and one application of discount 1. The following tables illustrates the discounts.

To read the tables, use one product from a row and one product from a column. For example, in the table for discount 1, when you combine the two \$20 products, you get \$10 off. In the table for discount 2, when you combine the \$15 product and the \$5 product, you get \$4 off.

Discount 1: Buy 1 Get 1 at 50% off

	\$ 20	\$ 20	\$ 15	\$ 5
\$ 20		\$ 10.0	\$ 7.5	\$ 2.5
\$ 20			\$ 7.5	\$ 2.5
\$ 15				\$ 2.5
\$ 5				

Discount 2: Buy 2 and get 20% off

	\$ 20	\$ 20	\$ 15	\$ 5
\$ 20		\$ 8.0	\$ 7.0	\$ 5.0
\$ 20			\$ 7.0	\$ 5.0
\$ 15				\$ 4.0
\$ 5				

First, we find the largest discount that is available from any two products by using either discount. The two tables show the discount amount for all combinations of the two products. The shaded portions of the tables represent either cases where a product is paired with itself, which we can't do, or a reverse pairing of two products that produces the same discount amount and can be ignored. By looking at the tables, you can see that discount 1 for the two \$20 items is the largest discount that is available for either discount on all four products. (This discount is highlighted in green in the first table.) That leaves only the \$15 product and the \$5 product. By looking at the two tables again, you can see that, for these two products, discount 1 gives a \$2.50 discount, whereas discount 2 gives a \$4 discount. Therefore, we select discount 2. The total discount is \$14. To make this discussion easier to visualize, here are two additional tables that show the effective discount percentage for all possible two-product combinations for both discount 1 and discount 2. Only half the list of combinations is included, because, for these two discounts, the order in which the two products are put into the discount doesn't matter. The highest effective discount (25 percent) is highlighted in green, and the lowest effective discount (10 percent) is highlighted in red.

Discount 1 – BOGO50%OFF

	\$20	\$20	\$15	\$5
\$20	n/a	25.0%	21.4%	10.0%
\$20	25.0%	n/a	21.4%	10.0%
\$15	21.4%	21.4%	n/a	12.5%
\$5	10.0%	10.0%	12.5%	n/a

Discount 2 – B2G20%OFF

	\$20	\$20	\$15	\$5
\$20	n/a	20%	20%	20%
\$20	20%	n/a	20%	20%
\$15	20%	20%	n/a	20%
\$5	20%	20%	20%	n/a

**NOTE**

When prices vary, and two or more discount compete, the only way to guarantee the best combination of discounts is to evaluate both discounts and compare them.

## Total possible combinations

This section continues the example from the previous section. We will add more products and another discount, and see how many combinations must be calculated and compared. The following table shows the number of possible discount combinations as the product quantity increases. The table shows what happens both when there are two overlapping discounts, as in the previous example, and when there are three overlapping discounts. The number of possible discount combinations that must be evaluated soon exceeds what even a fast computer can calculate and compare quickly enough to be acceptable for retail transactions.

Products in the transaction	Two 2-product discounts	Three 2-product discounts
	Possible pair combinations	Possible pair combinations
2	2	3
4	12	27
5	60	135
6	90	315
7	540	1890

When even larger quantities or more overlapping discounts are applied, the total number of possible discount combinations quickly goes into the millions, and the time that is required in order to evaluate and select the best possible combination quickly becomes noticeable. Some optimizations have been done in the price engine to reduce the total number of combinations that must be evaluated. However, because the number overlapping discounts and the quantities in a transaction aren't limited, a large number of combinations will always have to be evaluated whenever there are overlapping discounts. This issue is the issue that the marginal value ranking method addresses.

## Marginal value method

To resolve the issue of an exponentially increasing number of combinations that must be evaluated, an optimization exists that calculates the value per shared product of each discount on the set of products that two or more discounts can be applied to. We refer to this value as the **marginal value** of the discount for the shared products. The marginal value is the average per product increase in the total discount amount when the shared products are included in each discount. The marginal value is calculated by taking the total discount amount (DTotal), subtracting the discount amount without the shared products (DMinus\ Shared), and dividing that difference by the number of shared products (ProductsShared).

$$\frac{(D_{Total}) - (D_{Minus Shared})}{(Items_{Shared})} = \text{Marginal value}_D$$

After the marginal value of each discount on a shared set of products is calculated, the discounts are applied to the shared products in order, exhaustively, from highest marginal value to lowest marginal value. For this method, all remaining discount possibilities aren't compared every time after a single instance of a discount is applied. Instead, the overlapping discounts are compared one time and then applied in order. No additional comparisons are done. You can configure the threshold to switch to the marginal value method on the **Discount** tab of the **Commerce parameters** page. The acceptable time to calculate the total discount varies across retail industries. However, this time generally falls in the range of tens of milliseconds to one second.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Define channel-specific discounts

2/18/2021 • 2 minutes to read • [Edit Online](#)

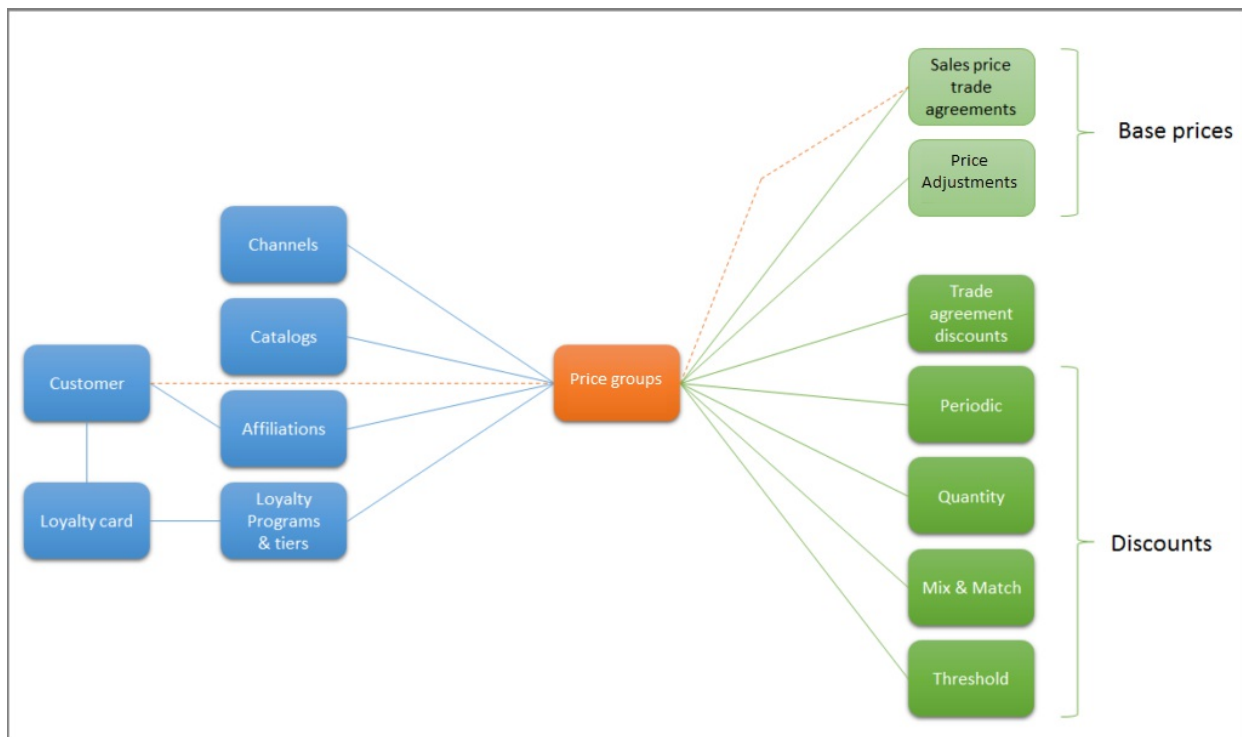
This topic reviews the concepts you need to know to create a discount for a specific channel.

## Channel-specific discounts

Retailers often offer different discounts in different channels. This may be done to address local market conditions or to deal with competing retailers.

Commerce uses price groups to define channel-specific discounts. Price groups can be assigned to one or more of the following entities: channels, catalogs, affiliations, and loyalty programs. This article discusses channels, but the same concepts apply to catalog discounts, affiliations discounts, and loyalty discounts.

## Price groups



The diagram above illustrates the relationship between entities that may be on a transaction (channel, catalog, affiliation, customer, loyalty card) and the various discount types that can be configured. All transactions occur in a channel, so the channel is guaranteed to be present on a transaction. The remaining entities are optional. On each master data page there is a link to a related price groups page where you can view and add price groups as needed. A price group is used to relate four different types of entities to discounts, price adjustments, and trade agreements. We recommend that you plan a strategy for how you will name your price groups to keep them organized. One option would be to use a letter or number prefix or suffix to distinguish between the different types. For example, 1-xxxx for channel price groups and 2-xxxx for catalog price groups. There are four inquiry pages that focus on each of the commerce entities that can have discounts associated to them.

- **Channel channel price groups** – This page shows a list of channels and discounts linked together for each price group.
- **Catalog price groups** – This page shows a list of catalogs and discounts linked together for each price group.

- **Loyalty price groups** – This page shows a list of loyalty programs and discounts linked together for each price group.
- **Affiliation price groups** – This page shows a list of affiliations and discounts linked together for each price group.

## Example channel discount set up

The following example illustrates the tasks involved in setting up a channel discount.

1. For this example, you have a channel called **Houston**, and you're going to create a new discount called **Back-to-School**.
2. Because the pricing and discount strategy includes the possibility of channel discounts, you always create a channel-specific price group when you create a channel.
3. You have the price group **Houston-PG** and it is assigned to the **Houston** channel.
4. After you create the new **Back-to-School** discount, you need to click **Price groups** on the top of the **Discount** page. The **Discount price groups** page will open. Next, click **New** and select the **Houston-PG** price group.
5. Now you can enable the discount and push it to the channel.

## Additional resources

[Price adjustments and discounts](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Options for preventing discounts for retail products

2/18/2021 • 2 minutes to read • [Edit Online](#)

There are various reasons why retailers may want to prevent some products from being discounted, either from a promotion or during the sale at the POS.

The following options, which can be found on the **Commerce** tab of released products, will allow the product to be configured to prevent all or manual discounts. The settings can also be specified at the category level from the category hierarchy.

- **Prevent all discounts** – Select this option to prevent all types of discounts from being applied to this product. This includes promotions such as mix and match, quantity and threshold discounts, as well as manual line and transaction discounts that are applied during a sale by a POS user.
- **Prevent manual discounts** – Select this option to only prevent the manual line or transaction discounts that are applied during a sale by a POS user. Products with this option selected are still eligible for promotions, such as mix and match and quantity and threshold discounts.

## NOTE

These settings do not restrict the price override operation, because that sets the base price and is not treated as a discount.

Commerce ^

<b>VARIANT</b>	<b>BAR CODE</b>	<b>PRODUCT LIFE CYCLE</b>
Color group <input type="text"/>	Bar code setup <input type="text"/>	Season <input type="text"/>
Size group <input type="text"/>	Bar code <input type="text"/>	Valid from <input type="text"/>
Style group <input type="text"/>	<b>ACTIVATION</b>	Valid to <input type="text"/>
Print variants shelf labels No <input checked="" type="checkbox"/>	Issue date <input type="text"/>	<b>POS REGISTERS</b>
<b>COMPARISON PRICE</b>		Scale product No <input checked="" type="checkbox"/>
Base comparison unit <input type="text"/>		Key in price <input type="text" value="Not mandatory"/>
Comparison price <input type="text" value="69.99"/>		Key in quantity <input type="text" value="Not mandatory"/>
<b>BLOCKING</b>		Must key in comment No <input checked="" type="checkbox"/>
Blocked at register No <input checked="" type="checkbox"/>		Zero price valid No <input checked="" type="checkbox"/>
Date blocked <input type="text"/>		Quantity becomes negative No <input checked="" type="checkbox"/>
Date to be blocked <input type="text"/>		<b>Prevent all discounts</b> No <input checked="" type="checkbox"/>
		<b>Prevent manual discounts</b> No <input checked="" type="checkbox"/>

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Tender-based discounts

2/18/2021 • 5 minutes to read • [Edit Online](#)

It's a common practice among retailers to release private, branded credit cards. The retailers benefit because they get preferred rates from the banks. Additionally, because these credit cards can encourage customers to visit the store more often, they help improve the retailer's bottom line. Therefore, retailers have an incentive to increase customer use of their branded credit cards. To achieve this goal, they often provide additional discounts to customers who use these credit cards.

Alternatively, retailers who don't provide branded credit cards might want to encourage customers to pay by using other tender types, such as cash, gift cards, or loyalty points. In this way, they can help reduce the expense of credit card processing fees. Therefore, retailers might provide discounts to customers who use these alternative tender types.

In Microsoft Dynamics 365 Commerce, retailers can configure a discount percentage that is applied to qualified lines if the customer pays by using the preferred tender type. The customer can decide whether to do a partial payment or a full payment, and Commerce determines the appropriate discount amount. Note that the discount is always given on the pre-tax amount of the qualified items.

Tender-based discounts don't compete with item-based discounts, such as periodic or manual discounts. They are always compounded over the item discounts. Therefore, even if an exclusive periodic discount is applied to an item, the tender-based discount is still applied on top of the exclusive periodic discount. Likewise, if a threshold discount is applied to the transaction, and the tender-based discount reduces the total below the threshold, the threshold discount is still applied to the transaction.

Even though tender-based discounts reduce the subtotal of the transaction, automatic charges that are applied to the transaction aren't affected. For example, if the delivery charges are calculated as \$5 because the subtotal was more than \$100, and the tender-based discount reduces the amount so that it's less than \$100, the delivery charges are still \$5 for the order.

## NOTE

Tender-based discounts are proportionally distributed to the qualified sales lines and reduce the pre-tax amount of the individual lines. If multiple tender-based discounts are configured for a tender type (for example, cash), only the best tender-based discount is applied.

Tender-based discounts can be applied only to sales lines where the prices aren't locked. If new sales lines are added to an order, the tender-based discount is applied to the new sales lines only during payment. While a customer order for pickup or shipment is being placed, the tender-based discount is applied only to the deposit amount. After the order is placed, during fulfillment, the prices of the sales lines are locked. Therefore, no tender-based discount is applied to any balance that is paid during pickup or authorized during shipment. The tender-based discount can be applied to the whole amount of a customer order only if the retailer collects the whole amount as a deposit while the order is being placed.

## IMPORTANT

In Commerce, tender-based discounts are currently limited to two payment types: credit cards and cash.

## POS user experience

If the tender-based discount is set up for cash, and the cashier at the point of sale (POS) selects a button that is mapped to the Pay cash operation, the tender-based discount is automatically applied to the transaction. The reduced amount is then shown as the balance. However, if the cashier selects the **Back** button on the payment screen, the discount is removed, and the original amount is shown on the transaction screen. The tender-based discount is removed if the payment line is voided.

For cards payments, retailers can set the tender-based discount on one or more types of credit cards. However, the system can't verify the type of credit card that is used unless the card is authorized. If the discount is applied after authorization, the payment authorization will be for a larger amount, but the payment capture will be for a smaller amount.

To help prevent this situation, if a customer pays with a credit card, the cashier sees a dialog box that lists credit cards that will bring the customer additional savings. The cashier can then ask whether the customer wants to use one of the preferred cards to get an additional discount. If the cashier uses a preferred card, the tender-based discount is applied to the transaction, and the reduced amount is shown on the payment screen. The authorization will be for the reduced amount. If the customer inserts a card that differs from the card that the cashier selected, an error message is shown, and the authorization is voided.

## Call center user experience

When the user selects **Complete** during a call center order, the **Totals** screen is shown. At first, the totals on this screen don't include tender-based discounts, because the payment method hasn't yet been selected. On the **Add payment** screen, if the user selects the payment method that the tender-based discount is configured for, the payment amount is automatically adjusted so that it reflects the discounted amount. Like the customer at the POS, the call center customer can decide whether to pay the full payment or a partial payment. Based on the amount that is paid, the tender-based discount is applied to the sales order.

### NOTE

Card validation isn't done for call center orders. For example, if the call center user selects Visa as the credit card, but the customer uses Mastercard, the system still applies the discount.

## Exclude items from discounts

Retailers often choose to exclude some products, such as new items or in-demand items, from discounts. However, they might still want to apply tender-based discounts. For example, a retailer configures Commerce so that it doesn't allow item-based discounts or manual discounts. However, if the customer pays by using the preferred tender, Commerce still applies the tender-based discount. To set up Commerce in this manner, retailers must go to **Product information management > Products > Released products**, select the item, and then, on the **Commerce FastTab**, set the **Prevent all discounts** and **Prevent tender based discounts** options to **No**, and the **Prevent discounts** and **Prevent manual discounts** options to **Yes**.

### NOTE

When the **Prevent all discounts** configuration is set to **Yes**, no discounts will be applied to the product. Not even tender-based discounts will be applied.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

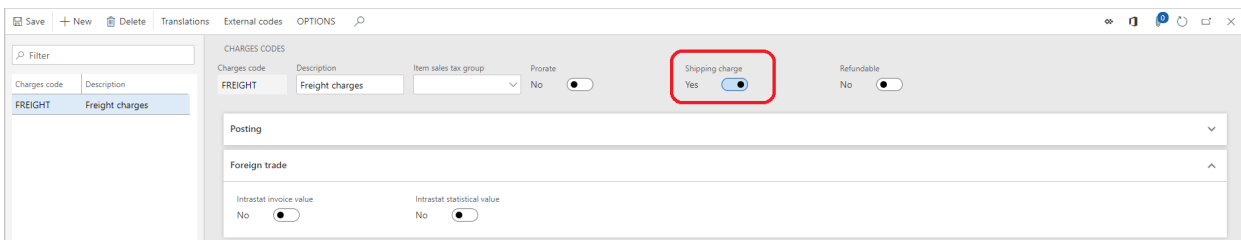
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Shipping discount overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

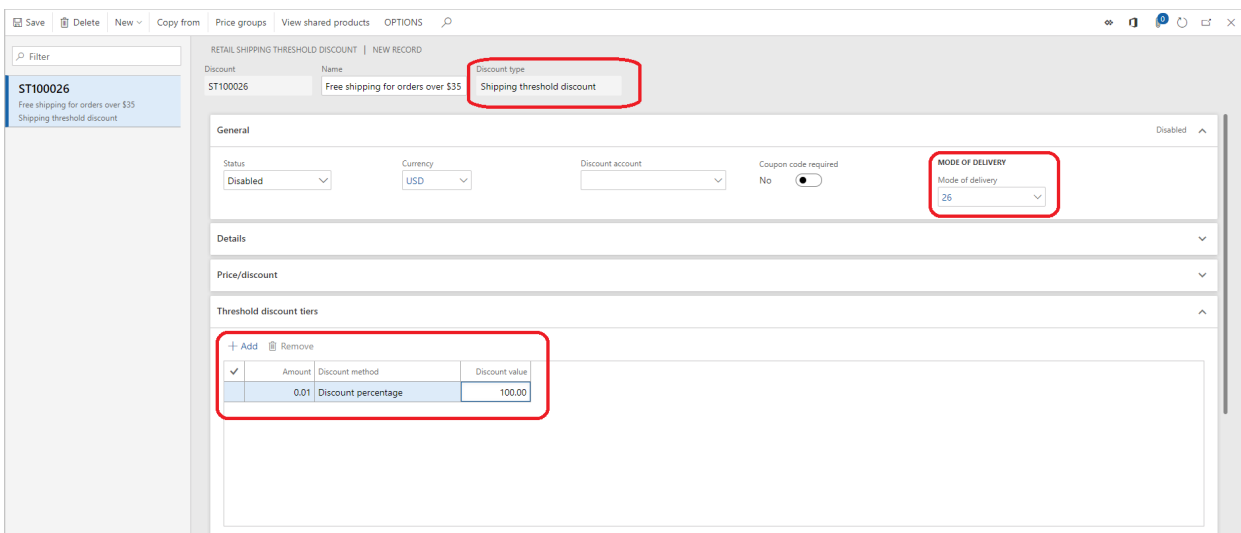
This topic provides an overview of the shipping promotion capability available within Dynamics 365 Commerce. Free or discounted shipping is one of the highly influencing factors driving the customers' online purchase decisions. Many retailers also leverage the free shipping benefit to motivate the customers to increase their basket size, thus increasing the revenue per transaction. With the 10.0 release of Retail, retailers can use "Retail shipping threshold discount" to define the thresholds, which once met, will qualify the customers for discounted or free shipping. For example, spend \$50 or more to get free 'Overnight shipping' or sign up for the loyalty program and get free 'Two-day shipping'.

This feature leverages the advanced auto charges capability that was available in the call center and e-Commerce modules but has now been made available in POS. For more information, see [Omni-channel advanced auto charges](#). These advanced auto charges need to be enabled for shipping promotion to work. These can be enabled by turning on the "Use advanced auto-charges" configuration on the **Commerce parameters > Customer orders** tab. Retailers can use the advanced auto charges feature to set various types of charges such as handling, installation, and disposal, however, the shipping discount is only applied to the shipping charges. Thus, the retailer needs to specify which of the charges are shipping charges. To specify a shipping charge, go to **Retail and Commerce > Retail and Commerce IT > Channel setup > Charges > Charge codes**. Select the **Shipping charge** check box for the desired charges. This is the only prerequisite for using the shipping threshold discount.



The screenshot shows the 'Charge codes' configuration page. The 'Shipping charge' checkbox is highlighted with a red box. The 'Refundable' checkbox is also visible. The 'Posting' and 'Foreign trade' sections are expanded, showing 'Intrastat invoice value' and 'Intrastat statistical value' options, both set to 'No'.

The next step is to configure the shipping discount itself. To do this, go to **Retail and Commerce > Pricing and discounts > shipping discounts > Shipping threshold discount**. You can define the thresholds, set the discount percent that should be applied when the thresholds are met, and choose a mode of delivery for which this discount applies, such as Standard overnight or Two-day shipping.



The screenshot shows the 'Retail shipping threshold discount' configuration page. The 'Discount type' is set to 'Shipping threshold discount'. The 'MODE OF DELIVERY' is set to '26'. The 'Threshold discount tiers' section is expanded, showing a table with one tier:

Amount	Discount method	Discount value
0.01	Discount percentage	100.00

Like product discounts, this discount honors all the existing standard discount capabilities, such as allowing the retailer to restrict these discounts with coupons so that only the customers with coupons can get these

discounts. Also, these discounts leverage the Price groups capability to determine the eligibility of the discount. For example, the retailer can choose to run these promotions only in the online channels and/or across channels for certain customer groups such as loyalty customers. Lastly, to view the charges applied on the sales lines and the applied promotion, you need to add **Manage charges** on the POS screen. Go to the **Screen layout** page to make those changes. Now run the jobs 1020, 1040, 1090, and 1110 to send the charges, shipping promotion, and screen layout information to the channels.

When the cashier creates a customer order on POS or the customer places an order on an e-Commerce website, the charges are calculated automatically. However, if the mode of delivery on the order meets the mode of delivery and the transaction amount meets the threshold set on the shipping threshold discount, then the shipping discount gets applied. Currently, the shipping discounts apply only on POS and e-Commerce orders. These discounts will also be available for call center orders in a future release of the app.

**NOTE**

Unlike product discounts such as quantity, simple, mix and match, and threshold discounts, the shipping discount does not create discount lines. Instead, the shipping discount edits the shipping charge directly and appends the name of the discount to the charge description.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up coupons for retail sales

2/18/2021 • 5 minutes to read • [Edit Online](#)

## Overview of coupons

Coupons are codes and bar codes that are used to add discounts to transactions. Each coupon can have multiple codes, and each code can have its own effective dates.

Each coupon is related to one discount. The price groups that are associated with the discount define the customers that can use a coupon or the channels where a coupon is valid.

Essentially, coupons are additional validation on top of discounts. The coupon provides the coupon codes and bar codes that are required, together with date ranges for those codes. The coupon also provides optional usage limits and customer required properties. The discount provides the set of products that the coupon is valid for. The price groups for the discount provide the set of customers, channels, or catalogs that the coupon is valid for.

To create a coupon, you create the discount and the coupon separately. You then link them by selecting the discount on the coupon page in Commerce.

### NOTE

After a coupon is linked to a discount, several fields on the discount page in Commerce become read-only, because they are managed by the coupon's settings. These fields include the fields for the status and standard date ranges.

### Limited-use coupons

Coupons can be configured as limited-use coupons. The usage limit can be defined per customer or channel, or as a global limit. This limit is enforced when the code or bar code is entered or scanned in POS or during sales order entry.

The limit is enforced per coupon code on a coupon. For example, a single-use coupon that has two coupon codes can be used two times: one time for each coupon code. Each code on a coupon can be independently set to active.

The coupons can be used across any selling channel however, for call center orders, the limited use coupons can be used for only those call centers orders where the **Order completion** setting on the call center is enabled. If this is not enabled, then only non-limited use type coupons can be used in call center orders.

### NOTE

After a coupon code has reached its usage limit, the system does *not* automatically change the status of the coupon code to "Used". However that coupon code has reached its usage limit and cannot be used. If the status of a coupon code is manually set to anything other than **Active**, then this coupon code cannot be used in any channel.

## Managing coupons

You must create the discount and the coupon separately. You then link them by selecting the discount on the coupon page. After you link a coupon to a discount, several fields for the discount become read-only, because they are managed by the coupon's settings. These fields include the fields for the status and standard date ranges.

Essentially, coupons are now additional validation on top of discounts. The coupon provides the coupon codes

and bar codes, together with date ranges, the usage limits, and the customer required property. The discount provides the set of products that the coupon is valid for. The discount's price groups provide the set of customers, channels, or catalogs that the coupon is valid for.

## System setup for coupons

Before you can set up a coupon, you must set up the coupon bar code and two coupon number sequences.

1. On the **Mask characters** page, create a new mask character for the coupon code. You can select any unused character.
2. On the **Bar code mask setup** page, create a new bar code mask. Set the **Type** field to **Coupon**.
3. On the **Bar code setup** page, create a new bar code that uses the bar code mask that you just created.
4. On the **Number sequences** page, create two new number sequences. One sequence is for the coupon code ID, and the other sequence is for the coupon number. The coupon code ID is the unique identifier for each coupon code for a coupon. The coupon number is the unique identifier for a coupon. Each coupon can have multiple codes and bar codes that trigger the coupon.

### NOTE

For both number sequences, you must set the **Scope** field to **Company**. In most cases, you should automatically generate both sequence numbers.

5. On the **Commerce parameters** page, on the **Bar codes** tab, select the bar code that you created earlier.
6. On the **Commerce shared parameters** page, on the **Number sequences** tab, select the number sequences that you created for the coupon number and coupon code ID.
7. You can now open the **Coupons** page and create new coupons.

## The effect of partial updates on coupons

Coupon functionality comprises multiple distinct features. Commerce Headquarters (HQ) and the channel can be partially updated across components. Therefore, it's important that you understand how partial updates affect coupon functionality as a whole.

- **HQ is partially updated, but Commerce Scale Unit and POS aren't updated.** In an HQ update, the coupon and discount pages are updated, and the commerce price engine is also updated. If only one of those two components is updated, some pages in Commerce won't match the price calculation data. Therefore, unexpected discount calculations or errors might occur during discount calculations.
- **HQ is updated, but Commerce Scale Unit and POS aren't updated (N-1).** Because not all stores can be updated at the same time, we recommend that you update HQ before you update stores. In the N-1 scenario, new functionality that is related to coupons won't be available in stores that haven't been updated yet. For example, the coupon functionality introduces "exclude" lines. If you use exclude lines on a discount, they won't be applied in a store that is running an earlier version.
- **HQ isn't updated, but Commerce Scale Unit and POS are updated (N+1).** Because the updated price engine in Commerce Scale Unit can handle legacy discount codes during price calculations, the update should have no functional impact in this scenario.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Show discounts in POS

2/18/2021 • 4 minutes to read • [Edit Online](#)

Promotions play an important role in motivating customers who are making purchasing decisions. For example, holidays can produce the highest number of sales for retailers, because the whole retail market is flooded with enticing promotions and discounts. If store associates know about and understand the promotions that are available, they can easily take advantage of those promotions to cross-sell and upsell items. This topic explains how Microsoft Dynamics 365 Commerce helps sales associates learn about promotions and how they can be used for cross-sell and upsell motions.

## Learn about store discounts

Commerce includes an operation that is named "View all discounts." This operation shows all the discounts that are currently running in a store. The "View all discounts" operation can be mapped to a button in the point of sale (POS), and that button can be added to the **Welcome** page or the **Transaction** page. The following illustration shows an example of the **All discounts** page that is opened.

NAME	DISCOUNT TYPE	COUPON REQUIRED	START DATE	END DATE
30% off select items	Discount	No	12/31/1899	08/06/2020
Take 20 off anything	Discount	Yes	12/31/1899	12/30/2154
Watches	Discount	No	12/31/1899	12/30/2154
Buy 2 or more and get 25% off	Multiple buy	No	12/31/1899	12/30/2154
Loyalty 20% off over \$300	Threshold discount	No	12/31/1899	12/30/2154
Loyalty 50% off sunglasses with a Sw...	Mix and match	No	12/31/1899	12/30/2154
10% Student discount on jeans	Discount	No	01/01/2020	12/30/2154

**Details**

**ADDITIONAL RESTRICTIONS**

**DESCRIPTION**  
30% off on "Suede Leather Slip-on Loafer" on Salmon color only

To show discounts, the system looks for all the discounts that match one or more of the following conditions:

- The price group of the discount matches the price group of the store.
- The price group of the discount is mapped to an affiliation or loyalty program.
- The price group of the discount is mapped to a catalog that is associated with the store.

The **All discounts** page shows only some coupon-based discounts, because retailers typically create thousands of coupons and corresponding discounts for unique customers, and this page isn't intended to show customer-specific discounts. Coupon-based discounts are shown only if the **Apply without a coupon code** option is turned on in each coupon header. In that case, cashiers can apply the coupon without having to enter or scan any coupon code or bar code.

When the **Apply without a coupon code** option is turned on, various scenarios become available. For example, cashiers can give additional discounts to customers for customer appeasement purposes or because of product defects. Printed coupon codes or bar codes don't have to be distributed to cashiers. Instead, cashiers can



select the **Apply coupon** button. The coupon is then automatically applied to the transaction. If multiple coupons exist for a coupon header, the system automatically selects the first active coupon on the transaction.

On the **All discounts** page, sales associates can also search discounts by keywords. The keyword search looks in the fields that hold the discount name and discount description. Sales associates can also filter discounts based on whether a discount requires a coupon code.

## Cross-sell and upsell by using discounts

Multiline discounts, such as quantity discounts, mix-and-match discounts, and threshold discounts, are a great way to motivate customers to buy more products to get larger discounts. Therefore, they also help increase the size of a customer's cart and retailer revenue. These discounts can be publicized on e-commerce websites, on social media, and on banners in the store.

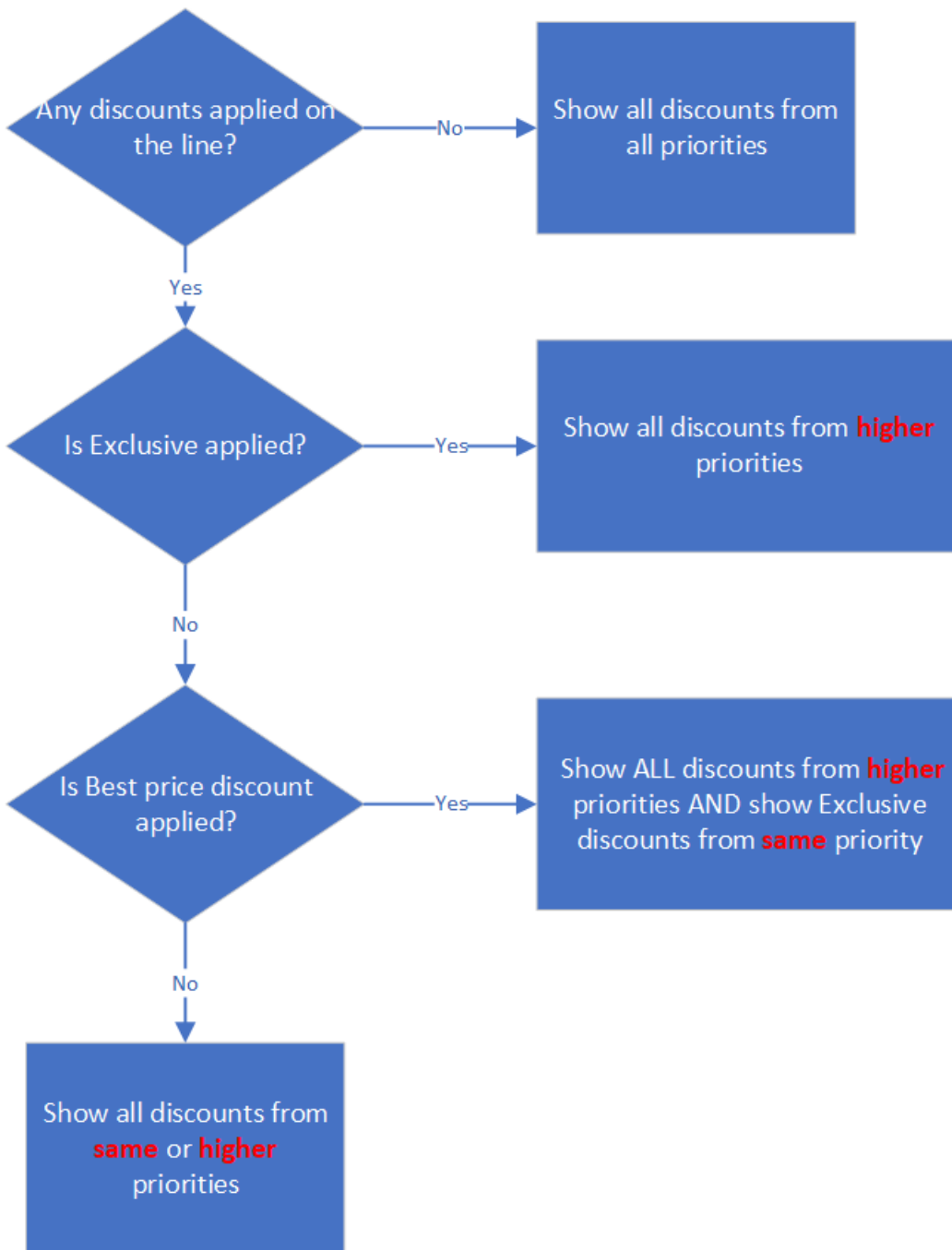
However, even when all these publicity methods are used, customers might miss the opportunity to take advantage of promotions. To make it easy for sales associates to learn what promotions are applicable to a selected line, or even to the whole cart, retailers can add the button for the "**View available discounts**" operation to the button grid on the **Transaction** page. As a result, sales associate can select a transaction line and then select the button to show all the discounts that are available for the selected line. The sales associate can also select another tab to show discounts that apply to the whole transaction. It is important to note that **View available discounts** does not show the discounts that are already applied on the sales line because the discount information is already shown on the sales line. The purpose of this scenario is to only show the discounts that are not yet applied. The exception to this is the discounts that are applied based on a coupon marked as "Apply without a coupon code". This makes it easy for the sales associate to easily remove the coupon they have applied.

The **All discounts** page shows only discounts that don't compete with any of the applied discounts. This behavior helps ensure that, if a sales associate informs a customer about a discount, and the customer takes the required action (for example, the customer buys one more item to get 10 percent off), the discount is applied to the transaction. The coupon-based discounts are shown only when the **Apply without a coupon code** option is turned on.

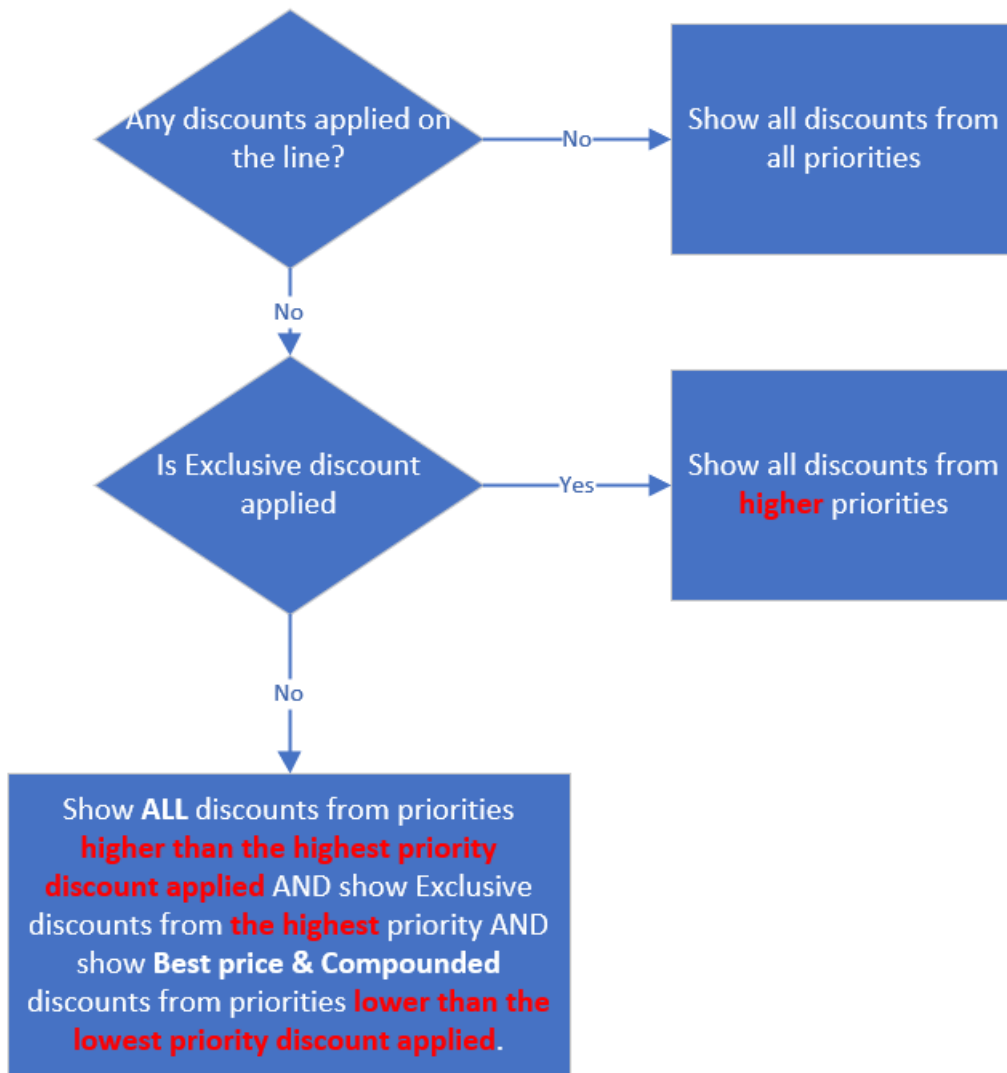
In a simple scenario where all discounts have the same priority, the discount concurrency mode is **Compounded**, and the discount concurrency control is set to **Best price and compound within priority, never compound across priorities**, the **All discounts** page shows all available discounts for a product, because all the discounts are compounded and don't compete with each other.

The following illustrations show the logic that determines which discounts are shown in advanced scenarios, such as a scenario where the discount concurrency mode is **Best price** or **Exclusive**, and two or more priorities are used. In these scenarios, the discounts that are shown are further affected based on whether the discount concurrency control is set to **Best price and compound within priority, never compound across priorities** or **Best price only within priority, always compound across priority**.

The following illustration shows the logic that is used when the discount concurrency control is set to **Best price and compound within priority, never compound across priorities**.



The following illustration shows the logic that is used when the discount concurrency control is set to **Best price only within priority**, always compound across priority.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail price reports

2/18/2021 • 2 minutes to read • [Edit Online](#)

In order to provide competitive prices to their customers, retailers often change prices of products. Store managers want the ability to easily access recent or upcoming price changes so that they can plan for the required resources to update the price labels displayed on the store shelves. With release 10.0 of Retail, a store manager can open the **Price** report by navigating to **All stores > Store > Price report** and viewing the updated prices for the products associated to the store.

To enable the price report, the **Enable price report for store** parameter must be turned on. This parameter is located on the **Commerce parameters > Discounts > Miscellaneous** tab. Opening the **Price report** page displays a dialog box with various configurations. The available configurations are listed below.

CONFIGURATION	DESCRIPTION
From date / To date	The date range for which the price report should be generated. The duration is currently limited to 7 days.
Channel	The store for which the price report should be generated.
Display products with available inventory	Setting this to <b>Yes</b> will show the prices for only those products which currently have physical inventory available in the store.
Display prices for variants	Setting this to <b>Yes</b> will display the prices of the variants along with the product masters. This should only be turned <b>On</b> if you have variant-specific prices, because the number of rows grows very large. In future releases, we will enable the dimensions-based prices so that the store manager can choose the dimensions for which the prices should be displayed.
Display products with price changes	Setting this to <b>Yes</b> will display the prices for only those dates on which the price has been changed. The price for <i>one day before</i> the selected <b>From date</b> will always be displayed, so that the store manager can easily identify the products which have not changed prices for the entire selected duration, and can also view the current price.

After the report is generated, the Excel file can be downloaded for any additional filtering needs. The price report can also be used to check the historical prices of products for past dates.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Base price and trade agreements

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through creating channel-specific sales price trade agreements. This procedure uses the USRT demo data company.

1. In the **Navigation pane**, go to **Modules > Retail and Commerce > Pricing and discounts management > Price groups > All price groups**. Price groups are how trade agreements are assigned to specific channels. Using price groups to assign trade agreements to a channel enables channel-specific pricing.
2. Click **New**.
3. In the **Price groups** field, type a value.
4. In the **Name** field, type a value.
5. Click **Save**.
6. Close the page.
7. In the **Navigation pane**, go to **Modules > Retail and Commerce > Channels > Stores > All stores**.
8. In the list, select 'New York'
9. On the Action Pane, click **Store**.
10. Click **Price groups**.
11. Click **New**.
12. In the **Price groups** field, click the drop-down button to open the lookup.
13. In the list, find and select the desired record.
14. Click **Save**.
15. Close the page.
16. Close the page.
17. In the **Navigation pane**, go to **Modules > Retail and Commerce > Products and categories > Released products by category**.
18. In the list, click the link in the selected row.
19. Click **Edit**.
20. Expand the **Sell** fastTab.
21. In the **Price** field, enter a number. This price is used if no applicable trade agreements are found.
22. Click **Save**.
23. On the **Action Pane**, click **Sell**.
24. Click **Create trade agreements**.
25. Click **New**.
26. In the **Name** field, click the drop-down button to open the lookup.
27. In the list, select **Commerce**. In the demo data, the **Commerce** journal name has the default relation of **Price (sales)**. That means all new lines created will default to sales price trade agreements.
28. On the **Action pane**, click **Lines**.
29. In the **Party code type** field, select 'Group'.
30. In the **Account selection** field, click the drop-down button to open the lookup.
31. In the list, find and select the desired record. This will complete the link from Channel to Price group to Trade agreement.
32. In the **Item relation** field, type a value.
33. In the **Amount in currency** field, enter a number.

34. In the **Details** fastTab, check or uncheck the **Find next** checkbox. When **Find next** is set to 'Yes', the pricing engine will continue to search for applicable trade agreements with a lower sale price. When **Find next** is set to 'No', the price engine stops searching and uses the trade agreement.
35. Click **Post**.
36. Click **OK**.
37. Close the page.
38. On the **Action Pane**, click **Sell**.
39. Click **Sales price**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Category pricing rules to create trade agreements

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates how to create sales price trade agreements using a category pricing rule. The demo data company used to create this task is USRT. This task is intended for the Commerce merchandising manager role.

1. Click Pricing and discount management.
2. Click New.
3. Click Category price rule.
4. In the list, mark the selected row.
5. In the Account code field, select an option.
  - A "Group" type account code is used to set up sales price trade agreements that are specific for Channels, Loyalty programs, Catalogs, and Affiliations.
6. In the Account selection field, enter or select a value.
7. In the Category field, enter or select a value.
8. In the Amount/Percent field, enter a number.
9. In the Rounding version field, enter or select a value.
10. Click Generate trade agreements.
11. Click Next.
12. In the From date field, enter a date.
13. In the To date field, enter a date.
14. Select Yes in the Find next field.
15. Click Next.
16. Click Finish.
  - This creates a Trade agreement journal and opens it for your review.
17. In the list, find and select the desired record.
  - The trade agreement journals created from the Category pricing rules aren't posted. You can review and edit the prices generated before posting them.
18. Click Edit.
19. In the Amount in currency field, enter a number.
20. Click Post.
21. Click OK.
22. Close the page.
23. Close the page.
24. Click the Category price rules tab.
  - Channel specific Category pricing rules will show in this list.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product recommendations overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

Microsoft Dynamics 365 Commerce can be used to show product recommendations on the e-Commerce website and point of sale (POS) device. Product recommendations are items that a customer might be interested in. The recommendations are based on the purchase trends of other customers in online and brick-and-mortar stores.

Product recommendations allow customers to easily and quickly find products that they want while they have an experience that serves them well. Cross-selling and upselling can even be used to assist customers find additional products that they didn't originally intend to buy. When recommendations are used to enhance product discovery, they create more conversion opportunities, help increase sales revenue, and even amplify customer satisfaction and retention.

In e-Commerce, product recommendations are powered by Microsoft Recommendations machine learning technologies on a large scale.

## Recommendation service

The product recommendations service utilizes artificial intelligence and machine learning (AI-ML) technologies in the following way:

- Data in the format that the Recommendation service requires is extracted from the Commerce operational database and sent to Azure Data Lake Storage or Entity store.
- The recommendations service uses the stored data to train recommendation models for the **People also like**, **Frequently bought together**, **New**, **Best selling**, and **Trending** lists.

## Scenarios

Product recommendations are available for the following scenarios:

- **On any store page for browsing or landing page in e-Commerce:** If customers or store associates visit a store page, the recommendation engine can suggest products in the **New**, **Best Selling**, and **Trending** lists.
- **On the Product details page:** If customers or store associates visit a **Product details** page, the recommendation engine suggests additional items that are also likely to be purchased. These items appear in the **People also like** list.
- **On the Transaction page or the checkout page:** The recommendation engine suggests items, based on the whole list of items in the basket. These items appear in the **Frequently bought together** list.
- **Personalized recommendations:** Merchandizers can provide signed-in customers a personalized **picks for you** list, in addition to new functionality that allows for existing list scenarios to be personalized based on that customer. To learn more, see [Enable personalized recommendations..](#)

### Types of product recommendations

The following table describes various types of automated product recommendations available for retailers to implement in their Dynamics 365 Commerce solution via the [product collection module](#). Retailers can also show personalized results for a signed-in user if the site author chooses that option.



PRODUCT COLLECTION MODULE	TYPE	DESCRIPTION
New	Algorithmic	This module shows a list of the newest products that have been recently assorted to channels and catalogs.
Best selling	Algorithmic	This module shows a list of products that are ranked by the highest number of sales.
Trending	Algorithmic	This module shows a list of the highest-performing products for a given period, ranked by highest number of sales.
Frequently bought together	AI-ML	This module recommends a list of products that are commonly purchased together with the contents of the consumers current cart.
People also like	AI-ML	This module recommends products for a given seed product based on consumer purchase patterns.
Picks for you	AI-ML	This module recommends a personalized list of products based on purchase patterns of the signed-in user. For a guest user, this list will be collapsed.

## Additional resources

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to enable and test Azure Data Lake Storage for a Dynamics 365 Commerce environment, which is a prerequisite for enabling product recommendations.

## Overview

In the Dynamics 365 Commerce solution, all product and transaction information is tracked in the environment's Entity store. To make this data accessible to other Dynamics 365 services, such as data analytics, business intelligence, and personalized recommendations, it is necessary to connect the environment to a customer-owned Azure Data Lake Storage Gen 2 solution.

As Azure Data Lake Storage is configured in an environment, all necessary data is mirrored from the Entity store while still being protected and under customer's control.

If product recommendations or personalized recommendations are also enabled in the environment, then the product recommendations stack will be granted access to the dedicated folder in Azure Data Lake Storage to retrieve the customer's data and compute recommendations based on it.

## Prerequisites

Customers need to have Azure Data Lake Storage configured in an Azure subscription that they own. This topic does not cover the purchase of an Azure subscription or the setup of an Azure Data Lake Storage-enabled storage account.

For more information about Azure Data Lake Storage, see [Azure Data Lake Storage Gen2 official documentation](#).

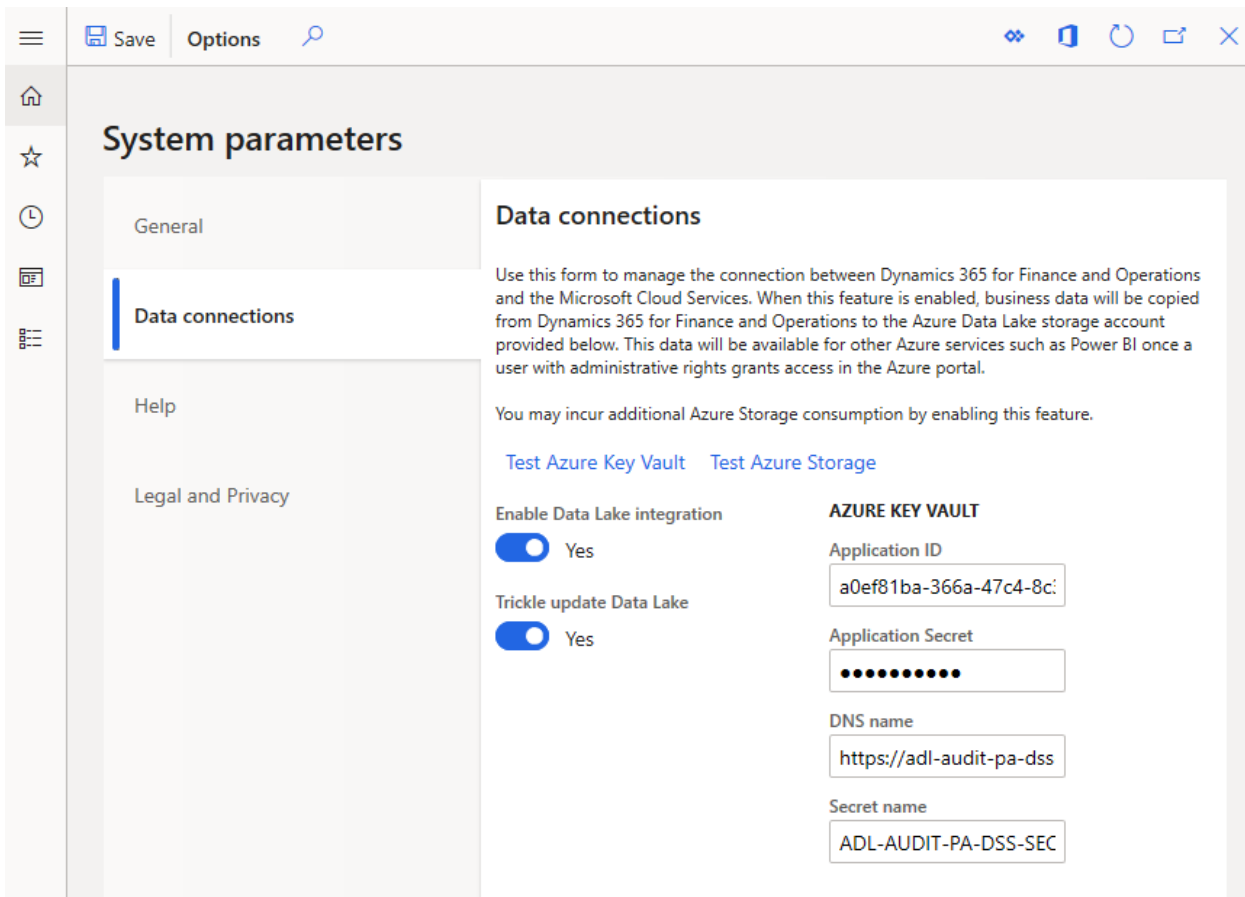
## Configuration steps

This section covers the configuration steps necessary for enabling Azure Data Lake Storage in an environment as it relates to product recommendations. For a more in-depth overview of the steps required to enable Azure Data Lake Storage, see [Make entity store available as a Data Lake](#).

### Enable Azure Data Lake Storage in the environment

1. Log in to the environment's back office portal.
2. Search for **System Parameters** and navigate to the **Data connections** tab.
3. Set **Enable Data Lake integration** to **Yes**.
4. Set **Trickle update Data Lake** to **Yes**.
5. Next, enter the following required information:
  - a. **Application ID // Application Secret // DNS Name** - Needed to connect to KeyVault where the Azure Data Lake Storage secret is stored.
  - b. **Secret name** - The secret name stored in KeyVault and used to authenticate with Azure Data Lake Storage.
6. Save your changes in the top left corner of the page.

The following image shows an example Azure Data Lake Storage configuration.



### Test the Azure Data Lake Storage connection

1. Test the connection to KeyVault using the **Test Azure Key Vault** link.
2. Test the connection to Azure Data Lake Storage using the **Test Azure Storage** link.

#### NOTE

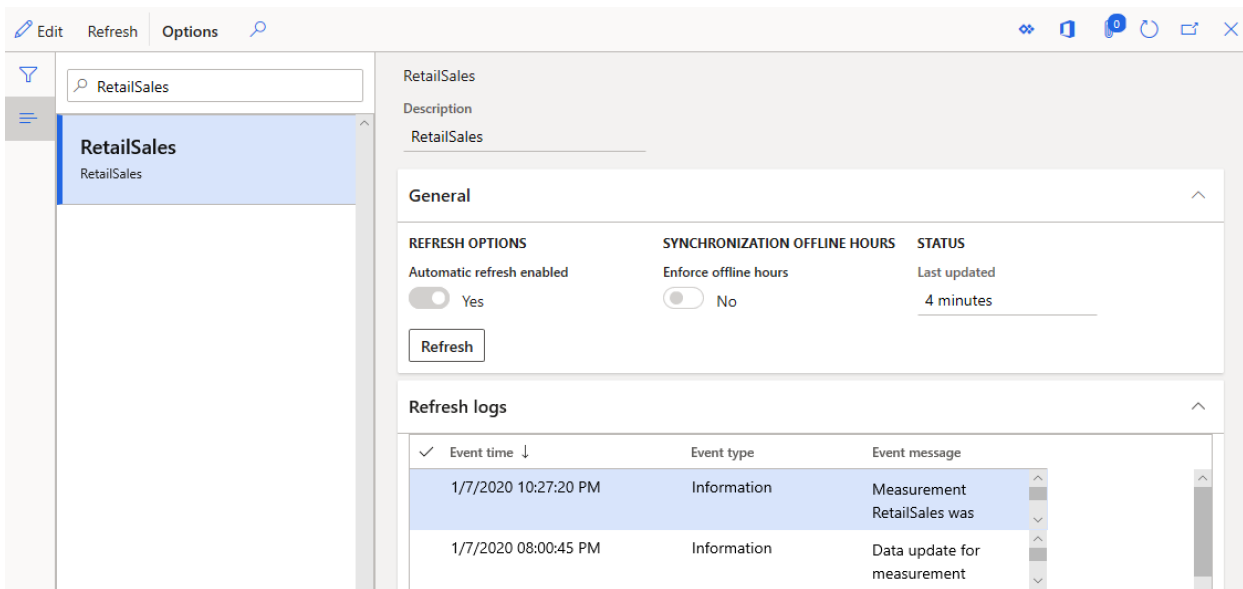
If the tests fail, double-check that all of the KeyVault information added above is correct, then try again.

Once the connection tests are successful, you must enable automatic refresh for Entity store.

To enable automatic refresh for Entity store, follow these steps.

1. Search for **Entity Store**.
2. In the list on the left, navigate to the **RetailSales** entry, and select **Edit**.
3. Ensure that **Automatic Refresh Enabled** is set to **Yes**, select **Refresh**, and then select **Save**.

The following image shows an example of Entity store with automatic refresh enabled.



Azure Data Lake Storage is now configured for the environment.

If not completed already, follow the steps for [enabling product recommendations and personalization](#) for the environment.

## Additional resources

[Make entity store available as a data lake](#)

[Product recommendations overview](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable product recommendations

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to make product recommendations that are based on artificial intelligence-machine learning (AI-ML) available for Microsoft Dynamics 365 Commerce customers. For more information about product recommendation lists, see [Product recommendations overview](#).

## Recommendations pre-check

Before enabling, note that product recommendations are only supported for Commerce customers who have migrated their storage to using Azure Data Lake Storage.

The following configurations must be enabled in the back office before enabling recommendations:

1. Ensure that Azure Data Lake Storage has been purchased and successfully verified in the environment. For more information, see [Ensure that Azure Data Lake Storage has been purchased and successfully verified in the environment](#).
2. Ensure that the entity store refresh has been automated. For more information, see [Ensure that the Entity store refresh has been automated](#).
3. Confirm that Azure AD Identity configuration contains an entry for Recommendations. More information on how to do this action is below.

Additionally, ensure that RetailSale measurements have been enabled. To learn more about this set up process, see [Work with measures](#).

## Azure AD Identity configuration

This step is required for all customers running an infra-structure as a service (IaaS) configuration. For customers running on service fabric (SF), this step should be automatic and we recommend verifying the setting is configured as expected.

### Setup

1. From the back office, search for the **Azure Active Directory applications** page.
2. Verify if an entry exists for "RecommendationSystemApplication-1".

If the entry does not exist, add a new entry with the following information:

- **Client Id** - d37b07e8-dd1c-4514-835d-8b918e6f9727
- **Name** - RecommendationSystemApplication-1
- **User Id** - RetailServiceAccount

Save and close the page.

## Turn on recommendations

To turn on product recommendations, follow these steps.

1. In Commerce headquarters, search for **Feature Management**.
2. Select **All** to see a list of available features.
3. In the search box, enter **Recommendations**.
4. Select the **Product recommendations** feature.

5. In the **Product recommendations** properties pane, select **Enable now**.

The screenshot shows the 'Feature management' interface. At the top, there's a 'Feature management' header with a sub-header 'Do not enable new features automatically'. Below this, there are three columns: 'New' with a count of 4, 'Not enabled' with a count of 374, and 'Scheduled' with a count of 0. There are two buttons: 'Enable all' and 'Check for updates'. Below the header, there are tabs for 'New', 'Not enabled', 'Scheduled', and 'All'. A search bar contains '\*recommendations'. A table lists features with columns: 'Feature name', 'Enable date', 'Feature added', 'Preview feature', and 'Module'. The table has two rows: 'Personalized product recommendations' and 'Product recommendations', both with an enable date of 4/24/2020 and module 'Retail and commerce'. The 'Product recommendations' row is selected. To the right of the table is a 'Product recommendations' properties pane. It shows 'Feature added 4/24/2020', 'Retail and commerce', and a 'Learn more' link. Below this is a 'Description' section with text: 'Enabling this feature will allow the system to use purchase and product information without associating personal information to generate machine-learning recommendations based on purchase trends in your online and brick-and-mortar stores. If you disable this feature you will not receive machine-generated recommendations.' There is also a 'Comments' section with a text area. At the bottom right of the properties pane are 'Schedule' and 'Enable now' buttons.

#### NOTE

This procedure starts the process of generating product recommendation lists. It may take several hours before the lists are available and can be viewed at the point of sale (POS) or in Dynamics 365 Commerce.

## Configure recommendation list parameters

By default, the AI-ML-based product recommendation list provides suggested values. You can change the default suggested values to suit the flow of your business. To learn more about how to change the default parameters, go to [Manage AI-ML-based product recommendation results](#).

## Show recommendations on POS devices

After enabling recommendations in Commerce back office, the recommendations panel must be added to the control POS screen using the layout tool. To learn about this process, see [Add a recommendations control to the transaction screen on POS devices](#).

## Enable personalized recommendations

In Dynamics 365 Commerce, retailers can make personalized product recommendations (also known as personalization) available. In this way, personalized recommendations can be incorporated into the online customer experience and at the point of sale. When the personalization functionality is turned on, the system can associate a user's purchase and product information to generate individualized product recommendations.

To learn more about personalized recommendations, see [Enable personalized recommendations](#).

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Opt out of personalized recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable personalized recommendations

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to make personalized product recommendations available for customers in Microsoft Dynamics 365 Commerce.

## Overview

In Dynamics 365 Commerce, retailers can make personalized product recommendations (also known as personalization) available. In this way, personalized recommendations can be incorporated into the customer experience online and at the point of sale (POS). When the personalization functionality is turned on, the system can associate a user's purchase and product information to generate individualized product recommendations.

## Personalization prerequisites

Before you make personalized product recommendations available for customers, note that product recommendations are supported only for Commerce users who have migrated their storage to Azure Data Lake Store. Before customers can receive personalized product recommendations, retailers must [turn on product recommendations](#).

### NOTE

By turning on product recommendations, you also turn on personalization. However, if you turn off personalization, you don't turn off the other types of product recommendations.

For more information about product recommendations, see the [Product recommendations overview](#).

## Turn on personalization

To turn on personalization, follow these steps.

1. In Commerce headquarters, search for **Feature Management**.
2. Select **All** to see a list of available features.
3. In the search box, enter **Recommendations**.
4. Select the **Personalized product recommendations** feature.
5. In the **Personalized product recommendations** properties pane, select **Enable now**.



**Feature management**  
Do not enable new features automatically

New: 4 | Not enabled: 372 | Scheduled: 0

Buttons: Enable all | Check for updates

Navigation: New | Not enabled | Scheduled | **All**

Search: \*Recommendations

Feature name ↑	Enable date	Feature added ↓	Preview feature	Module
Personalized product recommendations	8/14/2020	4/24/2020		Retail and commerce
Product recommendations		4/24/2020		Retail and commerce

**Personalized product recommendations**  
Feature added 4/24/2020  
Retail and commerce  
[Learn more](#)

**Description**  
Personalization allows the system to associate a user's purchase and product information to generate individualized product recommendations. Disabling personalization will turn off personalized product recommendations for all users. However, disabling personalization does not disable other recommendations features and users will continue to receive non-personalized recommendations.

**Comments**

Buttons: Schedule | **Enable now**

**NOTE**

When you turn on personalization, the process of generating personalized product recommendation lists is started. Up to one day might be required before these lists are available and visible online and at the POS.

## Personalized lists

In addition to allowing for personalization of existing machine-generated lists, the recommendations service allows for personalization of the product discovery experience both online and at the POS.

After personalization is turned on, retailers can show shoppers personalized "Picks for you" lists online or "Recommended for customer" lists on POS terminals. Additionally, retailers can apply personalization to existing product recommendation lists and provide General Data Protection Regulation (GDPR) opt-out experiences for authenticated users. If you turn off personalization, you also turn off these features.

### Online "Picks for you" lists

A "Picks for you" list is an artificial intelligence-machine learning (AI-ML) list that shows an authenticated user a personalized list of suggested products. This list is based on the user's omnichannel purchase history. Personalized recommendations are dynamically updated as the user makes more purchases. This type of list also supports category filtering, so that retailers can show top picks, based on navigational hierarchies.

Before the "Picks for you" list can appear on any e-Commerce page, the following user requirements must be met:

- Users must be signed in. Anonymous users won't see personalized recommendations.
- Users must have at least one purchase on their account.
- Users must opt in to receive personalized recommendations.

The following illustration shows an example of a "Picks for you" list on an online store page.

### Select product list configuration ✕

**Type**

- Products by Category
- Related to Products
- Curated
- New
- Best Selling
- Trending
- Frequently Bought Together
- People Also Like
- Picks For You**

**Picks For You**

This list shows personalized product recommendations for users based on their purchase history.

**List Source**

Page Context

Specific Category

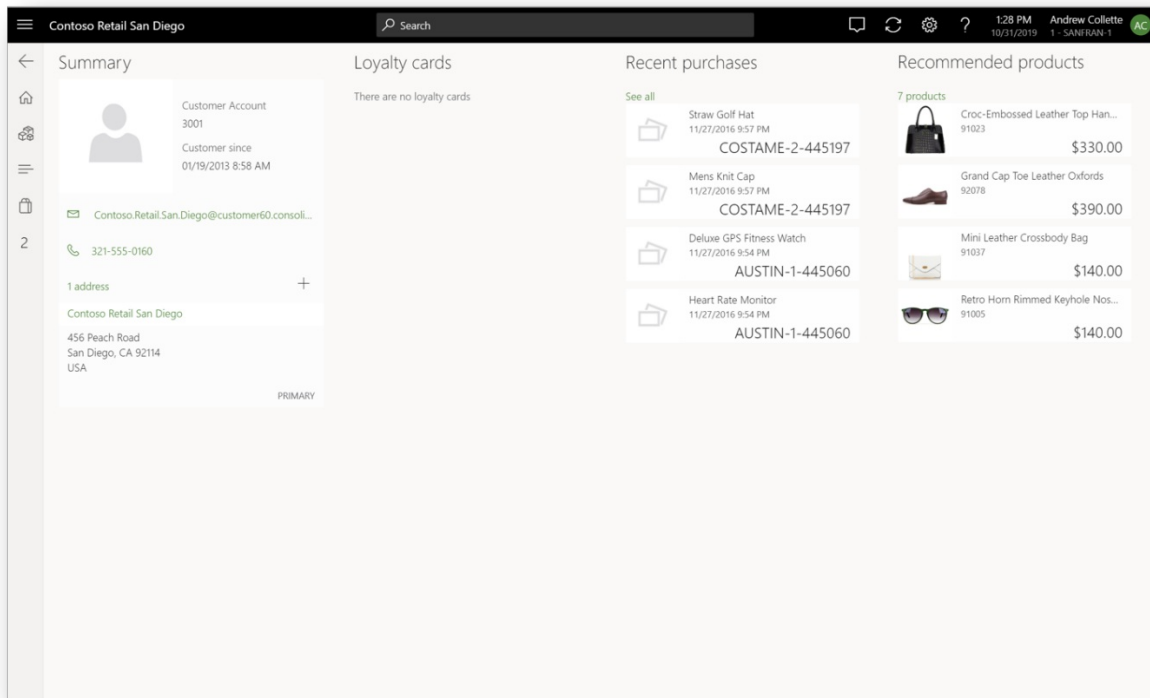
**Number of Items**

**OK** **Cancel**

### "Recommended for customer" lists at the POS

To enhance their clienteling experience, retailers can personalize existing customer details pages by adding a contextual "Recommended for customer" list.

The following illustration shows an example of a "Recommended for customer" list on a POS terminal.





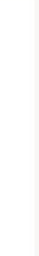


## Apply personalization to existing recommendation lists





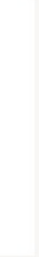
Retailers can apply personalization to existing recommendation lists, such as "New," "Trending," "Best selling," "People also like," and "Frequently bought together." When personalization is applied to existing lists, items that a signed-in user previously bought are removed from those lists. For both anonymous users and users who opted out of receiving personalized recommendations, default versions of the existing lists are shown. Therefore, retailers don't have to manually maintain separate page experiences.

For example, a signed-in user has already bought the black watch and the brown work boots that appear in the "Trending - default" list in the following illustration. Therefore, the user will see new products instead of those products, as shown in the "Trending - personalized" list.

Trending – default < >

				
<p>Large Face Black Leather Watch</p> <p><b>Free</b></p> <p>★★★★★ 4</p>	<p>Leather Pull-on Workboot</p> <p><b>\$170.00</b></p> <p>★★★★★ 4</p>	<p>Genuine Leather Suede Handbag Purse</p> <p><b>\$340.00</b></p> <p>★★★★★ 18</p>	<p>Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses</p> <p><b>\$140.00</b></p> <p>★★★★★ 18</p>	<p>Aqua GI</p> <p><b>\$190.0</b></p> <p>★★★</p>

Trending – personalized < >

				
<p>Genuine Leather Suede Handbag Purse</p> <p><b>\$340.00</b></p> <p>★★★★★ 18</p>	<p>Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses</p> <p><b>\$140.00</b></p> <p>★★★★★ 18</p>	<p>Aqua Glass Ball Necklace</p> <p><b>\$190.00</b></p> <p>★★★★★ 1</p>	<p>Round Pendant Hoop Earrings</p> <p><b>\$35.00</b></p> <p>★★★★★ 1</p>	<p>Silver CI</p> <p><b>\$360.00</b></p> <p>★★★</p>

To apply personalization to an existing recommendation list in the Commerce site builder, follow these steps.

1. Open an existing site builder page that contains a product collection module.
2. In the left navigation pane, select the product collection module.
3. In the right navigation pane, under **Products**, select the list.
4. In the **Select product list configuration** dialog box, under **Type**, select the list type.
5. Select the **Apply Personalization** check box, and then select **OK**.

### Select product list configuration ✕

**Type**

- Products by Category
- Related to Products
- Curated
- New
- Best Selling
- Trending**
- Frequently Bought Together
- People Also Like
- Picks For You

**Trending**

This list shows the highest performing products for a given time period.

List Source

Page Context

Specific Context

---

Apply Personalization

Number of Items

6. Save the page, finish editing it, and then publish it. After the page is published, signed-in users will see personalized trending lists.

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Opt out of personalized recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Opt out of personalized recommendations

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how you can let customers opt out of receiving personalized recommendations in Microsoft Dynamics 365 Commerce.

## Overview

During account creation, new customers are automatically set up to receive personalized recommendations. However, Dynamics 365 Commerce provides various ways for retailers to let users opt out of receiving these recommendations and restrict the processing of their personal data. Authenticated users who opt out of receiving personalized recommendations will immediately stop seeing personalized lists. Additionally, all personal data that is collected for personalization will be removed from personalized recommendations models.

For more information about personalized product recommendations, see [Enable personalized recommendations](#).

## Ways for retailers to implement an opt-out experience

Retailers have three ways to implement an opt-out experience.

### Opting out on behalf of users

In Account management in Commerce back office, retailers can opt out on behalf of users.

1. From the back-office home page, search for **all customers**.
2. Search for and select a customer, and then select the **Retail** FastTab.

All customers  
004320 : Tiddly Borgensen

Contact information

Miscellaneous details Always

Sales demographics USD

Credit and collections No | 0.00

Sales order defaults ..

Payment defaults Net10 | CRED

Invoice and delivery WA

Retail

<b>RETAIL</b> Amount charged, not posted 0.00	<b>RECEIPT</b> Receipt option Standard receipt Receipt email	<b>LOYALTY ENROLLMENT</b> Block customer for loyalty enrollment <input type="radio"/> No	<b>PRIVACY</b> Disable personalization <input type="radio"/> No Do not track web activity <input type="radio"/> No
---	---	--	--

3. Under **Privacy**, set the **Disable personalization** option to **Yes**.

**Retail**

<b>RETAIL</b> Amount charged, not posted 0.00	<b>RECEIPT</b> Receipt option Standard receipt Receipt email	<b>LOYALTY ENROLLMENT</b> Block customer for loyalty enrollment <input type="radio"/> No	<b>PRIVACY</b> Disable personalization <input type="radio"/> No Do not track web activity <input type="radio"/> No
---	---	--	--

4. Select **Save**, and close the page.

## Module-based opt-out experience

Retailers can let authenticated users opt out of personalized recommendations by themselves. To provide this opt-out experience, add the user opt-out module to customer account profile pages.

## Custom extensions

Retailers can create their own extensions to manage the opt-out experience for users. For more information, see [Call Retail Server APIs](#) and [Online channel extensibility](#).

# Obtain a digital copy of personalized recommendations data on behalf of an authenticated user

Customers might want to obtain a digital copy of their personal data and also see an exported view of their recommendations results. If a customer requests this information, the retailer must create a customized extension that calls the Retail Server application programming interface (API) and queries for the full results from the **Picks for you** list, based on the customer's customer ID. The results can then be exported in comma-separated values (CSV) format and shared with the customer.

The following example shows how a retailer can accomplish this task.

1. The retailer creates a custom extension to pull personal recommendations data on behalf of the user. For information about how to create modules, clone existing modules, call Retail Server APIs, and call data actions, see [Online channel extensibility](#).
2. The custom extension makes a call to the **get-recommendations** core data action and passes the required information to it, based on the requirements of the list. In the case of the **Picks for you** list, the extension must pass the correct list name and customer ID to the data action.

One way to create the custom extension is to clone the existing product collection module that is used to return recommendations results. By cloning this existing module, a retailer can modify the existing code and add a new button that exports the recommendations results to a CSV file. For more information, see [Clone a module library module](#) and [Product collection modules](#).

For a full view of the Retail Server API library, see [Retail Server Customer and Consumer APIs](#).

3. After the custom extension is created, the retailer can export a CSV file of all recommendations results, based on the unique customer ID of the authenticated user.
4. The retailer can share the exported CSV file that contains the full personalized list of recommended products with the authenticated user.

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

Create recommendations with demo data

Product recommendations FAQ

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Enable "shop similar looks" recommendations

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to enable "shop similar looks" product recommendations in Microsoft Dynamics 365 Commerce.

## Overview

The "shop similar looks" recommendations feature in Dynamics 365 Commerce uses the power of artificial intelligence and machine learning (AI-ML) to deliver recommendations for visually similar products to customers. By making "shop similar looks" recommendations available for all retail channels in Commerce, retailers can increase customer satisfaction by helping customers easily find what they want.

The functionality for "shop similar looks" recommendations uses product images of seed product variants to find and recommend visually similar products in a retailer's product catalog.

"Shop similar looks" recommendations are available in both the point of sale (POS) and e-Commerce experiences.

### Example scenarios

- A customer views a black striped sweater and receives a recommendation for a similar sweater in red. The customer selects the recommended product instead of the originally viewed product and then receives recommendations for similar products in red.
- A customer uses "shop similar looks" recommendations to discover matching earrings for a ring that the customer is interested in buying.

## Enable "shop similar looks" recommendations in Commerce headquarters

Product recommendations are supported only for Commerce users who have migrated their storage to Azure Data Lake Gen2.

### Prerequisites

Before retailers can begin to show "shop similar looks" recommendations to customers, there are two prerequisite steps:

- [Enable product recommendations](#) in Commerce headquarters.
- Confirm that the media server supports HTTPS calls.

For the recommendations engine to access the product images, retailers must generate the product URLs. To generate product URLs in Commerce headquarters, follow these steps.

1. Go to **Product images**.
2. On the Action Pane, select **Define media template**.
3. In the **Define media template** properties pane, under **Media URLs**, select **Generate URLs**.

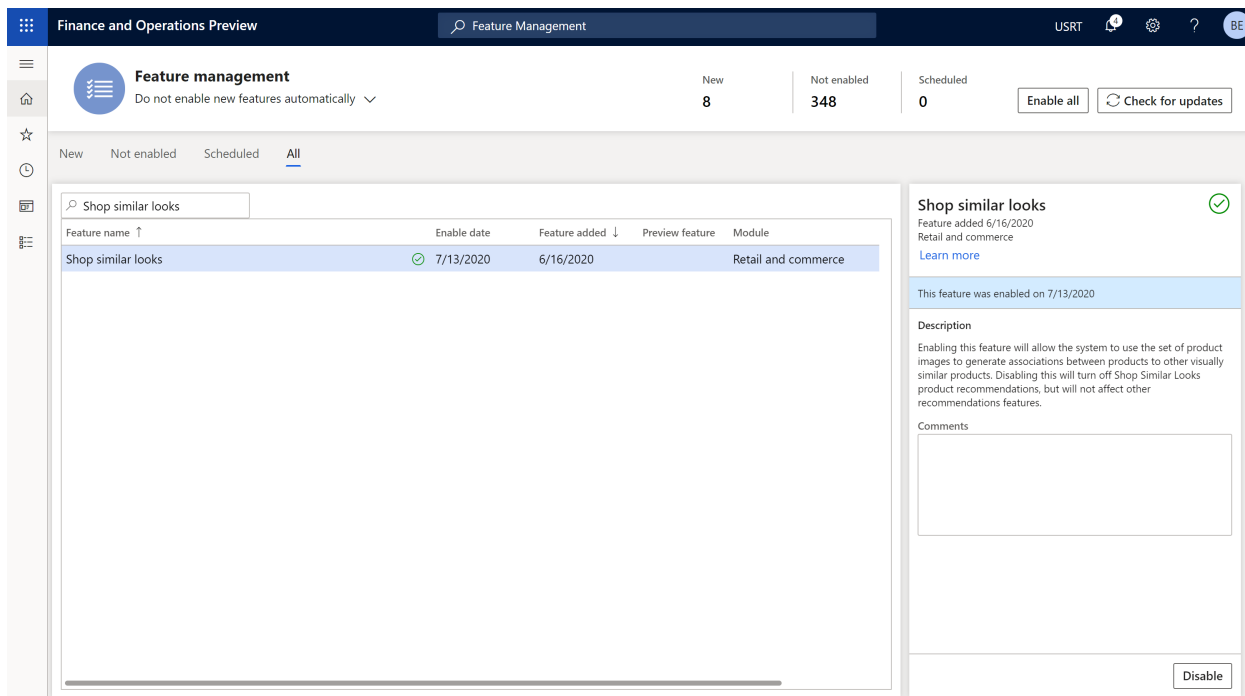
#### NOTE

When you enable the "shop similar looks" recommendations feature, the process of generating product recommendation lists begins. Up to a day might be required before those lists are available and visible online and on POS terminals.

To enable the "shop similar looks" recommendations feature in Commerce headquarters, follow these steps.

1. Go to **Feature management**.
2. In the list of available features, search for and select **Shop similar looks**.
3. In the right pane, select **Enable** to turn on the service.

The following illustration shows the **Shop similar looks** feature on the **Feature management** page in Commerce headquarters.



After the preceding tasks been completed, POS terminals are automatically enhanced with a contextual **Shop similar products** panel. By selecting **See more**, POS terminal users can be taken to a dedicated "shop similar looks" page that can be filtered further.

#### NOTE

If you turn off the "shop similar looks" recommendations feature, no other types of product recommendations are affected. For more information about product recommendations, see [Product recommendations overview](#).

## Add a Shop similar looks button to product details pages by using Commerce site builder

After you enable the "shop similar looks" recommendations feature in Commerce headquarters, an option in Commerce site builder lets retailers to add a **Shop similar looks** button to the buy box on any product details page (PDP). A customer who selects this button is taken to a dedicated "shop similar looks" page that returns visually similar products. There, the customer can use selectors to further filter the products.

To add a **Shop similar looks** button to a PDP by using Commerce site builder, follow these steps.

1. Open an existing site builder page that contains a buy box module.
2. In the left navigation pane, select the buy box module.
3. In the right navigation pane, select the **Enable Shop Similar Looks Link** check box.
4. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it. After the page is published, the PDP will include a **Shop similar looks** button.

The following illustration shows the **Enable Shop Similar Looks Link** check box and **Shop similar looks**

button on an example PDP in site builder.

The screenshot displays the Dynamics 365 Commerce site builder interface for a Product Detail Page (PDP). The main content area shows the product '3" Feather Earrings' priced at \$55.00. The product description reads: 'Black 3" feathers held by gold clips.' The quantity is set to 1, and there is a note 'Low stock available'. The 'Add to bag' button is highlighted in green. Below the product image, there are buttons for 'Find in store' and 'Shop similar looks', both of which are highlighted with red boxes. The right sidebar contains the 'Buybox' configuration panel, where the 'Enable Shop Similar Looks Link' checkbox is checked and highlighted with a red box. The left sidebar shows the page structure with various slots and containers.

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Opt out of personalized recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable "shop similar description" recommendations

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to enable "shop similar description" product recommendations in Microsoft Dynamics 365 Commerce.

The "shop similar description" recommendations feature in Dynamics 365 Commerce uses artificial intelligence and machine learning (AI-ML) to deliver recommendations for products that have descriptions that are similar to what the customer is looking for. By making "shop similar description" recommendations available for all retail channels in Commerce, retailers can help customers easily find what they want.

The functionality for "shop similar description" recommendations uses the product name and description of seed products to find and recommend similar products in a retailer's product catalog.

"Shop similar description" recommendations are available in both the point of sale (POS) and e-commerce experiences.

## Example scenarios

The following example scenarios show the types of recommendations that the "shop similar description" functionality can provide:

- A customer views a pair of retro-style horn-rimmed glasses and receives a set of recommendations for other glasses that have a similar design, in the context of the retailer's industry.
- A customer uses "shop similar description" recommendations to discover coffee flavors that are similar to a flavor that they previously purchased from the retailer.

## Set up "shop similar description" recommendations

Product recommendations are supported only for Commerce users who have migrated their storage to Azure Data Lake Storage Gen2.

### Prerequisites

Before "shop similar description" recommendations can be shown to customers, you must complete the following prerequisites:

- [Enable product recommendations](#) in Commerce headquarters.
- Confirm that the media server supports HTTPS calls.

### Turn on the "shop similar description" recommendations feature

To turn on the "shop similar description" recommendations feature in Commerce headquarters, follow these steps.

1. In the **Feature management** workspace, in the list of available features, search for and select **Shop similar description**.
2. In the right pane, select **Enable**.

## NOTE

When you turn on the feature, the system starts to generate product recommendation lists. It might take up to a day for those lists to become available and visible online and on POS terminals.

If you turn off the feature, other types of product recommendations aren't affected. For more information about product recommendations, see [Product recommendations overview](#).

## Add a Shop similar description button to product details pages

After you turn on the "shop similar description" recommendations feature in Commerce headquarters, you can add a **Shop similar description** button to the buy box on any product details page (PDP). A customer who selects this button is taken to a dedicated **Shop similar description** page that shows visually similar products. The customer can then use selectors to further filter the products.

To add a **Shop similar description** button to a PDP by using Commerce site builder, follow these steps.

1. Open an existing site builder page that contains a buy box module.
2. In the left navigation pane, select the buy box module.
3. In the right pane, select the **Enable Shop Similar description Link** check box.
4. Select **Save**.
5. Select **Finish editing** to check in the page, and then select **Publish** to publish it. After the page is published, the PDP will include a **Shop similar description** button.

The following illustration shows the **Enable shop similar description Link** check box and the **Shop similar description** button on an example PDP in site builder.

The screenshot displays the Commerce site builder interface for a product details page (PDP). On the left, a navigation pane shows the page structure, with the 'Buybox (Required)' module selected. The main content area shows a product titled '3" Feather Earrings' with a price of 'Free'. Below the product image, there are buttons for 'Add to bag', 'Find in store', 'Shop similar looks', and 'Shop similar description'. On the right, the 'Buybox' configuration panel is visible, with the 'Enable Shop Similar description Link' checkbox checked and highlighted by a red box. Other options in the panel include 'Enable Shop Similar Looks Link', 'Allow custom price', 'Minimum price', 'Maximum price', and 'Event IDs to track'.

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Product recommendations FAQ](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add product recommendations on POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

At its core, product recommendations are a transformative business application that span across all commerce spaces to create rich, engaging, and tailored product discovery experiences. To implement this feature on POS, follow the steps on [how to add recommendations to your POS devices](#).

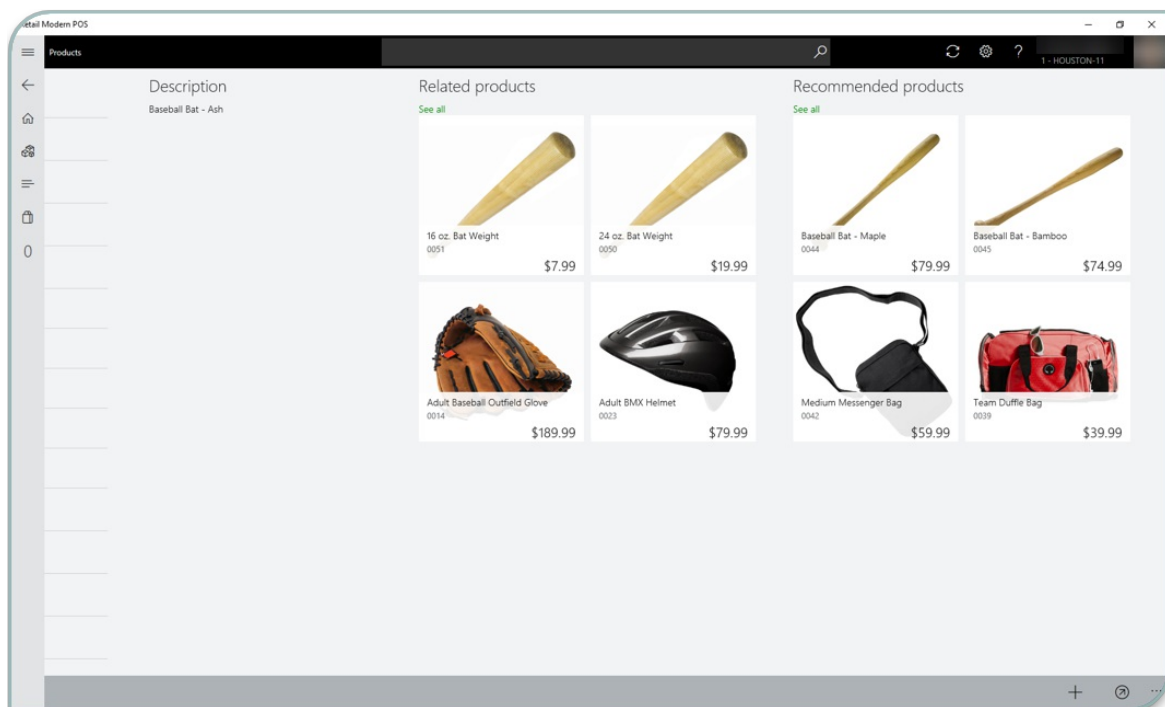
For more information about product recommendations features, read the [product recommendations overview](#).

## Scenarios

Product recommendations are enabled for the following POS scenarios. They are available in Cloud POS or Modern POS (MPOS).

### 1. On the **Product details** page:

- If a store associate visits a **Product details** page when looking at previous transactions across different channels, the recommendations service suggests additional items that are likely to be purchased together.

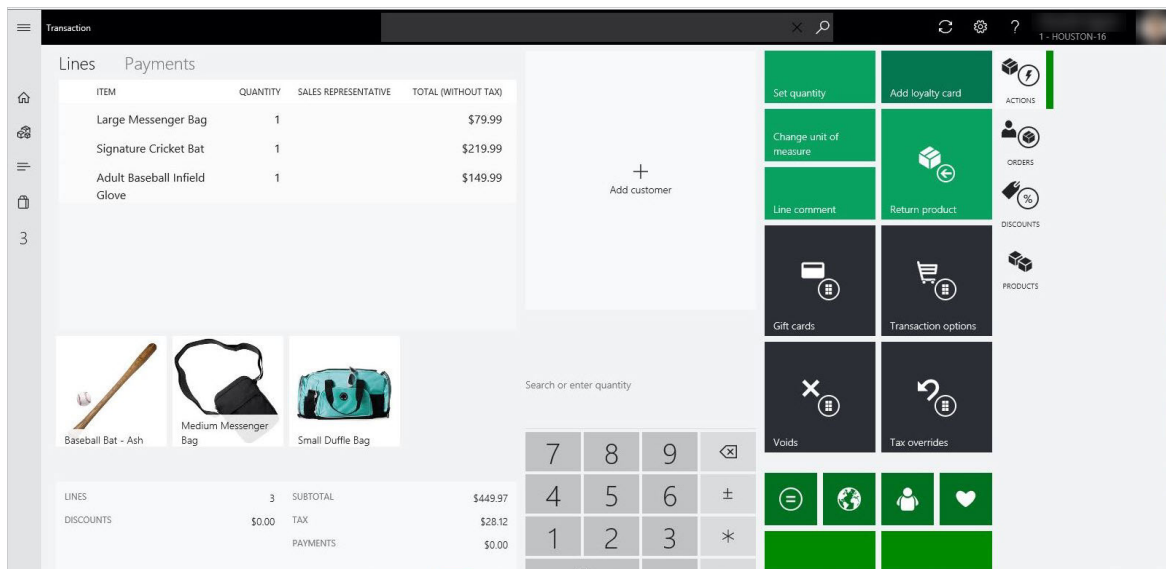


### 2. On the **Transaction** page:

- The recommendation engine suggests items based on the entire list of items in the basket that are frequently bought together.

#### **NOTE**

To display recommendations on the **Transaction** page, the retailer needs to update the screen layout in Dynamics 365 Commerce. The **Recommendations** control must be dropped onto the **Transaction** page.



## Configure Commerce to enable POS recommendations

To set up product recommendations, follow these steps:

1. Ensure your service has been updated to the **10.0.6 build**.
2. Follow the instructions on how to [enable product recommendations](#) for your business.
3. Optional: To display recommendations on the transaction screen, go to **Screen Layout**, choose your screen layout, launch the **Screen layout designer**, and then drop the **recommendations** control where needed.
4. Go to **Commerce parameters**, select **Machine-learning**, select **Yes** under **Enable POS recommendations**.
5. To see recommendations on POS, run global configuration job **1110**. To reflect changes made to POS screen layout designer, run channel configuration job **1070**.

## Troubleshoot issues where you have Product recommendations already enabled

- Navigate to **Commerce Parameters > Recommendation lists > Disable product recommendations** and run **Global configuration job [9999]**.
- If you added the **Recommendations** control to your transaction screen using the **Screen layout designer**, please remove that as well.
- If you have additional questions, check out the [Product recommendations FAQ](#) for more information.

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)



[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add recommendations to the transaction screen

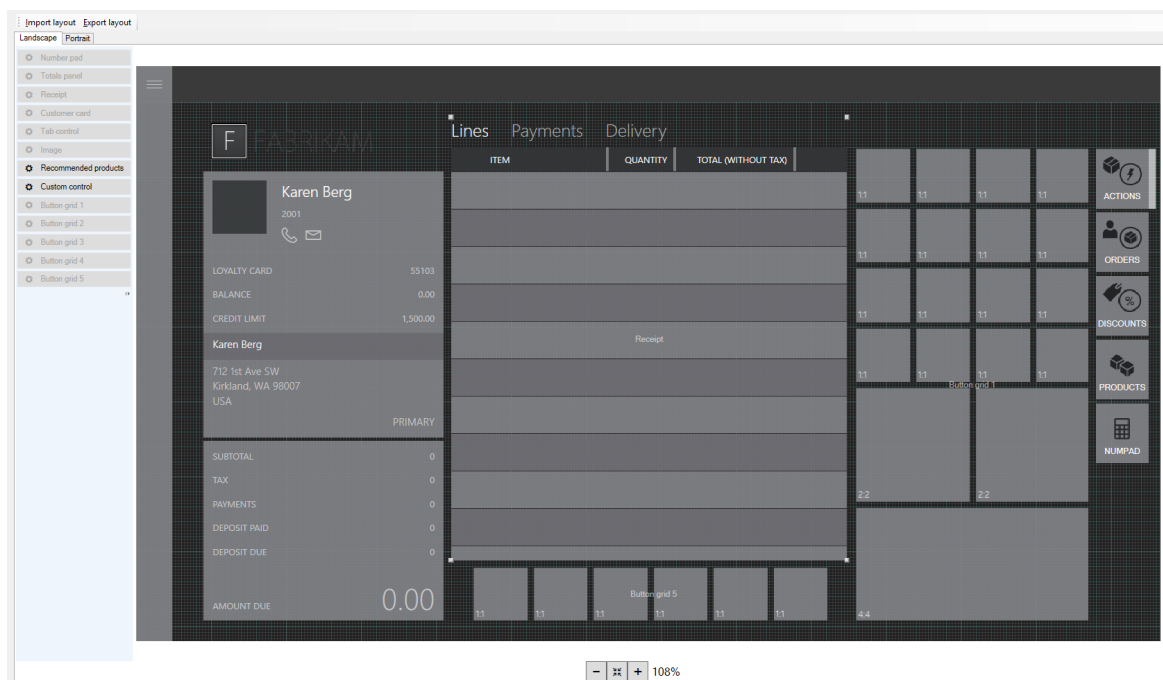
2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add a recommendations control to the transaction screen on a point of sale (POS) device using the screen layout designer in Microsoft Dynamics 365 Commerce. For more information about product recommendations, read the [product recommendations on POS documentation](#).

You can display product recommendations on your POS device when you use Commerce. To display product recommendations, you need to add a control to the transaction screen using the screen layout designer.

## Open Layout designer

1. Go to **Retail and Commerce > Channel setup > POS setup > POS > Screen layouts**.
2. Use the Quick Filter to find the screen that you want to add the control to. For example, filter on the **Screen layout ID** field using a value of **F2CP16:9M**.
3. In the list, find and select the desired record. For example, select **Name: F2CP16:9M Screen Layout ID: F2CP16:9M**.
4. Click **Layout designer**.
5. Follow the prompts to launch the layout designer. When prompted for credentials, enter the same credentials that were in use when the Layout designer was launched from **Screen layouts** page.
6. When you log in, a page similar to the one below appears. The layout will be different depending on the customizations that were made for your store.



## Choose a display option

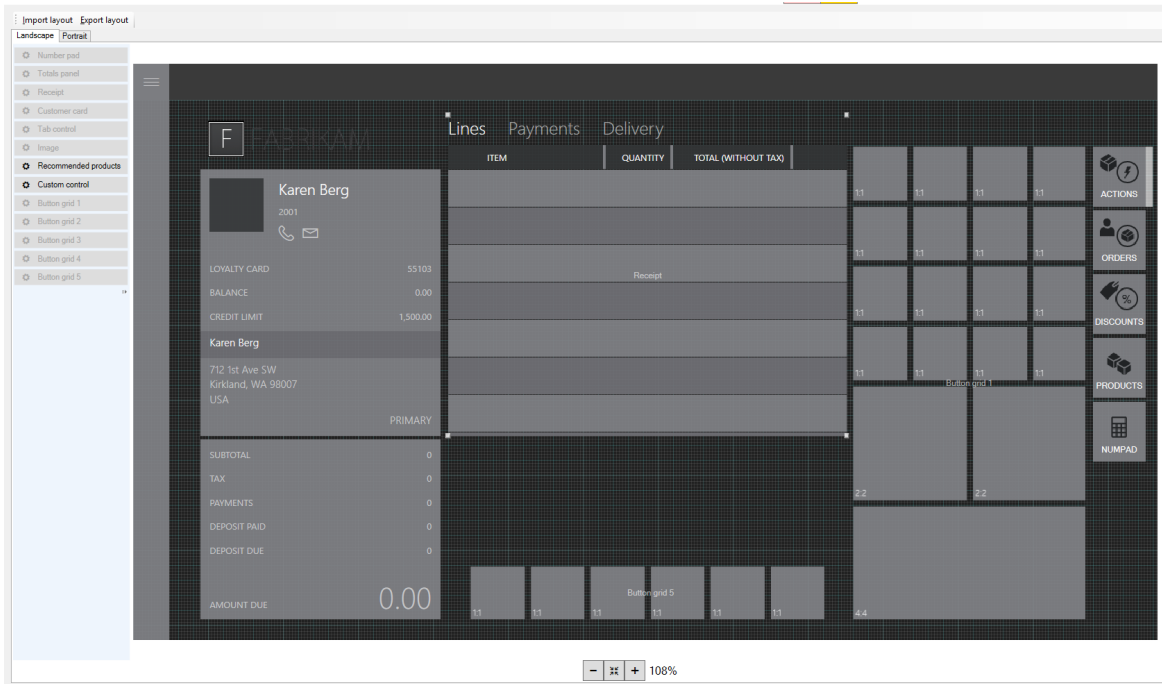
There are two configurations options available. Choose the option that works best for your store, and follow the remaining instructions to finish setting up the control. The two options are:

- Recommendations are always visible.

- A **Recommendations** tab appears in the grid on the right side of the screen.

### Make recommendations always visible

1. Reduce the height of the transaction lines details area so that it is the same height as the customer panel to its left.



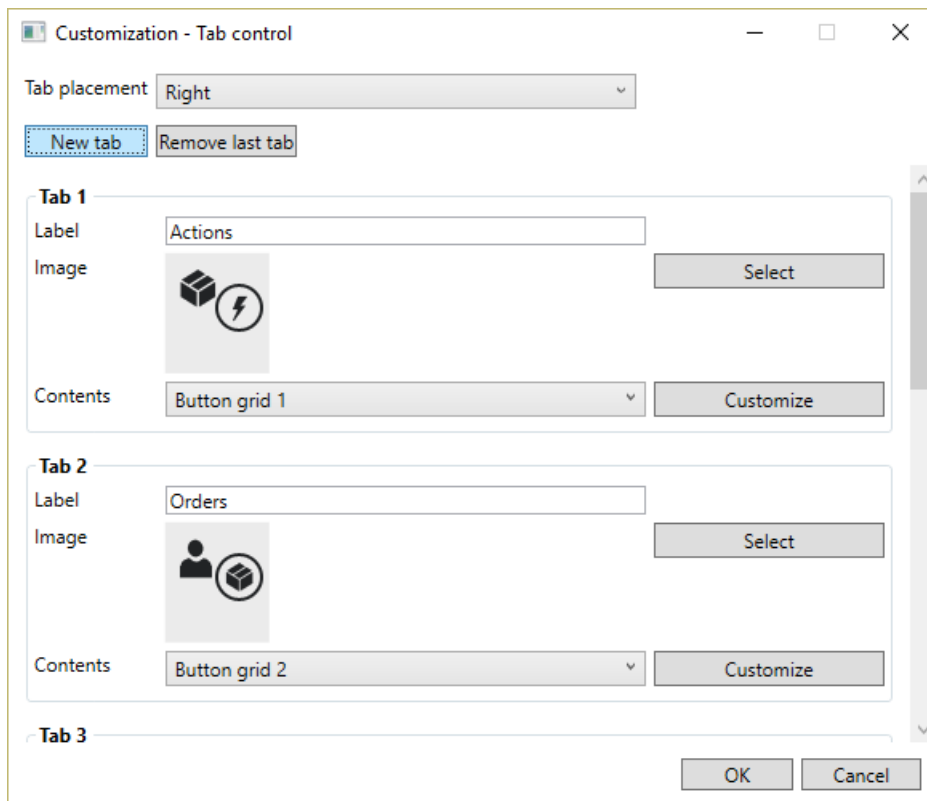
2. From the menu on the left, drag and drop the recommendations control to between the transaction line details area and the button grid in the center bottom of the transaction screen. Resize the control so it fits in that space.



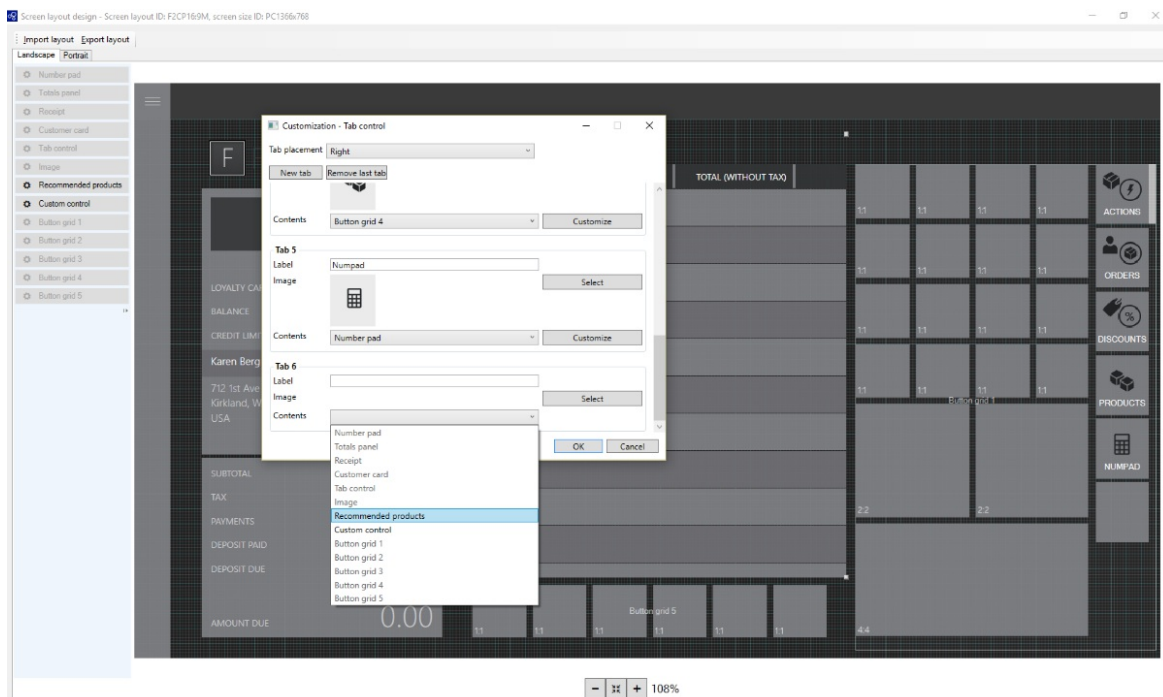
3. Click the **X** to save and exit Layout designer.
4. In Commerce, go to **Retail and Commerce > Retail and Commerce IT > Distribution schedules**.
5. In the list, select **1090 Registers**.
6. Click **Run now**.

### Add a Recommendations tab to the button grid on the right side of the screen

1. Right-click in the empty space below the last tab on the button grid located on the right side of the page.
2. Click **Customize**.



3. Click **New tab**.
4. Find the new tab that you just added. You may need to scroll down.
5. In the **Contents** drop-down, select **Recommended products**.



6. In the **Label** field, type a name for the recommendations tab. For example, type 'Recommended products'.
7. In the **Image** field, select the image to appear on the tab.
8. Click **OK**. The new tab appears in the button grid.

9. Click the **X** to save and exit Layout designer.
10. In Commerce, go to **Retail and Commerce > Retail and Commerce IT > Distribution schedules**.
11. In the list, select **1090 Registers**.
12. Click **Run now**.

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Adjust AI-ML-based product recommendation results

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how to adjust product recommendation results based on artificial intelligence-machine learning (AI-ML) to your business.

After enabling product recommendations, the default settings will take effect; these parameters will work for many needs. It is best to plan to spend some time evaluating whether the results fit the selling motion of products. We suggest evaluating results for a few days before changing parameters as needed before testing again.

## Understanding recommendation list parameters

Before changing the parameters, learn about how they will affect the results below.

### "Trending" product list

The "Trending" product list has two parameters that can be changed:

**Trending product list configuration**

This list shows the highest performing products for a given time period.

Include new products from last X days

Include sales from last X days

1. **Include new products from last X days** - Products that have been added within the specified number of days before the current date can be used to select product candidates. The default value in the picture suggests that products as old as 180 days can be used in the trending product list.
2. **Include sales from last X days** - Sales transactions that have occurred within the specified number of days before the current date can be used to order the products. The default value above suggests that all purchases made of a product in the last 30 days would be used to determine the placement of the product in the trending product list.

### "Best selling" product list

Depending on your business, the "Best selling" list can bring different results than trending, even though they both use transaction data to order products. Because best selling has no cut off based on assortment date, Best selling can still highlight very popular, older products that might have been dropped from the trending list.

The "Best selling" product list has one parameter that can be changed:

**Bestselling list configuration**

This list shows products ranked by the highest number of sales.

Include sales from last X days

1. **Include sales from last X days** - Sales transactions that have occurred within the specified number of

days before the current date can be used to order the products. The default value above suggests that all purchases made of a product in the last 30 days would be used to determine the placement of the product in the Best selling product list.

## Manually add or remove products from recommendation lists

### For "New," "Trending," or "Best selling" lists

1. Go to **Retail and Commerce > Product recommendations > Recommendation parameters.**
2. In the list of shared parameters, select **Recommendation lists.**
3. Select the list add or remove products from.
4. To add products to the table, select **Add line.**
5. Under the Product column, search for a product by **Name** or **Product number.**

#### New product list configuration

This list shows the newest products assorted to channels and catalogs.  
Products included will appear at the beginning of the list. Products excluded will not appear in the list.

[+ Add line](#) [🗑 Remove](#)

✓	Product	Line type	Display order
	0004	Include	1.00
	0061	Exclude	0.00
✓	0001	Include	2.00

Product number ↑      Product name

- 0001 Youth Accessory Combo Set
- 0002 Adult Helmet Accessory Combo...
- 0003 Signature Mountain Bike Tire
- 0004 Premium Mountain Bike Tire
- 0005 Basic Inner Tube
- 0006 Inner Tube Patches
- 0007 Full Finger BMX Gloves

6. Under the Line type column, select one of two options:
  - **Include** – forces a product to the front of the list
  - **Exclude** – removes a product from appearing in the list

## New product list configuration

This list shows the newest products assorted to channels and catalogs.

Products included will appear at the beginning of the list. Products excluded will not appear in the list.

[+ Add line](#) [Remove](#)

✓	Product	Line type	Display order
	0061	Exclude	0.00
	0001	Include	2.00
✓	0004	Include	1.00

7. Changing the **Display order** will change the order that products marked **include** will appear in the list.

- If two products have the same **display order** value, then the final order of those two results may differ from the back office.

8. To remove products from the table: select the line to remove and select **Remove**.

### For "People also like" or "Frequently bought together" lists

In the context of "Frequently bought together" or "People also like" lists, machine learning is used to analyze consumer purchase patterns to recommend related products commonly purchased together for a unique seed product.

A *seed product* is the product you want to generate results for. In the context of manually adjusting recommendation lists, you are adding or removing results for this product.

Follow these steps to manually add or remove results for a seed product:

1. Select the **Seed product**.
2. Under the **Product** column, search for a product by **Name** or **Product number**.

### Frequently bought together configuration

This list uses machine learning to analyze consumer purchase patterns to recommend related items that are commonly purchased for a given seed item.

Products included will appear at the beginning of the list. Products excluded will not appear in the list.

[+ Add line](#) [Remove](#)

✓	Seed product	Product	Line type	Display order
✓	0006		Exclude	0.00

Product number ↑	Product name
0001	Youth Accessory Combo Set
0002	Adult Helmet Accessory Combo...
0003	Signature Mountain Bike Tire
0004	Premium Mountain Bike Tire
0005	Basic Inner Tube
0006	Inner Tube Patches
0007	Full Finger BMX Gloves

3. Under the **Line type** column, select one of two options:

- **Include** – forces a product to the front of the list
- **Exclude** – removes a product from appearing in the list



## Frequently bought together configuration

This list uses machine learning to analyze consumer purchase patterns to recommend related items that are commonly purchased for a given seed item.

Products included will appear at the beginning of the list. Products excluded will not appear in the list.

[+ Add line](#) [Remove](#)

✓	Seed product	Product	Line type	Display order
	0006	0002	Include	1.00
	0049	0094	Include	1.00
✓	0006	0124	Include	2.00
			Exclude	
			Include	

4. To remove products from the table: select the line to remove and select Remove.

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manually create curated recommendations

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how merchandizers can manually create and manage product recommendations lists for Microsoft Dynamics 365 Commerce customers.

Curated lists are collections of individual content, created and curated by people.

## Create a new list

To create a curated product recommendation list, follow these steps.

1. Go to **Retail and Commerce > Product recommendations > Recommendation lists**.
2. Select **New**.
3. In the **List Id** field, enter a value.
4. In the **List name** field, enter a value.
  - The **List name** is the title of the list that will appear in the curated lists section of the **Product collection** module.
5. To add products to the list, select **Add products**.
6. To change the order of the products in the list, enter a value in the **Display order** column.
  - If two products have the same display order value, then the final order of those two results may differ from the back office.
7. Select **Save** to save the list.


## Example List

[Recommendation lists](#)

### sb1-popular : Popular near you

---

#### General

List ID	List title	List type
sb1-popular 	Popular near you	Curated

---

#### Products

[+ Add products](#) [Remove](#)

✓	Product name	Type	Display order ↑
	Cotton Polo	Product	1.00
	Slim Fit Dress Shirt	Product	2.00
	Pearl Sapphire Pendant Necklace	Product	3.00
	Violet Heart Pendant Necklace	Product	4.00
	Brown Leopardprint Sunglasses	Product	5.00
	Yellow Snakeskin Bag	Product	6.00
	White Fur Insert Coat	Product	7.00

# Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Create recommendations with demo data](#)

[Product recommendations FAQ](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create recommendations with demo data

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides guidance on how to leverage omni-channel product recommendations in Tier-1 single box environments using pre-populated, customizable demo data.

Omni-channel product recommendations provide a set of editorially curated or programmatically generated list of products. These lists can be used in several scenarios, depending on the business need. For more information about product recommendation lists, see [Product recommendations overview](#).

For Tier-2 and higher Dynamics 365 environments, product recommendations are automatically computed based on customer data. Using product recommendations demo data does not disable any product recommendations solution already provisioned in the environment and any costs associated with its usage.

For Tier-1 environments, product recommendations are based only off the static demo data stored in a .csv file.

## Enabling product recommendations demo data in an environment

To enable product recommendations demo data, you need to deploy the Dynamics 365 Commerce Preview Demo Extension to the respective environment. Doing so automatically enables product recommendations demo data.

## Default demo data

Each OneBox type environment comes with a preloaded set of product recommendations demo data stored in the comma separated 'reco\_demo\_data.csv' file, located on the Commerce Scale Unit.

The data is structured along the following columns.

COLUMN NAME	MANDATORY	DESCRIPTION	POSSIBLE VALUES
RecoList	✓	The specific product recommendation list type that the demo data point is to generate.	<ul style="list-style-type: none"><li>• RecoBestSelling</li><li>• RecoNew</li><li>• RecoTrending</li><li>• RecoCart</li><li>• RecoPeopleAlsoBuy</li></ul>
OperatingUnitNumber	✓	The specific operating unit number where product recommendations are expected to be surfaced.	
Category		The category the specific list should be returned for. If no category is specified, the list is for top of navigation hierarchy only.	

COLUMN NAME	MANDATORY	DESCRIPTION	POSSIBLE VALUES
SeedItemId		For lists that require seed (RecoPeopleAlsoBuy and RecoCart), the product those lists should show additional products for.	
CustomerId		For lists that require a customer identifier (RecoPicks). The default value '0' applies to all customers.	
ItemIds	✓	One or more products to be returned as the result, separated by ','.	

## Customize demo data

You can edit the default demo data with any product and category information configured in HQ. After you update the .csv, the product recommendations that are returned to customers will immediately reflect the changes.

The extension contains a datafile called 'RecoMockDataset.csv', which allows you to control the dataset used to power the mock recommendations results. The file name can be controlled through extension configuration using the **ext.Recommendations.DemoFilePath** setting. This enables you to have multiple datasets available that can be switched between easily through configuration.

```
<settings>
  <add name="ext.Recommendations.DemoFilePath" value="RecoMockDataset.csv" />
</settings>
```

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Product recommendations FAQ](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product recommendations FAQ

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides information about processes and tools that you can use to troubleshoot issues that are related to [product recommendations](#) or their results.

## Best practices

It's very important to utilize the concept of product masters and variants. The sensible grouping of variants to a parent product master helps the list algorithms and service create better models. Additionally, the service can serve just one instance of a product instead of putting all closely related variants in a list. When all closely related variants are put in a list, erroneous or duplicate results can occur.

## Why are products missing from my recommendation lists?

Typically, if an item is missing from a product recommendation list, there might be a product configuration issue. For example, there might be an incorrect product start date or end date, a dimension might be misconfigured, or the product might not be in the correct channel assortment, etc.

If an item is missing from a recommendation list that is based on artificial intelligence-machine learning (AI-ML), the item might not fit the criteria of the recommendation list, or it might not have enough purchase transactions for the recommendation list to show it.

We recommend that you check these steps:

1. **Make sure that product recommendations have been enabled in HQ.** For more information about how to enable this service, see [Enable product recommendations](#).
2. **Make sure that key product properties are set.** For example, product assortments must be set to **Include**.
3. **For newly assorted products, it might take up to 3 hours before the product will start appearing in the new list.**
4. **If a product is still not appearing in Trending, Best selling, People also like, or Frequently bought together, then that product might not have enough transactions.** In this case, you can either wait for more transactions to occur, update the default recommendation list parameters, or use manual intervention to modify the recommendation product list results. For more information about recommendation parameters, see [Manage AI-ML-based product recommendation results](#).
5. **Make sure that the product meets the recommendation criteria for the list.** For more information about product recommendation parameters, see [Manage AI-ML-based product recommendation results](#).

## How can I prevent poor recommendation results from being returned?

Recommendation lists require a large volume of transactions to produce results. Therefore, it's important that users provide full historical transaction data.

Additionally, products that have no transactions or few transactions typically don't have **People also like** or **Frequently bought together** results, and don't appear in **Trending** or **Best selling** recommendation lists. This situation can often occur for very new products, or for old products that have a small number of purchases. Popular new items will easily overcome this issue.

We recommend that you follow these steps:

1. **Make sure that the product meets the recommendation criteria for that list.** For more information about product recommendation parameters, see [Modify AI-ML-based product recommendation results](#).
  2. **If the product is new, consider modifying a recommendation list until the product has more transactions.** For more information about how to modify recommendation list results, see [Manage AI-ML-based product recommendation results](#).
- **Make sure that the product meets the recommendation criteria for that list.** For more information about product recommendation parameters, see [Manage AI-ML-based product recommendation results](#).
  - **If the product is new, consider modifying a recommendation list until the product has more transactions.** For more information about how to modify recommendation list results, see [Manage AI-ML-based product recommendation results](#).

## Can I remove a product but still see it in the store?

You can adjust lists that are algorithmically generated if a business need arises. However, if a product is removed from a recommendation list, the product will remain discoverable in the store. For more information about how to modify product recommendation results, see [Manage AI-ML-based product recommendation results](#).

If you must block an item from being discovered in the store, you must change the **Item assortments** value to **Exclude**.

## How do I add a list to an e-Commerce page?

For information about how to add product recommendation pages to your e-Commerce website, see [Add product recommendation lists to pages](#).

## How do I enable recommendations on POS?

After enabling product recommendations, you will need to add the recommendations panel to the control POS screen. For more information, see [Add a recommendations control to the transaction screen on POS devices](#).

## Additional resources

[Product recommendations overview](#)

[Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#)

[Enable product recommendations](#)

[Enable personalized recommendations](#)

[Opt out of personalized recommendations](#)

[Enable "shop similar looks" recommendations](#)

[Add product recommendations on POS](#)

[Add recommendations to the transaction screen](#)

[Adjust AI-ML recommendations results](#)

[Manually create curated recommendations](#)

[Create recommendations with demo data](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Ratings and reviews overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers ratings and reviews in Microsoft Dynamics 365 Commerce.

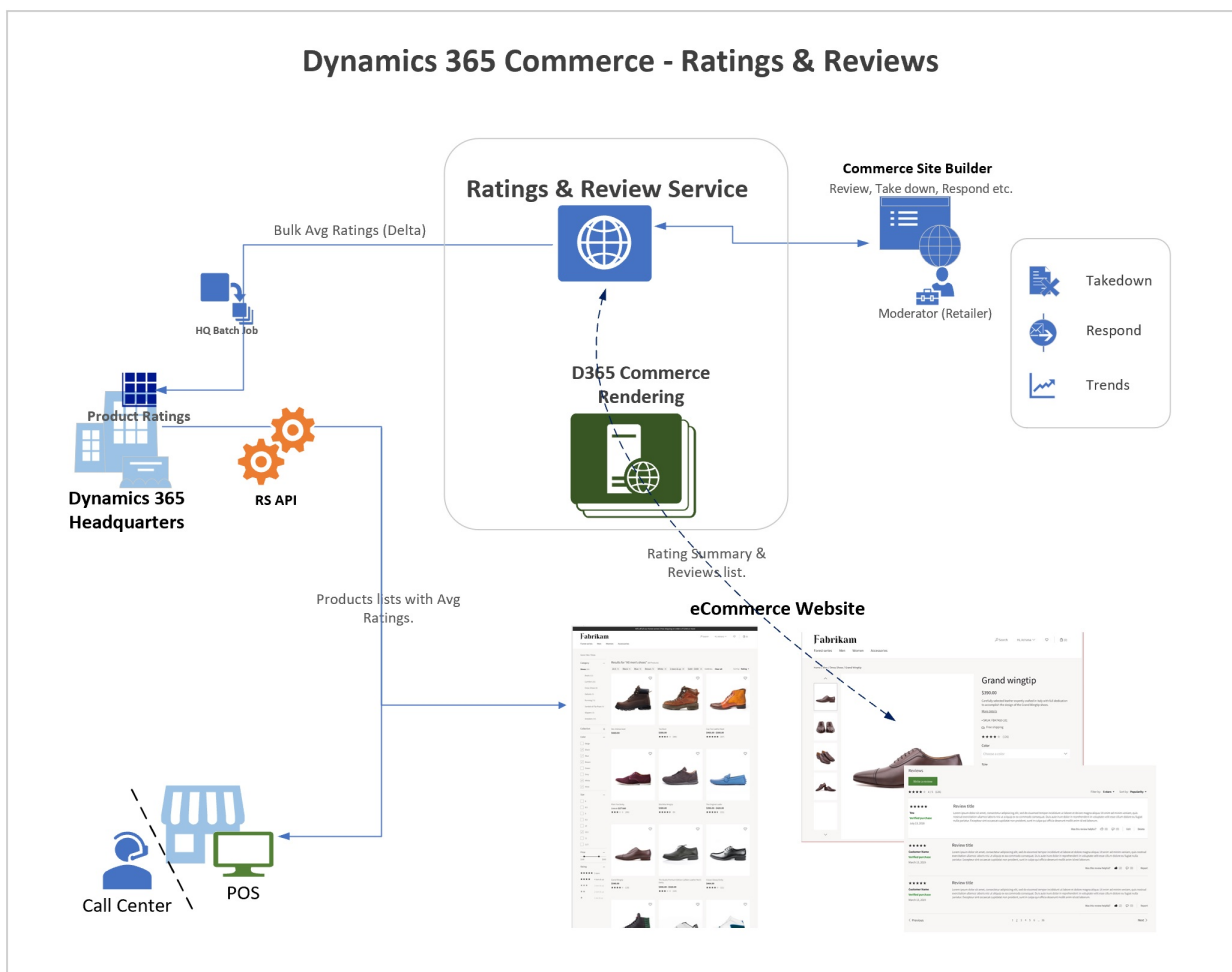
## Overview

Ratings and reviews are crucial for e-Commerce customers who want to know how other customers perceive a product. They can also help consumers make purchase decisions. In Dynamics 365 Commerce, the ratings and reviews solution lets retailers capture product reviews and ratings from customers. Retailers can then show average ratings and review information across their e-Commerce website.

Average rating information is shown in point of sale (POS) and call center channels. Therefore, sales associates can use it to help users make decisions. Ratings and reviews can also serve as a feedback mechanism that retailers can use to improve the quality of a product and therefore increase sales.

Ratings and reviews functionality in Dynamics 365 Commerce is an omnichannel solution and is natively available as part of the platform. The ratings and reviews solution is built on top of Microsoft Azure, which provides high scalability and reliability.

The following illustration shows how the ratings and reviews solution works in Dynamics 365 Commerce.



The ratings and reviews solution in Dynamics 365 Commerce uses Azure Cognitive Services to offer automatic moderation of profane words in 40 languages. Because human approval isn't required, moderation costs are reduced. The system also offers moderator tools that can be used to respond to customer concerns, feedback,

and take-down requests, and to address data requests from users.

The ratings and reviews solution provides widgets that show rating summaries in product lists, in search results, on product details page, and in other places. The widgets show complete review lists, and they also provide sorting and filtering options.

The ratings and reviews solution also provides a business intelligence (BI) template that includes a set of metrics to provide insights into ratings and reviews. Ratings and reviews data can be exported for further analysis.

## Additional resources

[Opt in to use ratings and reviews](#)

[Manage ratings and reviews](#)

[Configure ratings and reviews](#)

[Sync product ratings in Dynamics 365 Commerce](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Opt in to use ratings and reviews

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to opt in to use ratings and reviews on your Microsoft Dynamics 365 Commerce site.

## Overview

The ratings and reviews solution is an omni-channel solution that you can make available in Dynamics 365 Commerce by using Microsoft Dynamics Lifecycle Services (LCS). LCS is an administration portal that retailers use to manage their environments from provisioning to decommissioning.

If you want to use the ratings and reviews solution on your Commerce website, you must opt in for ratings and reviews during deployment of your e-Commerce site on Dynamics 365 Commerce.

## Opt in to use ratings and reviews

To opt in to use ratings and reviews on your site, follow these steps.

1. Follow the steps in [Deploy a new e-Commerce site](#).
2. While you're still in LCS, go to **Retail deployment setup > Other settings**.
3. Set the **Enable ratings and reviews service** option to **Yes**.
4. In the **AAD security group for ratings and review moderator (security group object id)** field, enter the ID of the Microsoft Azure Active Directory (Azure AD) security group that includes the ratings and reviews moderators.

### Other Settings

Supported Host Names (Separated by ';')

Ex - www.Contoso.com;www.ContosoUSA.com \*

AAD Security Group for System Admin (Security Group Object ID)

Ex - 64cdfb02-69c7-4c1c-80c3-a20f0b760357 \*

B2C Tenant Name

Ex - contosob2c.onmicrosoft.com

B2C Client ID

Ex - 1607521e-3a35-499a-b4c8-92c59a97c0bd

B2C Login Custom Domain

Ex - login.b2clogin.com

B2C Reply URL

Ex - https://www.contoso.com

B2C SignUp SignIn Policy ID

Ex - b2c\_1a\_signup\_signin

B2C Reset Password Policy ID

Ex - B2C\_1a\_PasswordReset

B2C Edit Profile Policy ID

Ex - B2C\_1a\_ProfileEdit

Enable Ratings And Reviews Service

Yes

AAD Security Group for Ratings and Review Moderator (Security Group Object ID)

Ex - d31304d5-edea-4864-a860-2feb86f8ecf0 \*

Initialize Cancel

5. Complete the e-Commerce initialization process.

#### NOTE

If you are an existing Dynamics 365 Commerce customer who has already deployed an e-Commerce site without having opted in for ratings and reviews and now want to use ratings and reviews from the Dynamics 365 Commerce package, please submit a service request. For information about how to submit a service request, see [Submit service requests process](#).

## Additional resources

[Ratings and reviews overview](#)

[Manage ratings and reviews](#)

[Configure ratings and reviews](#)

[Sync product ratings in Dynamics 365 Commerce](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage ratings and reviews

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how to manage ratings and reviews in Microsoft Dynamics 365 Commerce site builder.

## Overview

Dynamics 365 Commerce uses Microsoft Azure Cognitive Service to automatically moderate review text by redacting profane words. In addition, moderators can use Dynamics 365 Commerce site builder to implement the following manual tasks:

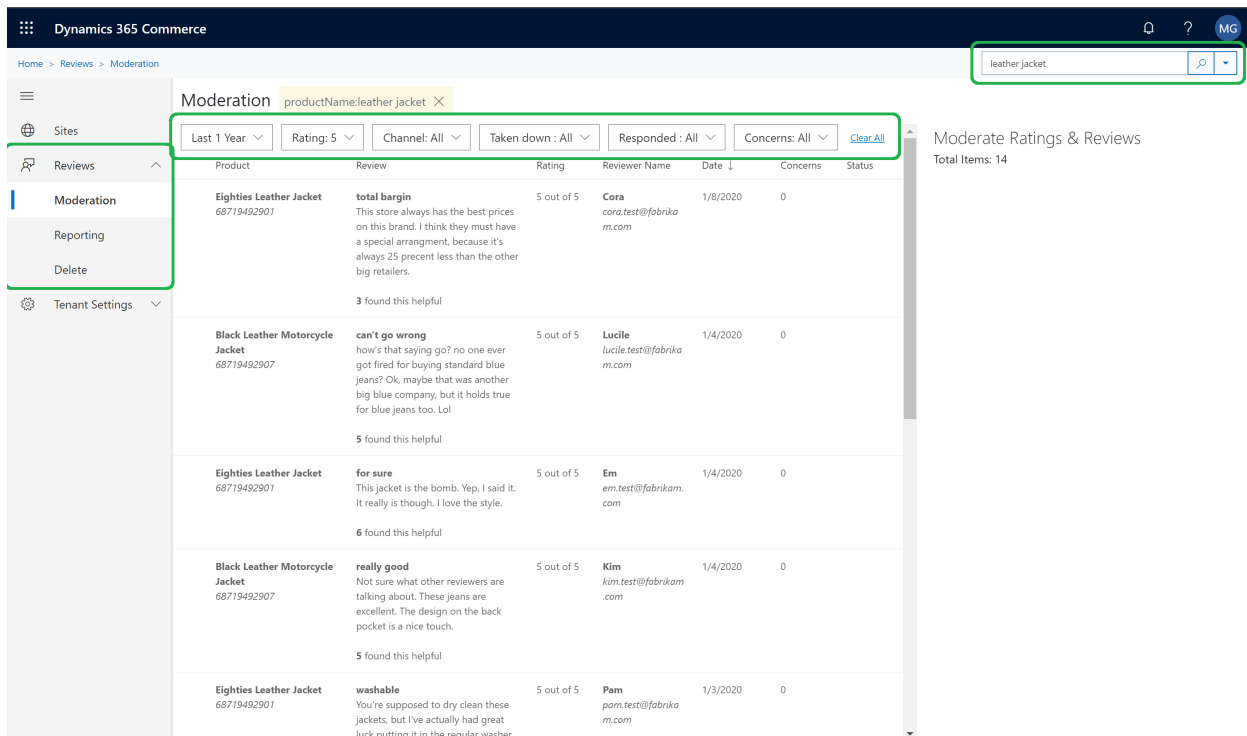
- Moderate reviews by responding to them or removing them.
- Delete a customer's reviews at the customer's request.
- Bulk-import ratings and reviews data for all products into a Microsoft Power BI template, so that trends for ratings and reviews can be analyzed.

## Read a review

To read to a review in Commerce site builder, follow these steps.

1. Go to **Home > Reviews > Moderation**.
2. Use the search field in the upper right of the page to filter the reviews that are shown by product ID, product name, or review text.

Additional filters let you limit the reviews by period, rating, channel, or concern status (taken down, responded, or reported).



The screenshot shows the Dynamics 365 Commerce Moderation interface. The search bar at the top right contains the text "leather jacket". The main content area displays a table of reviews for the product "Eighties Leather Jacket". The table has columns for Product, Review, Rating, Reviewer Name, Date, Concerns, and Status. The reviews are filtered by "Last 1 Year", "Rating: 5", "Channel: All", "Taken down: All", "Responded: All", and "Concerns: All".

Product	Review	Rating	Reviewer Name	Date	Concerns	Status
Eighties Leather Jacket 68719492901	<b>total bargain</b> This store always has the best prices on this brand. I think they must have a special arrangement, because it's always 25 percent less than the other big retailers.  3 found this helpful	5 out of 5	Cora cora.test@fabrikam.com	1/8/2020	0	
Black Leather Motorcycle Jacket 68719492907	<b>can't go wrong</b> how's that saying go? no one ever got fired for buying standard blue jeans? Ok, maybe that was another big blue company, but it holds true for blue jeans too. Lol  5 found this helpful	5 out of 5	Lucile lucile.test@fabrikam.com	1/4/2020	0	
Eighties Leather Jacket 68719492901	<b>for sure</b> This jacket is the bomb. Yep, I said it. It really is though. I love the style.  6 found this helpful	5 out of 5	Em em.test@fabrikam.com	1/4/2020	0	
Black Leather Motorcycle Jacket 68719492907	<b>really good</b> Not sure what other reviewers are talking about. These jeans are excellent. The design on the back pocket is a nice touch.  5 found this helpful	5 out of 5	Kim kim.test@fabrikam.com	1/4/2020	0	
Eighties Leather Jacket 68719492901	<b>washable</b> You're supposed to dry clean these jackets, but I've actually had great luck putting it in the regular washer	5 out of 5	Pam pam.test@fabrikam.com	1/3/2020	0	

## Respond to a review

Sometimes, customers who purchased a product express their satisfaction or dissatisfaction, or they don't understand how to use the product. As a moderator, you can post a response to a review. This response appears

together with the review on the site.

To respond to a review in Commerce site builder, follow these steps.

1. Go to **Home > Reviews > Moderation**.
2. Find and select the review that requires a response.
3. In the properties pane on the right, select **Add a response**.
4. Enter the response text and the name that should be shown for the responder. The default responder name is **Moderator**.
5. When you've finished, select **Post response**.

The screenshot shows the Dynamics 365 Commerce Moderation interface. The main area displays a table of reviews for the product 'leather jacket'. The table has columns for Product, Review, Rating, Reviewer Name, Date, Concerns, and Status. The second review, 'can't go wrong' by Lucile, is selected. On the right, the properties pane shows the review details and a 'Respond to Review' form with an 'Add a response' button highlighted in green.

Product	Review	Rating	Reviewer Name	Date	Concerns	Status
Eighties Leather Jacket 68719492901	<b>total bargain</b> This store always has the best prices on this brand. I think they must have a special arrangement, because it's always 25 percent less than the other big retailers. 3 found this helpful	5 out of 5	Cora cora.test@fabrikam.com	1/8/2020	0	
<b>Black Leather Motorcycle Jacket</b> 68719492907	<b>can't go wrong</b> how's that saying go? no one ever got fired for buying standard blue jeans? Ok, maybe that was another big blue company, but it holds true for blue jeans too. Lol 5 found this helpful	5 out of 5	Lucile lucile.test@fabrikam.com	1/4/2020	0	
Eighties Leather Jacket 68719492901	<b>for sure</b> This jacket is the bomb. Yep, I said it. It really is though. I love the style. 6 found this helpful	5 out of 5	Em em.test@fabrikam.com	1/4/2020	0	
Black Leather Motorcycle Jacket 68719492907	<b>really good</b> Not sure what other reviewers are talking about. These jeans are excellent. The design on the back pocket is a nice touch. 5 found this helpful	5 out of 5	Kim kim.test@fabrikam.com	1/4/2020	0	
Eighties Leather Jacket 68719492901	<b>washable</b> You're supposed to dry clean these jackets, but I've actually had great luck putting it in the regular washer on the gentle cycle. Be sure to only put a few drops of detergent in though. No shrinkage so far -- fingers crossed. 5 found this helpful	5 out of 5	Pam pam.test@fabrikam.com	1/3/2020	0	
Eighties Leather Jacket 68719492901	<b>get it</b> want a nuck confidene builder? Buy this jacket. How can you not	5 out of 5	Archibald archibald.test@fabrikam.com	1/3/2020	0	

## Take down a review

Sometimes, there is a business justification for moderators to take down customer reviews.

To take down a review in Commerce site builder, follow these steps.

1. Go to **Home > Reviews > Moderation**.
2. Find and select the review that must be taken down.
3. In the properties pane on the right, select a takedown reason under **Takedown Review**, and then select **Take down**.

## Delete a customer's reviews at the customer's request

Sometimes, customers want their ratings and reviews data to be permanently deleted from an e-Commerce website. A moderator who receives a removal request from a customer can remove the customer's data by using the review deletion feature. To find and delete a customer's data, the moderator requires the email address that the customer used to sign in and provide reviews.

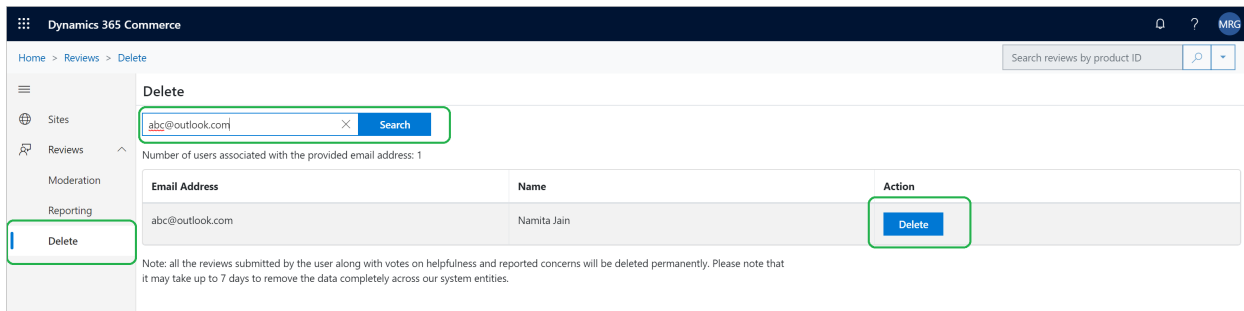
To find and delete customer data in Commerce site builder, follow these steps.

1. Go to **Home > Reviews > Delete**.
2. In the **Search for users by email address** box, enter the customer's email address, and then select **Search**.
3. If the customer has any review activity (for example, review submissions, votes about the helpfulness of



another customer's reviews, or comments about another customer's review), the results are shown. For each item, there is a **Delete** button.

4. For each item that must be deleted, select **Delete**. When you're prompted for confirmation, select **Yes**.



#### NOTE

- It can take up to seven days for data to be completely removed from the system. Moderators should notify customers about this delay.
- If customers have changed their name in their account settings, multiple items might appear in the search results. In this case, to completely delete the customer's data, the moderator must select **Delete** for each item.

## Download ratings and reviews data

Commerce site builder lets moderators import ratings and reviews data in bulk, so that they can analyze trends. A Power BI template that includes basic metrics is available. Moderators can use this template to connect bulk-imported data and view a dashboard. They don't have to create a custom dashboard. Moderators can also customize the Power BI template to meet their specific needs.

To download ratings and reviews data in Commerce site builder, follow these steps.

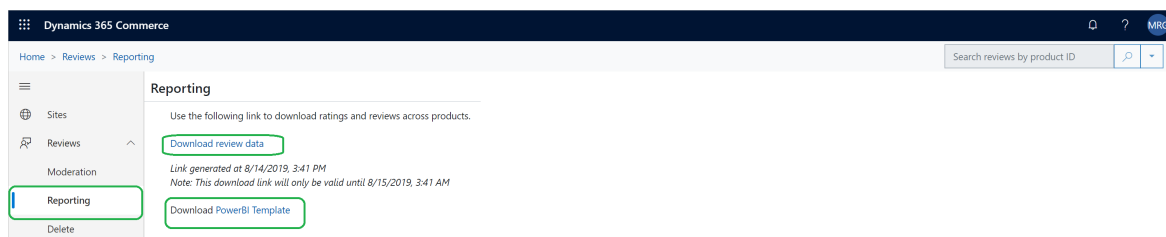
1. Go to **Home > Reviews > Reporting**.
2. Select **Download review data** to download ratings and reviews data in bulk in comma-separated values (CSV) format.

## View ratings and reviews trends

Moderators can download the Power BI template so that they can view trends in a dashboard.

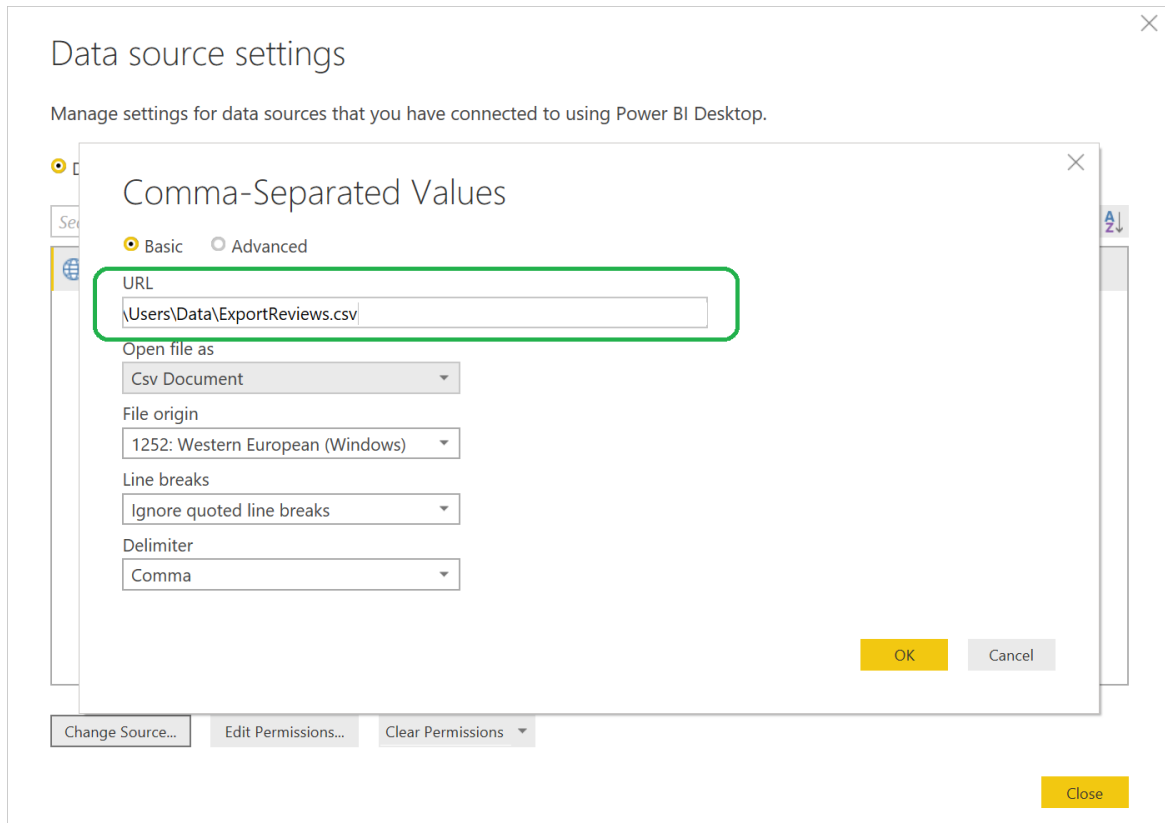
To view ratings and reviews trends in Commerce site builder, follow these steps.

1. Go to **Home > Reviews > Reporting**.
2. Select **PowerBI template** to download the template.

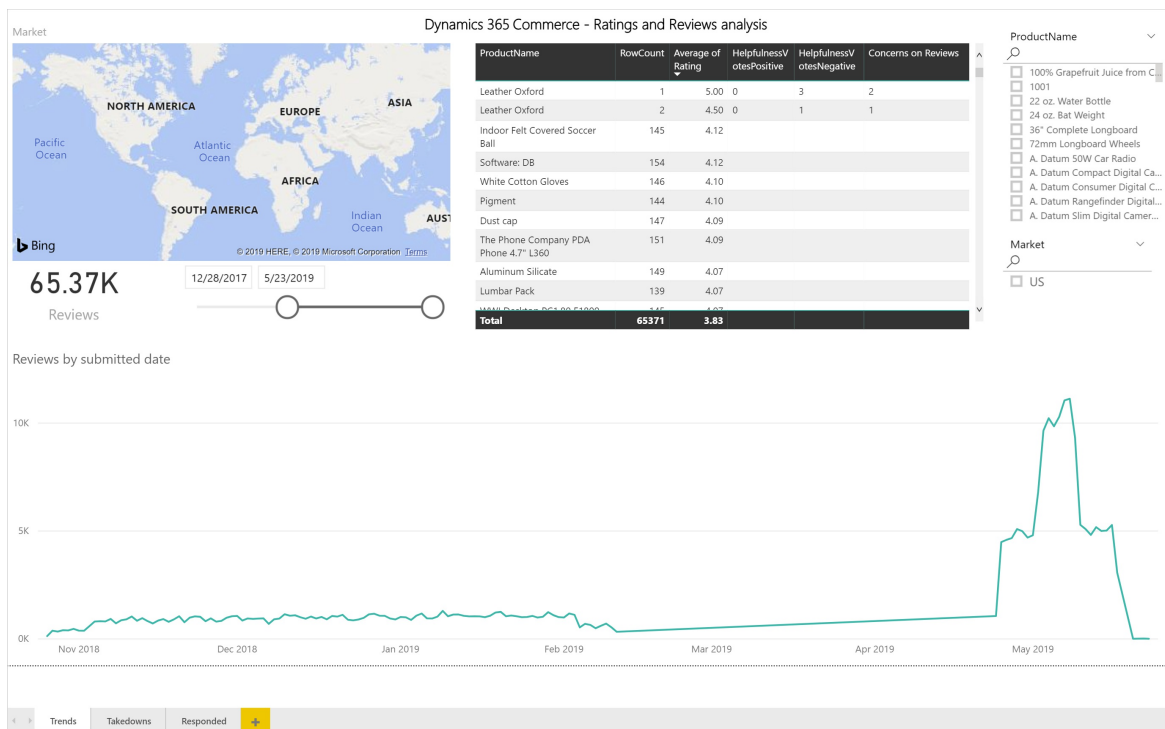


3. Open the downloaded template by using the Power BI app. Close the **Access to web content** dialog box that appears, and then close the "Refresh" error message that appears.
4. Go to **Home**, select **Edit queries**, and then select **Data source settings**.
5. In the **Data source settings** dialog box, select **Change Source**.

- In the URL field, enter the path of the reviews data that you downloaded in the previous procedure (for example, c:\reviews\ReviewsData.csv).



- Select **OK**, and then select **Apply changes**. It will take one to two minutes to apply your changes to the data source.
- Select **Trends sheet** to view ratings and reviews trends.



## Additional resources

[Ratings and reviews overview](#)

[Opt in to use ratings and reviews](#)

## Configure ratings and reviews

### Sync product ratings in Dynamics 365 Retail

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure ratings and reviews

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to configure your e-Commerce site to show customer ratings and reviews in Microsoft Dynamics 365 Commerce.

## Overview

Ratings and reviews on e-Commerce websites help customers learn about products before they make a purchase decision by showing them what other customers think about those products. For e-Commerce websites, ratings and reviews are also a mechanism for collecting customer feedback about products.

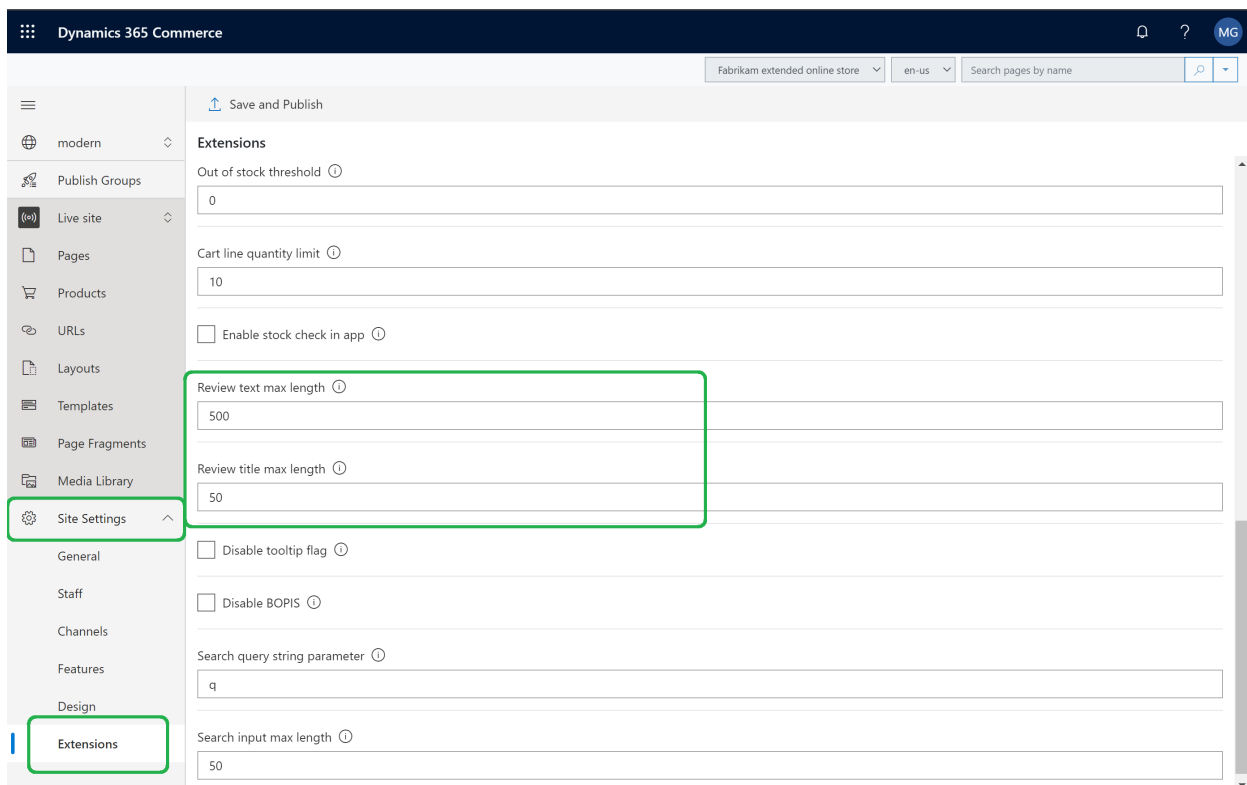
## Configure a site to show ratings and reviews

Configuration values for ratings and reviews, such as the tenant ID, review text length, and review title length, are configured at the site level.

To configure a site to show ratings and reviews, follow these steps.

1. Go to **Home > Sites**.
2. Select the name of your site.
3. Go to **Site settings > Extensions**.
4. In the **Review text max length** field, enter the maximum number of characters that review text can have (for example, 1000).
5. In the **Review title max length** field, enter the maximum number of characters that review titles can have (for example, 55).
6. Select **Save and Publish**.

The following illustration shows what this configuration looks like in Dynamics 365 Commerce.



The screenshot shows the Dynamics 365 Commerce interface. The top navigation bar includes the Dynamics 365 Commerce logo, a search bar, and user information (MG). The left sidebar shows the navigation menu with 'Site Settings' and 'Extensions' highlighted. The main content area displays the 'Extensions' configuration page for the 'Fabrikam extended online store' in the 'en-us' locale. The 'Review text max length' field is set to 500, and the 'Review title max length' field is set to 50. Other fields include 'Out of stock threshold' (0), 'Cart line quantity limit' (10), 'Enable stock check in app' (unchecked), 'Disable tooltip flag' (unchecked), 'Disable BOPIS' (unchecked), 'Search query string parameter' (q), and 'Search input max length' (50).

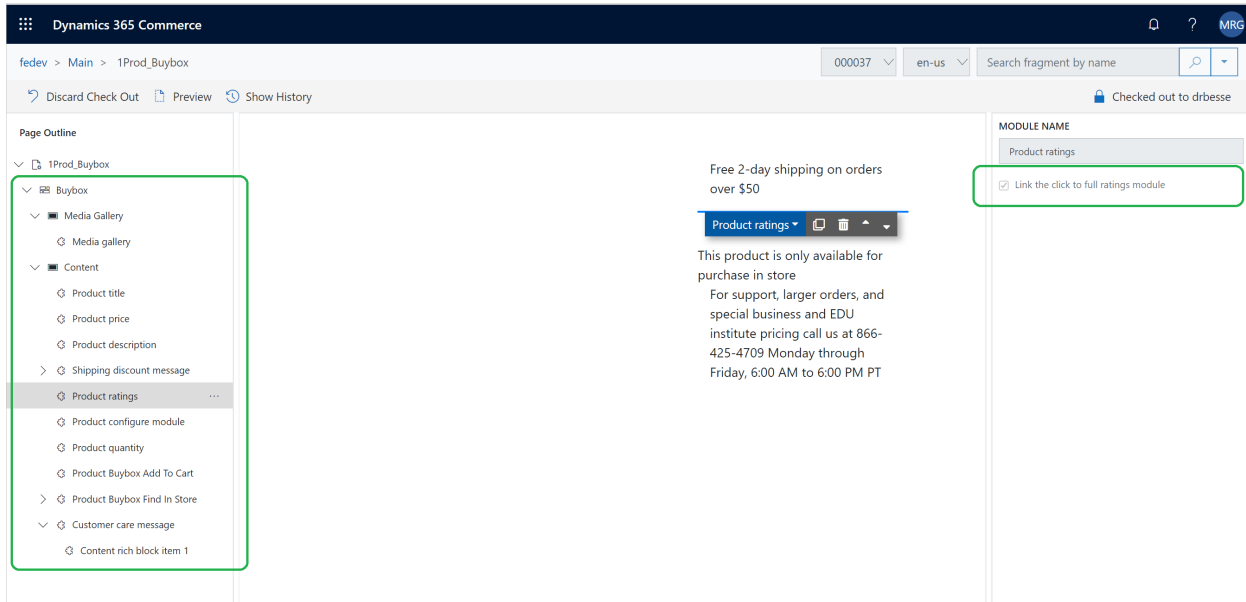
# Link a product rating to the Reviews section of a PDP

A product rating is shown below the product title at the top of PDP. The product rating can be configured so that it's linked to the **Reviews** section of the same PDP.

To link a product rating to the **Reviews** section of the PDP, follow these steps.

1. Open the PDP template.
2. Go to **Buy box container module settings**.
3. Under **Buy box**, select **Product ratings**, and then select the **Link the click to full reviews module** check box.

The following illustration shows what this configuration looks like in Dynamics 365 Commerce.



# Configure the link for the privacy and policy page

To configure the link for the privacy and policy page, follow these steps.

1. Go to **Home > Sites**.
2. Select the name of your site.
3. Go to **Site settings > Extensions**.
4. On the **Routes** tab, under **RNR Privacy and Policy**, select **Add a link**. If a link is already entered, and you want to replace it, select the link.
5. In the **Add a link** dialog box, select the link for the privacy and policy page, and then select **OK**.
6. Select **Save and Publish**.

The following illustration shows what this configuration looks like in Dynamics 365 Commerce.



Site Settings

Channels

Security

**Extensibility**

Global Settings

Save and Publish

Configuration **Routes**

Cart Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAiQ4I

Checkout Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAiQ5n

Account URL ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAiVgd

Account profile URL ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAiXRt

Order details Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAiXRz

Wishlist Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAj33R

Order Confirmation Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAiQ5p

Home Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAiQ42

**RNR Privacy Policy ⓘ**

Page - cms/api/fabrikamsb/identity/IDNMAiQ42

Loyalty Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAj8ij

Loyalty sing up Page Route ⓘ

Page - cms/api/fabrikamsb/identity/IDNMAkOne

## Configure ratings and reviews modules on product details pages

For information on configuring ratings and reviews modules on product details pages, see [Ratings and reviews modules](#).

### Additional resources

[Ratings and reviews overview](#)

[Opt in to use ratings and reviews](#)

Manage ratings and reviews

Configure ratings and reviews modules on product details pages

Sync product ratings in Dynamics 365 Retail

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Sync product ratings in Dynamics 365 Commerce

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to sync product ratings in Microsoft Dynamics 365 Commerce.

## Overview

To consume product ratings in omnichannels, such as at the point of sale (POS) and in call centers, product ratings from the ratings and reviews service must be imported into the Commerce channel database. When product ratings are made available in omnichannels, they can help customers indirectly during their interactions with sales associates.

This topic describes following tasks:

1. Configure **Product ratings sync job** as a batch job to synchronize product ratings from the **Ratings and Reviews service**.
2. Verify that the batch job for product rating synchronization was successful.
3. Make product ratings available at the POS.

## Configure a batch job to synchronize product ratings

### IMPORTANT

Before you start, make sure that version 10.0.6 or later of Dynamics 365 Commerce is installed.

### Initialize the commerce scheduler

To initialize the commerce scheduler, follow these steps.

1. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Initialize commerce scheduler**. Alternatively, search for "Initialize commerce scheduler."
2. In the **Initialize commerce scheduler** dialog box, make sure that the **Delete existing configuration** option is set to **No**, and then select **OK**.

### Verify the RetailProductRating subjob

To verify that the **RetailProductRating** subjob exists, follow these steps.

1. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Scheduler subjobs**. Alternatively, search for "Scheduler subjobs."
2. In the subjob list, find or search for the **RetailProductRating** subjob.

The following illustration shows an example of the subjob details in Commerce.



The screenshot displays the Commerce scheduler interface. At the top, there are navigation options: Edit, New, Delete, Copy scheduler subjob, and OPTIONS. A search bar is present. The left sidebar shows a list of subjobs, with 'RetailProductRating' highlighted and circled in green. The main area shows the configuration for the 'RetailProductRating' subjob. The 'SCHEDULER SUBJOBS' header is also circled in green. The configuration includes a table for 'SCHEDULER SUBJOBS' with columns 'Subjob number' and 'Description', showing 'RetailProductRating' with subjob number 'RetailProductRating'. Below this is the 'Setup' section with options for 'Reference only' (No), 'Pull data' (No), 'Replicate DataAreaID' (No), and 'Channel schema'. The 'Scheduled by' section shows a table with columns 'Job name' and 'Description', listing '1040' with description 'Products'.

#### NOTE

If you don't find the **RetailProductRating** subjob, you might already have run the **Sync product ratings** job and the **1040 CDX** job before you initialized the Commerce scheduler. In this case, follow these steps to run the **Full data sync** job.

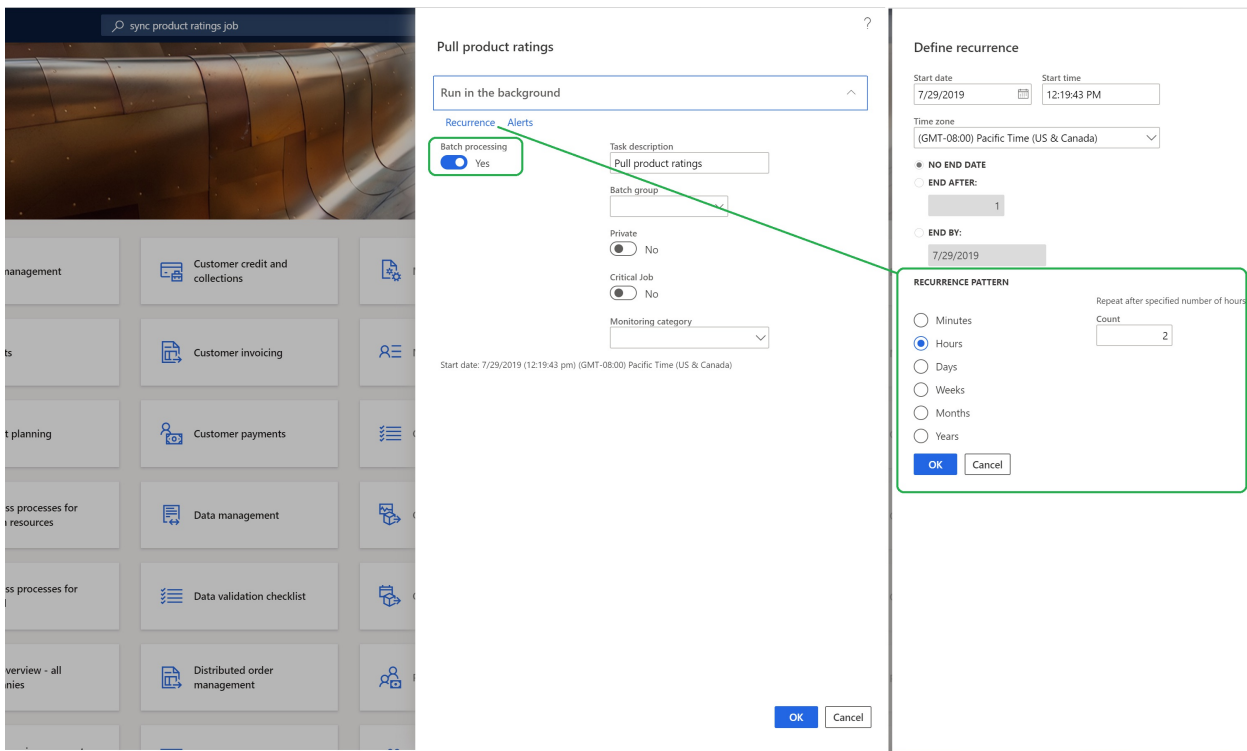
1. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database**. Alternatively, search for "Channel database."
2. Select the channel database to sync.
3. On the action pane, select **Full data sync**.
4. In the **Select a distribution schedule** drop-down dialog box, select **1040 - products**, and then select **OK**.
5. Repeat the steps of the previous procedure to verify that the **RetailProductRating** subjob has been created.

#### Import product ratings

To import product ratings into Commerce from the ratings and reviews service, follow these steps.

1. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Sync product ratings job**. Alternatively, search for "Sync product ratings job."
2. In the **Pull product ratings** dialog box, on the **Run in the background** FastTab, select **Recurrence**.
3. In the **Define recurrence** dialog box, set up a recurrence pattern. (The suggested value is two hours.) Don't schedule a recurrence that is less than one hour.
4. Select **OK**.
5. Set the **Batch process** option to **Yes**. This setting helps guarantee that you will be able to audit the logs and verify the status of batch job runs.
6. Select **OK** to schedule the batch job.

The following illustration shows an example of batch job configuration in Commerce.

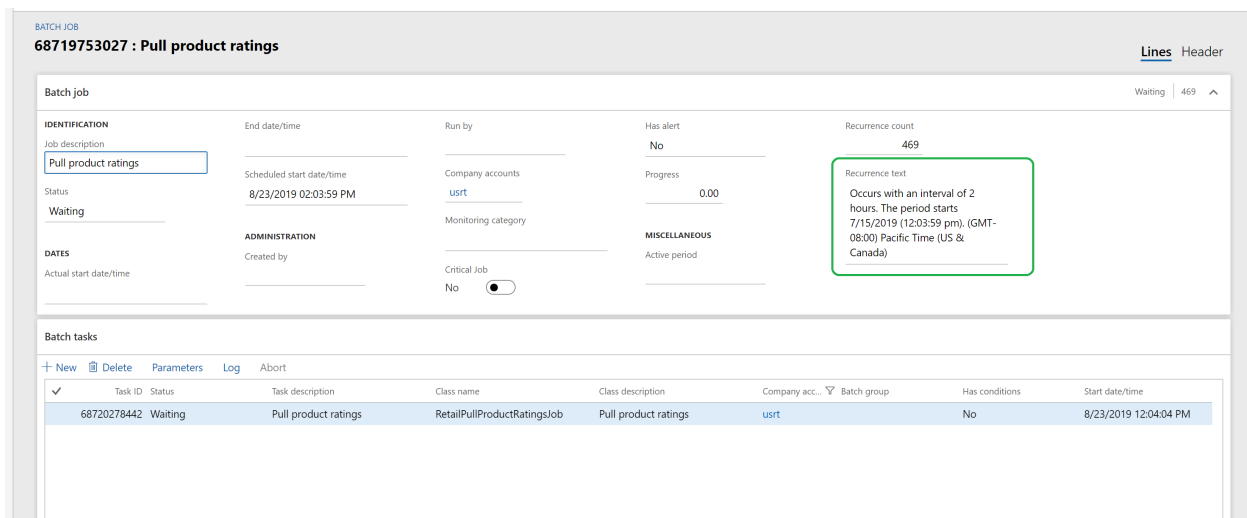


## Verify that the batch job for product rating synchronization was successful

To verify that the **Sync product ratings** batch job was successful, follow these steps.

1. Go to **Retail and Commerce > System administrator > INQUIRIES > Batch jobs** or, if you're using a Commerce-only stock keeping unit (SKU), **Retail and Commerce > INQUIRIES and reports > Batch jobs** instead. Alternatively, search for "Batch jobs."
2. To view the details of the batch job, in the batch job list, in the **Job description** column, search for a description that contains "Pull product ratings."
3. Select the job ID to view the batch job details, such as the scheduled start date/time and the recurrence text.

The following illustration shows an example of the batch job details in Commerce when the batch job is scheduled to run at two-hour intervals.



## Make product ratings available at the POS

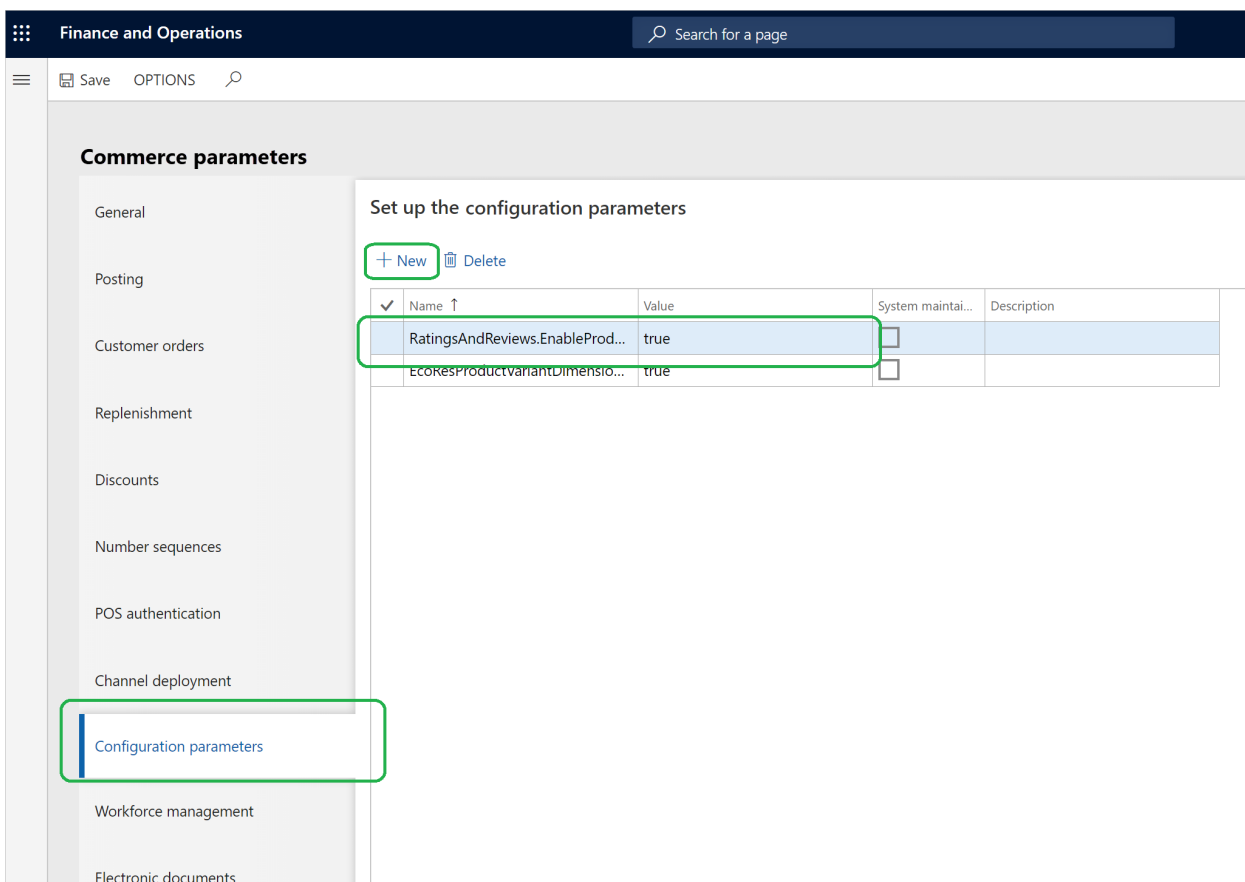
The ratings and reviews solution in Dynamics 365 Commerce is an omnichannel solution. However, products ratings aren't shown at the POS by default. To help customers in stores see ratings and reviews when they are

being helped by sales associates, you must turn on product ratings at the POS.

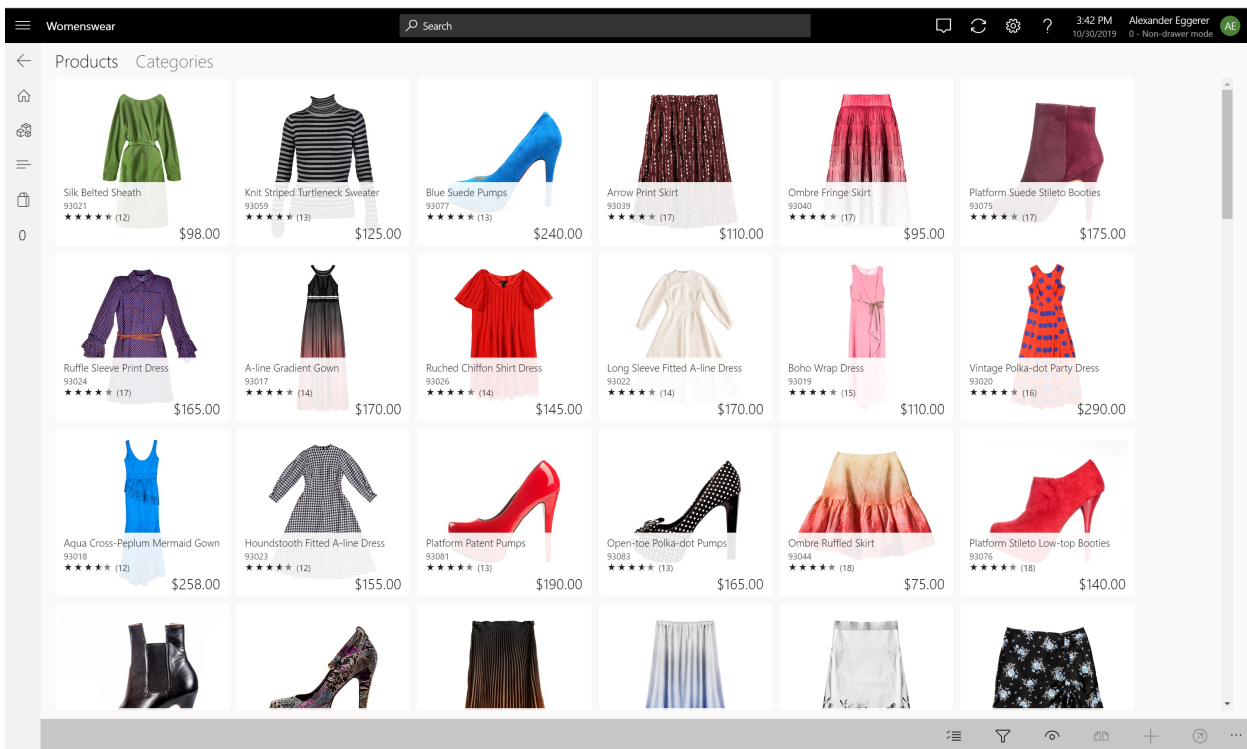
To turn on product ratings at the POS, follow these steps.

1. Go to **Retail and Commerce > Commerce setup > Parameters > Commerce parameters**.  
Alternatively, search for "Commerce parameters."
2. On the **Configuration parameters** tab, select **New**.
3. Enter a name such as **RatingsAndReviews.EnableProductRatingsForRetailStores**, and set the value to **true**.
4. Select **Save**.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**. Alternatively, search for "Distribution schedule."
6. In the job list, select **1110 (Global configuration)**, and then select **Run now**.
7. After the job has successfully run, verify that products ratings are now shown at the POS.

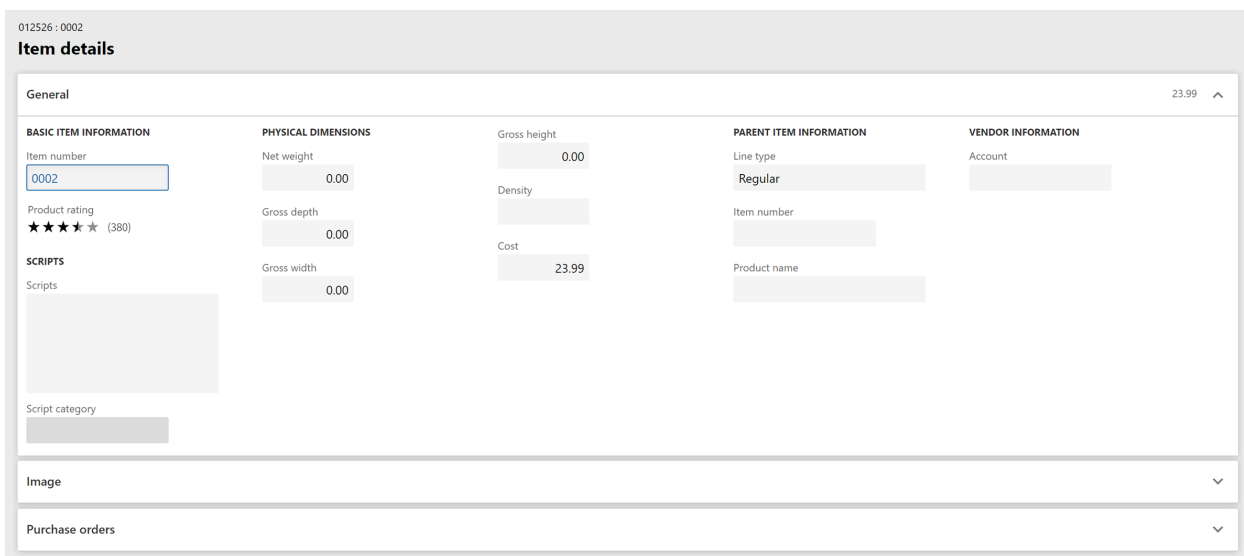
The following illustration shows an example of the configuration of the Commerce parameters to turn on product ratings at the POS.



The following illustration shows an example of product ratings at the POS.



The following illustration shows an example of product ratings in call center channels.



## Additional resources

[Ratings and reviews overview](#)

[Opt in to use ratings and reviews](#)

[Manage ratings and reviews](#)

[Configure ratings and reviews](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce inventory management

2/18/2021 • 5 minutes to read • [Edit Online](#)

When you work with inventory in Microsoft Dynamics 365 Commerce and use any of the Commerce applications that are connected to a Commerce Scale Unit (CSU), it's important to know that the order processing logic in CSU provides limited support for some inventory dimensions and some inventory item types. Commerce applications don't support the full range of item configuration capabilities that is available through the item configuration options in Dynamics 365 Supply Chain Management.

The Commerce applications running on CSU don't support the following product dimensions and item configurations:

- Configuration product dimension and bill of materials (BOM) items (except retail kit products, which use some components of the BOM framework)
- Catch weight items
- Version product dimension-controlled items

The Commerce applications running on CSU do not support the following tracking dimensions:

- Owner dimension
- The point of sale (POS) application can offer limited support for the following dimensions. POS may automatically enter some of the dimensions in inventory transactions, based on the configuration of the warehouse or store setup. POS won't fully support the dimensions in the way that they are supported if a sales transaction is manually entered in Commerce headquarters, however.
- **Warehouse Location** – When they use the new [Inbound operation](#) and [Outbound operation](#) POS operations, users can select a warehouse inventory location to receive items into or ship outbound transfer order items out of. If they use the obsolete **Picking and receiving** operation, limited location management support is available for receiving and shipping outbound transfers. That support is available only if the **Use warehouse management process** option has been turned on for the item and the store warehouse. An inventory location can't currently be used with the **Stock count** operation or the **Inventory lookup** operation.
- **License plate** – License plates are applicable only when the **Use warehouse management process** option has been turned on for the item and the store warehouse. In POS, if inventory is received into a store warehouse by using the **Inbound operation** operation or the **Picking and receiving** operation where the warehouse management process has been turned on, and if the location that has been selected to receive the item into is linked to a location profile that requires license plate control, the POS application systematically applies a license plate to the receiving line. POS users can't change or manage this license plate data. If full management of license plates is required, we recommend that the store use the [warehousing app](#) or the back-office client to manage the receipt of these items.
- **Serial number** – The POS application provides limited support for registration of a single serial number on a sales transaction line for orders that are created in POS and include serialized items. This serial number isn't validated against registered serial numbers that are already in inventory. If a sales order is created in the call center channel or fulfilled through enterprise resource planning (ERP), and multiple serial numbers are registered on a single sales line during the fulfillment process in ERP, those serial numbers can't be applied or validated if a return is processed for the order in POS. When inventory is received by using the **Inbound operation** operation, users can [register or confirm the serial numbers received](#).

- **Batch ID** - The POS application provides limited support during statement posting if a batch-controlled item is sold, but POS users aren't able to define the batch ID that was sold or picked when using the POS application.
- **Inventory status** – For items that use the warehouse management process and require an inventory status, this status field can't be set or modified through the POS application. The default inventory status that is defined in the store warehouse configuration is used when items are received into inventory.

#### NOTE

All organizations must test item configurations through Commerce apps in development or test environments before they deploy those item configurations to production environments. Test your items by using them to perform regular cash-and-carry sales transactions in POS and create customer orders (if applicable) through POS, call center or e-commerce to validate they can be fully supported. You should also test POS fulfillment and inventory processes (such as inventory receiving and order fulfillment operations) before you deploy any new item configurations, to make sure that the POS application can support them. Testing must include running a full statement/order posting process in your test environment and verifying that no posting issues occur when orders for these items are created and posted in Commerce headquarters.

If items are configured in a way that isn't supported by the Commerce applications and appropriate testing isn't done, data failures that aren't easily corrected or may not be able to be corrected at all, may occur.

## Purchase orders

Purchase orders are created in Commerce headquarters. If a store warehouse is included in the purchase order header or on purchase order lines, the lines can be received at the store by using the [Inbound operation](#) operation in POS.

## Transfer orders

Transfer orders can be created in Commerce headquarters, or through either the [Inbound operation](#) or [Outbound operation](#) operation in POS. Use the **Inbound operation** POS operation to create a transfer order request to have inventory sent to the store from another warehouse or store location. Use the **Outbound operation** POS operation to create a transfer order request to have inventory shipped from the store to another warehouse or store location. After a transfer order for a store is created, that store can manage the receipt of inventory for the transfer order through the **Inbound operation** operation in POS. If the store is shipping inventory to another location, the **Outbound operation** operation in POS is used to manage that store's outbound shipment process.

## Stock counts

Stock counts can be either scheduled or unscheduled. Scheduled stock counts are created through Commerce headquarters by creating a Counting journal document that is linked to the store's warehouse. This journal specifies the items that must be counted. The store can then access these predefined counting journals and act on them by using the **Stock count** operation in POS. Store users initiate an unscheduled stock count as it's required when they use the **Stock count** operation in POS. Unlike scheduled stock counts, unscheduled stock counts don't have a predefined list of items. When a stock count of either type is completed in POS, it's committed and sent to the head office. At the head office, the count must then be validated and posted in Commerce headquarters as a separate step.

## Inventory lookup

The product quantity that is currently on hand for multiple stores and warehouses can be viewed on the **Inventory lookup** page. In addition to the current on-hand quantity, the future available-to-promise (ATP)

quantities can be viewed for each store. Select the store to view the ATP quantities for, and then select **Show store availability**. For information about the configuration options that are available, see [Calculate inventory availability for retail channels](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure inventory buffers and inventory levels

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic explains how to configure inventory buffers and inventory levels that determine the messaging about inventory availability on Microsoft Dynamics 365 Commerce sites.

## Overview

Dynamics 365 Commerce headquarters holds inventory data and various channels such as point of sale (POS) applications, e-Commerce storefronts, and other custom integrated applications that pull and push inventory around in an asynchronous manner. Therefore, the available inventory values that are obtained through the on-hand inventory page in Commerce headquarters, through the POS user interface (UI), and through e-Commerce inventory availability APIs aren't always 100-percent accurate in real time.

Instead of showing actual inventory values in e-Commerce storefronts, many retailers prefer just to show messaging about inventory availability status (for example, "Available" or "Out of stock") to inform customers whether an item is available for purchase or potentially out of stock. For this approach, inventory buffers and inventory levels that determine the inventory availability messaging must be made available and configured.

## Prerequisite: Turn on the inventory buffers and inventory levels feature

The feature for inventory buffers and inventory levels is controlled through Feature management in Commerce headquarters. To turn on the feature, follow these steps.

1. Go to **System administration > Workspaces > Feature management**.
2. Search for the **Enable inventory buffers and inventory levels** feature, select its row, and then select **Enable now**.

After the feature is turned on, you can find inventory levels at **Retail and Commerce > Inventory management**.

## Create and configure an inventory level profile

An *inventory level profile* determines whether a given product quantity status is considered in stock, out of stock, or some other custom status. You can create and configure multiple inventory level profiles per legal entity. Each profile consists of a set of inventory levels, and each level is defined by a *range*, a *code*, and a *label*.

- **Range** – Each range is defined by a *start quantity* and an *end quantity*. A quantity value falls in a range if it's more than the start quantity of that range and not more than the end quantity.
- **Code** – A code is an internal abbreviation that represents the level. Customers who directly integrate with the inventory APIs can use codes to build additional logic for a given inventory level. For example, they can turn off the purchase capability for a product when its inventory level code is **OOS** ("out of stock").
- **Label** – A label is a meaningful customer-facing message that conveys an inventory level to customers on an e-Commerce site.

### Create an inventory level profile

To create an inventory level profile, follow these steps.

1. Go to **Retail and Commerce > Inventory management > Inventory levels**.
2. On the Action Pane, select **New**, and then enter values in the **Profile ID** and **Description** fields.



3. On the **Ranges** FastTab, select **Add** to add a new level, and then enter values in the **Start quantity**, **End quantity**, **Code**, and **Label** columns for that level. Repeat this step to add more levels. As you require, you can edit the values in the data grid, or you can select **Delete** to remove a level.
4. On the Action Pane, select **Save**.

When a new profile is created, two inventory levels are automatically initialized:

- **OOS** – The "out of stock" level, where the lower bound of the range is negative infinity. The start quantity and code can't be edited for this level.
- **AVAIL** – The "available" level, where the upper bound of the range is infinity. The end quantity and code can't be edited for this level.

#### **NOTE**

There can be no gaps or overlap between ranges in the profile definition.

You can use the **Translations** button on the Action Pane to configure localized strings for the label message. You must then run the 1110 (**Global configuration**) distribution schedule job to sync the localized strings to channels.

### **Configure an inventory level profile**

You can configure an inventory level profile at either the product category level or the individual product level. When an inventory level profile is configured for a product, the inventory level is determined based on the ranges that are defined in the linked profile. Otherwise, the "available" or "out of stock" inventory level is determined based on whether the product has a positive on-hand quantity.

To configure an inventory level profile for a category, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Commerce product hierarchy**.
2. Select the category to configure an inventory level profile for.
3. On the **Sell product properties** FastTab, select a legal entity.
4. In the **Commerce inventory** section, in the **Inventory level profile** field, select one of the predefined inventory level profiles.

You can use the **Update products** button on the Action Pane to propagate the value of the category-level profile to the category's underlying products. For more information, see [Manage product categories and products](#).

To configure an inventory level profile for a released product, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Released products by category**.
2. Select a product, and then open its product details page.
3. On the **Sell** FastTab, in the **Commerce inventory** section, in the **Inventory level profile** field, select one of the predefined inventory level profiles.

When a new product is created, the **Inventory level profile** field, like many other product-level attributes, will be set to the value that is configured for the category that the product is associated with.

#### **NOTE**

The inventory level profile is a legal entity-specific attribute. For the same category or product, the inventory level profile value can vary across legal entities.

To sync the configurations of inventory level profiles to channels, follow these steps.

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Run the **1040 (Product)** distribution schedule.

## Configure an inventory buffer

The *inventory buffer* is a user-defined value that subtracts the additional quantity of an item from the original quantity to calculate the estimated quantity. This estimated quantity gives retailers a safe buffer so that they don't oversell a product by selling more than its actual on-hand inventory. You can configure the inventory buffer at either the product category level or the individual product level. If no inventory buffer is specified, the default value of **0** (zero) is used.

To configure the inventory buffer for a category, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Commerce product hierarchy**.
2. Select the category to configure the inventory buffer for.
3. On the **Sell product properties** FastTab, select a legal entity.
4. In the **Commerce inventory** section, in the **Inventory buffer** field, enter a positive value.

You can use the **Update products** button on the Action Pane to propagate the value of the category-level buffer to the category's underlying products. For more information, see [Manage product categories and products](#).

To configure the inventory buffer for a released product, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Released products by category**.
2. Select a product, and then open its product details page.
3. On the **Sell** FastTab, in the **Commerce inventory** section, in the **Inventory buffer** field, enter a positive value.

When a new product is created, the **Inventory buffer** field will be set to the value that is configured for the category that the product is associated with.

### NOTE

If both the inventory buffer and inventory level profiles are configured for a product, the product's estimated quantity (that is, the original quantity minus the buffer value) will be used for range calculation to determine the inventory level.

To sync the configurations of inventory buffers to channels, follow these steps.

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Run the **1040 (Product)** distribution schedule.

## Use inventory buffers and inventory levels in e-Commerce scenario

Commerce site builder uses the inventory buffer and inventory level capabilities in Commerce headquarters to determine inventory availability messaging on e-Commerce sites. For more information, see [Apply inventory settings](#).

Alternatively, if you integrate with a third-party e-Commerce solution, you can use the **GetEstimatedAvailability** and **GetEstimatedProductWarehouseAvailability** APIs to show inventory availability for a product in your e-Commerce scenario. For more information about these APIs, see [Calculate inventory availability for retail channels](#).

The introduction of inventory buffers and inventory levels enables these APIs to return inventory level codes and label messages that are determined based on total available and available physical values. The APIs can be

further configured to determine whether the inventory quantity is returned together with the message, and whether the available quantity is reduced by the inventory buffer value.

To configure the response of the product availability APIs, follow these steps.

1. Go to **Retail and commerce > Headquarters setup > Parameters > Commerce parameters**.
2. In the **Store inventory** section, on the **Inventory** tab, in the **Product availability APIs for e-Commerce** field, select a value.
3. To apply the settings to channels, run the **1110 (Global configuration)** distribution schedule job.

## Additional resources

[Manage product categories and products](#)

[Apply inventory settings](#)

[Calculate inventory availability for retail channels](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Work with serialized items in the POS

2/18/2021 • 11 minutes to read • [Edit Online](#)

Many retailers sell products that require serial control. These products are referred to as *serialized items*. Some retailers might want to maintain serial numbers in store or warehouse inventory for tracking purposes. Other retailers might want to capture serial numbers during the sales process, for service and warranty purposes. This topic explains how you can manage serialized items in the Microsoft Dynamics 365 Commerce point of sale (POS) application.

## Serial number configurations

An item is considered a serialized item if it's assigned a tracking dimension group that's set up to allow for serial numbers. In Commerce headquarters, on the **Tracking dimension groups** page, select the **Active** option to enable serial numbers for the inventory process, or select the **Active in sales process** option to enable serial numbers for the sales process.

On the **Tracking dimensions** FastTab, turn on the **Blank receipt allowed** parameter to allow serial number to be an optional input during the inventory receipt process of serialized item. Turning off this parameter enforces serial number to be a required input. Similarly, the **Blank issue allowed** parameter controls whether serial number is required during the inventory shipment process.

### NOTE

To use the **Inbound inventory** and **Outbound inventory** POS operations to register or validate serial numbers against a serialized item, you must configure that item to be assigned a tracking dimension group that's set up to allow for serial numbers for the **Active** option, not the **Active in sales process** option.

## Serial number management page

In the **Inbound inventory** and **Outbound inventory** POS operations, if the item that is being selected, received, or shipped is a serialized item, the **Details** pane contains a **Manage serial number** option that links to the **Serial number management** page where you can register or validate serial numbers for the item. Alternatively, you can open the **Serial number management** page either by selecting the **Serial number** action on the app bar of the order details view, or by selecting the **Manage serial number** option in the dialog box that prompts you during the receiving or shipping process.

The **Serial number management** page lists all open serial number lines that are pending registration or validation. There might be two tabs on this page: one for the current item and another for all serialized items in the order.

The **Status** field on the **Serial number management** page provides information about the current stage that each serial number is in:

- **Not registered** – The serial number hasn't been provided, or the preregistered serial number hasn't yet been validated (in the receiving process).
- **Registering** – The serial number has been registered and saved locally to the store's channel database, or the preregistered serial number has been validated. Only serial numbers that have a status of **Registering** will be submitted to Commerce headquarters when you finish receiving or fulfillment.

## Receive serialized items

The **Inbound inventory** POS operation lets users perform the following tasks for serialized items:

- Register serial numbers against serialized items when those items are received in the store via a purchase order.
- Validate preregistered serial numbers against serialized items when those items are received in the store via a purchase order or transfer order.

### **Register serial numbers against serialized items**

For a purchase order, you'll be prompted with a dialog box with the **Manage serial number** option during the receiving process of a serialized item. You can select that option to open the **Serial number management** page and start to register serial numbers. You can also skip this step during the receiving process and provide the input later, before the receipt is posted.

By default, the tab for the current item is shown. All serial number lines have an empty serial number value and a status of **Not registered**. You can scan serial number bar codes, or you can select **Serial number** on the app bar to continuously enter serial numbers. The serial numbers that you enter appear in the list, and their status is changed to **Registering**. The maximum number of serial numbers that you can register in the list is equal to the receiving quantity. If you make a mistake, you can select **Edit** or **Clear** in the **Details** pane to make changes to the serial numbers that you entered.

You can also register serial numbers on the **All serialized items** tab of the **Serial number management** page. In the list, select the item that you want to register serial numbers against.

### **Validate serial numbers on serialized items**

For a transfer order, the outbound side must preregister serial numbers on the serialized items during the shipment process. For a purchase order, the supplier might provide serial number information through an Advance Shipment Notice (ASN), and you can preregister the numbers on the items that will be shipped. In both cases, the serial numbers are known before the receipt. Therefore, on the inbound side, you just have to validate that you received what you were supposed to receive.

To validate serial numbers, you can open the **Serial number management** page either during the receiving process or at any time before the receipt is posted. For each serialized item that has preregistered serial numbers, all the serial numbers are automatically set to an initial status of **Not registered** on this page. To validate serial numbers, you can scan or enter them. As serial number are entered, the application validates whether they match preregistered serial numbers. If they match, their status is changed to **Registering**. Otherwise, you receive an error message. Alternatively, you can directly select a serial number, then select the **Validate serial number** option in the **Details** pane to quickly mark that serial number as validated. The maximum number of serial numbers that you can validate in the list is equal to the receiving quantity.

You can also validate serial numbers on the **All serialized items** tab of the **Serial number management** page. In the list, select the item that you want to validate serial numbers against.

## **Ship serialized items**

You can use the **Outbound inventory** POS operation to register serial numbers against serialized items when shipping them out of current store via a transfer order.

### **Register serial numbers against serialized items**

For a transfer order, you'll be prompted with a dialog box with the **Manage serial number** option during the shipping process of a serialized item. You can select the option to open the **Serial number management** page and start to register serial numbers. You can also skip this step during the shipping process and provide the input later, before the shipment is posted.

By default, the tab for the current item is shown. All serial number lines have an empty serial number value and a status of **Not registered**. You can scan serial number bar codes, or you can select **Serial number** on the app

bar to continuously enter serial numbers. The serial numbers that you enter appear in the list, and their status is changed to **Registering**. The maximum number of serial numbers that you can register in the list is equal to the shipping quantity. If you make a mistake, you can select **Edit** or **Clear** in the **Details** pane to make changes to the serial numbers that you entered.

You can also register serial numbers on the **All serialized items** tab on the **Serial number management** page. In the list, select the item that you want to register serial numbers against.

Optionally, you can enable serial number availability validation during the serial number registration against an outbound transfer order. With this validation, if you attempt to ship a serial number that is not available in the inventory of the shipping store, you will receive an error message and must provide a different number.

To enable such validation, as a prerequisite, you need to schedule the following jobs to run on a recurring basis:

- **Retail and Commerce > Retail and Commerce IT > Products and inventory > Product availability with tracking dimensions**
- **Retail and Commerce > Distribution schedules > 1130 (Product availability)**

## Sell serialized items in POS

While the POS application has always supported selling serialized items, in Commerce version 10.0.17 and later, organizations can enable functionality that enhances the business logic that's triggered when selling products that are configured for serial number tracking.

When the **Enhanced serial number validation in POS order capture and order fulfillment** feature is enabled, the following product configurations are evaluated when selling serialized products in POS:

- **Serial type** setup for the product (**active** or **active in sales**).
- **Blank issue allowed** settings for the product.
- **Physical negative inventory** settings for the product and/or the selling warehouse.

### Active serial configurations

When items are sold in POS that are configured with an **Active** serial number tracking dimension, POS initiates validation logic that prevents users from completing the sale of a serialized item with a serial number that can't be found in the selling warehouse's current inventory. There are two exceptions to this validation rule:

- If the item is also configured with **Blank issue allowed** enabled, users can skip the entry of the serial number and sell the item with no serial number designation.
- If the item and/or the selling warehouse is configured with **Physical negative inventory** enabled, the application accepts and sells a serial number that can't be confirmed to be in inventory at the warehouse that it's being sold against. This configuration allows the inventory transaction for that specific item/serial number to go negative, and therefore the system will allow for sales of unknown serial numbers.

#### IMPORTANT

To ensure that the POS application can properly validate whether the serial numbers being sold for **Active** serial type items are in the selling warehouse's inventory, it's required that organizations run the **Product availability with tracking dimensions** job in Commerce headquarters and the accompanying **1130** product availability distribution job through Commerce headquarters on a frequent basis. As new serialized inventory is received into selling warehouses, in order for the POS to validate inventory availability of serial numbers being sold, the inventory master must frequently update the channel database with the most up-to-date inventory availability data. The **Product availability with tracking dimensions** job takes a current snapshot of master inventory, including serial numbers, for all company warehouses. The **1130** distribution job takes that inventory snapshot and shares it with all configured channel databases.

### Active in sales process serial configurations

Items configured with the serial dimension as **Active in sales process** don't go through any inventory validation logic, as this configuration implies that the inventory serial numbers aren't pre-registered into stock and the serial numbers are only captured at the time of sale.

If **Blank issue allowed** is also configured for **Active in sales process** configured items, the serial number entry can be skipped. If **Blank issue allowed** isn't configured, the application requires the user to enter a serial number, even though it won't be validated against available inventory.

### Apply serial numbers during creation of POS transactions

The POS application immediately prompts users for serial number capture when selling a serialized item, but the application allows users to skip the entry of serial numbers up to a certain point in the selling process. When the user begins to capture payment, the application enforces and requires serial number entry for any items that aren't configured to be fulfilled through future shipments or pickups. Any serialized items configured for cash and carry or carryout fulfillment require the user to capture the serial number (or agree to leave it blank if the item configuration allows for it) before completing the sale.

For serialized items sold for future pickup or shipment, POS users can skip entering the serial number initially and still complete the customer order creation.

#### NOTE

When selling or fulfilling serialized products through the POS application, a quantity of "1" is enforced for the serialized items on the sales transaction. This is a result of how the serial number information is tracked on the sales line. When selling or fulfilling a transaction for multiple serialized items through POS, each sales line must be only configured with a quantity of "1".

### Apply serial numbers during customer order fulfillment or pickup

When fulfilling customer order lines for serialized products using the **Order Fulfillment** operation in POS, POS enforces the capture of the serial number before final fulfillment. Therefore, if a serial number wasn't provided during the initial order capture, it must be captured during the pick, pack, or ship processes in POS. A validation is done at each step and the user will only be asked for serial number data if it's missing or no longer valid. For example, if a user skips the pick or pack steps and immediately initiates a shipment, and a serial number hasn't been registered for the line, POS will require the serial number to be entered before completion of the final invoice step. When enforcing the capture of the serial number during order fulfillment operations in POS, all rules mentioned previously in this topic still apply. Only serialized items configured as **Active** go through a serial number inventory stock validation. Items configured as **Active in sales process** won't get validated. If **Physical negative inventory** is allowed for **Active** products, any serial number will be accepted, regardless of stock availability. For both **Active** and **Active in sales process** items, if **Blank issue allowed** is configured, a user can leave serial numbers blank if desired during the pick, pack, and ship steps.

Validations for serial numbers will also occur when a user performs the pick up operations on customer orders in POS. The POS application doesn't allow a pick up to be finalized on a serialized product unless it passes the validations as mentioned earlier. The validations are always based on the product's tracking dimension and selling warehouse configurations.

## Additional resources

[Inbound inventory operation in POS](#)

[Outbound inventory operation in POS](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Push products from distribution center to store using buyer's push

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through the steps to create and process a Buyer's push to distribute products from one location to one or many stores. The user can define multiple configurations and have the system suggest how to distribute the products, or manually enter where the products are distributed to and how much gets distributed to each store. This procedure doesn't include setup of data that can be used in the Buyer's push, such as replenishment rules, organizational hierarchies, and store weights. This procedure uses the USRT demo company.

1. Go to Buyer's push.
2. Click New.
3. In the Description field, type a value.
4. In the Site field, enter or select a value.
5. In the Warehouse field, enter or select a warehouse that has products with on-hand quantities.
6. Click Add.
7. In the list, mark the selected row.
8. In the Item number field, enter or select a product.
9. Click Add.
10. In the list, mark the selected row.
11. In the Item number field, enter or select a variant product.
  - When entering a variant product, lines will be created for each variant.
12. In the list, mark a row.
13. In the Pushed quantity field, type how many of the selected product you want to distribute.
14. In the Additional quantity to push field, enter the quantity of the products that have available quantity to distribute.
15. In the Distribution field, enter 'Location weight'.
  - You can select the other types to use other rules for the distribution.
16. In the Replenishment hierarchy field, select a value.
17. Select Yes in the Respect assortments field.
18. Click Calculate quantities and review the quantities that are added to the rows in the Warehouse section.
19. Click Create order.
20. Click Yes.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up rules and parameters for cross docking and buyer's push

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates the steps to create Replenishment rules. Replenishment rules can be used to control how products are distributed to stores when using Cross-docking and Buyer's push. Replenishment rules can be set up for stores or store groups. The weight defined for each line in a rule will control how the quantities of products will get distributed between the stores when using Replenishment rules as the distribution method in Cross-docking or Buyer's push. This procedure uses the USRT demo company.

1. Go to Replenishment rules.
2. Click New.
3. In the Replenishment rule field, type a value.
4. In the Description field, type a value.
5. Click Save.
6. Click Add.
7. In the list, mark the selected row.
  - You can choose Replenishment hierarchy or Channel for the type. The value controls whether the selection in Name will be a hierarchy of channels or a specific channel. For this example, leave it set as Replenishment hierarchy.
8. In the Name field, select a value.
  - The default weight value is populated from the weight defined on the warehouse. This weight can be used for the Replenishment rule or you can enter a new weight in the Weight field.
9. In the Weight field, enter a number.
10. Click Add.
11. In the list, mark the selected row.
12. In the Type field, select 'Channel'.
13. In the Name field, enter or select a value.
14. In the Weight field, enter a number.
15. Click Save.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Inbound inventory operation in POS

2/18/2021 • 20 minutes to read • [Edit Online](#)

In Microsoft Dynamics 365 Commerce version 10.0.10 and later, inbound and outbound operations in the point of sale (POS) replace the picking and receiving operation.

## NOTE

In Commerce version 10.0.10 and later, any new features in the POS application that are related to receiving store inventory against purchase orders and transfer orders will be added to the **Inbound operation** POS operation. If you're currently using the picking and receiving operation in POS, we recommend that you develop a strategy for moving from that operation to the new inbound and outbound operations. Although the picking and receiving operation won't be removed from the product, there will be no further investments in it, from a functional or performance perspective, after version 10.0.9.

## Prerequisite: Configure an asynchronous document framework

The inbound operation includes performance improvements to ensure that users who have high volumes of receipt postings across many stores or companies, and large inventory documents, can process those documents to Commerce Headquarters without experiencing time-outs or failures. These improvements require use of an asynchronous document framework.

When an asynchronous document framework is used, you can commit inbound document changes from POS to Commerce Headquarters and then move on to other tasks while the processing to Commerce Headquarters occurs in the background. You can check the status of a document through the **Inbound operation** document list page in POS to make sure that posting was successful. In the POS application, you can also use the inbound operation active document list to see any documents that could not be posted to Commerce Headquarters. If a document fails, POS users can make corrections to it and then try again to process it to Commerce Headquarters.

## IMPORTANT

The asynchronous document framework must be configured before a company tries to use the inbound operation in POS.

To configure an asynchronous document framework, complete the following procedures.

### Create and configure a number sequence

1. Go to **Organization administration > Number sequences > Number sequences**.
2. On the **Number sequences** page, create a number sequence.
3. In the **Number sequence code** and **Name** fields, enter user-defined values.
4. On the **References** FastTab, select **Add**.
5. In the **Area** field, select **Commerce parameters**.
6. In the **Reference** field, select **Retail document operation identifier**.
7. On the **General** FastTab, in the **Setup** section, set the **Continuous** option to **No** to ensure that there are no performance issues.

### Create and schedule two batch jobs for the document processing and monitoring tasks

#### NOTE

In Commerce version 10.0.13 and later, you don't have to configure these batch jobs through the batch job framework. The batch processes can be configured from the **Retail and Commerce > Retail and Commerce IT** menu. Use the **Retail document operation monitor** and **Retail document operation processing** menu options to configure the batch jobs.

The batch jobs that you create will be used to process documents that fail or time out. They will also be used when the number of active inventory documents that are being processed from POS exceeds a system-configured value.

1. Go to **System administration > Inquiries > Batch jobs**.
2. On the **Batch job** page, create two batch jobs:
  - Configure one job to run the **RetailDocumentOperationMonitorBatch** class.
  - Configure the other job to run the **RetailDocumentOperationProcessingBatch** class.
3. Schedule the new batch jobs to run on a recurring basis. For example, set the schedule so that the jobs are run every five minutes.

## Prerequisite: Add Inbound operation to the POS screen layout

Before your organization can use the inbound operation functionality, it must configure the **Inbound operation** POS operation on one or more of your [POS screen layouts](#). Before you deploy the new operation in a production environment, make sure that you thoroughly test it and train your users to use it.

## Overview

The inbound operation lets POS users perform the following tasks:

- Receive inventory into store stock from either confirmed purchase order documents or shipped transfer order documents.
- View information about historical inventory receipts for a period of seven days after the document has been fully received.
- Create new inbound transfer order requests.

When the inbound operation is started from the POS application, a list page view appears. This view shows open purchase order and transfer order documents that have inventory lines that are scheduled to be received by the current store. To find and select a specific document, users can scroll the list or use the search feature.

The inbound inventory document list has three tabs:

- **Active** – This tab shows documents that are fully or partially open, and that contain lines or quantities on lines that must still be received.
- **Draft** – This tab shows new inbound transfer order requests that the store has created. However, the documents have only been saved locally. They haven't yet been submitted to Commerce Headquarters for processing.
- **Complete** – This tab shows a list of purchase order or transfer order documents that the store has fully received during the last seven days. This tab is for informational purposes only. All the information about the documents is read-only data for the store.

When you view documents on any of the tabs, the **Status** field can help you understand the stage that the document is in.

- **Draft** – The transfer order document has only been saved locally to the store's channel database. No

information about the transfer order request has yet been submitted to Commerce Headquarters.

- **Requested** – The purchase order or transfer order has been created in Commerce Headquarters, and is fully open. No receipts have yet been processed against the document. For documents of the purchase order document type, receiving can begin at any time while the status is **Requested**.
- **Partially shipped** – The transfer order document has one or more lines or partial line quantities that have been posted as shipped by the outbound warehouse. These shipped lines are available to be received through the inbound operation.
- **Fully shipped** – The transfer order has had all its lines and full line quantities posted as shipped by the outbound warehouse. The whole document is available to be received through the inbound operation.
- **Partially received** – Some of the lines or line quantities on the purchase order or transfer order document have been received by the store, but some lines remain open.
- **Fully received** – All lines and quantities on the purchase order or transfer order document have been fully received. The documents are accessible only on the **Complete** tab and are read-only by store users.
- **In progress** – This status is used to inform device users that the document is being actively worked on by another user.
- **Paused** – This status is shown after **Pause receiving** is selected to temporarily stop the receiving process.
- **Processing in HQ** – The document was submitted to Commerce Headquarters from the POS application, but it hasn't yet been successfully posted to Commerce Headquarters. The document is going through the asynchronous document posting process. After the document is successfully posted to Commerce Headquarters, its status should be updated to **Fully received** or **Partially received**.
- **Processing failed** – The document was posted to Commerce Headquarters and rejected. The **Details** pane shows the reason for the posting failure. The document must be edited to fix data issues, and then it must be resubmitted to Commerce Headquarters for processing.

When you select a document line in the list, a **Details** pane appears. This pane shows additional information about the document, such as shipment and date information. A progress bar shows how many items must still be processed. If the document wasn't successfully processed to Commerce Headquarters, the **Details** pane also shows error messages that are related to the failure.

In the document list page view, you can select **Order details** on the app bar to view the document details. You can also activate receipt processing on eligible document lines.

In the document list page view, you can also create a new inbound transfer order request for a store. The documents remain in **Draft** status, and they can be adjusted or deleted until they are submitted to Commerce Headquarters for processing. After they are submitted to Commerce Headquarters, the transfer order lines can no longer be changed from the POS application.

## Receiving process

After you select a purchase order or transfer order document on the **Active** tab, you can select **Order details** to begin the receiving process.

By default, the **Receiving now** view is shown. This view is optimized for bar code scanning. It can be used to build a list of the items that have been scanned, so that those items can be received. To begin the receiving process, you can start to scan item bar codes.

As item bar codes are scanned in the **Receiving now** view, the application validates the items against the selected purchase or transfer order document, to make sure that each scanned item matches a valid item on the document. In the **Receiving now** view, each scan of a bar code is assumed to represent the receipt of a quantity of one unit, unless a quantity is embedded in the bar code. You can repeatedly scan bar codes in this view to build a list of all the items and quantities for the receipt.

### Example scenario

A user receives a purchase order that contains 10 units of bar code 5657900266. The user can scan that bar code 10 times to update the **Receiving now** field by one unit per scan. When the user has completed the scans, the **Receiving now** field for the item's line will show that a quantity of 10 was received.

Alternatively, in a scenario where the item quantity is large, the user might prefer to manually enter the quantity instead of scanning the bar code for each item that is received. In this case, the user can scan the bar code one time to add the item to the **Receiving now** list. The user can then select the associated line in the **Receiving now** view and then, in the **Details** pane that appears on the right side of the page, update the **Receiving quantity** field for the item.

Although the **Receiving now** view is optimized for bar code scanning, users can also select **Receive product** on the app bar, and then enter the item ID or bar code data through a dialog box. After the item that was entered is validated, the user is prompted to enter the receipt quantity.

The **Receiving now** view provides a focused way for users to see which products they are receiving. Alternatively, the **Full order list** view can be used. This view shows the whole list of document lines for the selected purchase or transfer order document. Users can manually select lines manually in the list and then, in the **Details** pane, update the **Receiving quantity** field for the selected line. In the **Full order list** view, users can scan bar codes, or they can use the **Receive product** function to enter the item ID or bar code, and data about the received quantity, without first having to select the matching item line in the list.

### **Over-receiving validations**

Validations occur during the receiving process for the document lines. They include validations for over-delivery. If a user tries to receive more inventory than was ordered on a purchase order, but either over-delivery isn't configured or the amount that is received exceeds the over-delivery tolerance that is configured for the purchase order line, the user receives an error and isn't allowed to receive the excess quantity.

Over-receiving isn't permitted for transfer order documents. Users will always receive errors if they try to receive more than was shipped for the transfer order line.

### **Close purchase order lines**

You can close the remaining quantity on an inbound purchase order during the receiving process if the shipper has confirmed that they can't ship the full quantity that was requested. To do so, the company must be configured to allow underdelivery of purchase orders. Additionally, an underdelivery tolerance percentage must be defined for the purchase order line.

To configure the company to allow underdelivery of purchase orders, in Commerce headquarters, go to **Procurement and sourcing > Setup > Procurement and sourcing parameters**. On the **Delivery** tab, turn on the **Accept underdelivery** parameter. Then run the **1070 (Channel configuration)** distribution schedule job to sync the setting changes to channels.

Underdelivery tolerance percentages for a purchase order line can be predefined on products as part of the product configurations in Commerce headquarters. Alternatively, they can be set or overwritten on a specific purchase order in Commerce headquarters.

After an organization completes the purchase order underdelivery configurations, POS users will see a new **Close remaining quantity** option in the **Details** pane when an inbound purchase order line is selected in the **Inbound inventory** operation. If the user closes the remaining quantity, POS performs a validation to verify whether the quantity being closed is within the underdelivery tolerance percentage defined on the purchase order line. If the underdelivery tolerance is exceeded, an error message is displayed and the user won't be able to close the remaining quantity until the previously received quantity plus the **Receiving now** quantity meets or exceeds the minimal quantity that needs to be received based on the underdelivery tolerance percentage.

With **Close remaining quantity** option turned on for a purchase order line, when the user completes the receipt by using the **Finish receiving** action, a closure request is also sent to Commerce headquarters, and any unreceived quantity of this order line will be cancelled. At that point the line is considered fully received.

## Receiving location-controlled items

If the items that are being received are location-controlled, users can select the location where they want to receive the items during the receiving process. We recommend that you configure a default receiving location for your store warehouse, to make this process more efficient. Even if a default location is configured, users can override the receiving location on selected lines as they require.

The operation respects the **Blank receipt allowed** configuration on the **Location** storage dimension and doesn't require that a location dimension be entered if blank receipt is configured. If blank receipt locations aren't allowed for an item, the POS application shows an error and requires that a location be entered before the receipt can be posted.

## Receive all

As you require, you can select **Receive all** on the app bar to quickly update the **Receiving now** quantity for all the document lines to the maximum value that is available to be received for those lines.

## Receipt of unplanned items on purchase orders

In Commerce version 10.0.14 and later, users can receive a product that was not originally on the purchase order. To enable this functionality, turn on **Add lines to Purchase Order during Point of Sale receiving**.

This feature only works for purchase order receiving. It's not possible to receive items against transfer orders when the items weren't previously ordered and shipped from the outbound warehouse.

Users can't add new products to the purchase order during POS receiving if purchase order [change management workflow](#) is enabled in Commerce headquarters (HQ). To enable change management, all changes to a purchase order must first be approved before receiving is allowed. Because this process allows a receiver to add new lines to the purchase order, receiving will fail if the change management workflow is enabled. If change management is enabled for all purchase orders or for the vendor linked to the purchase order actively being received in POS, the user can't add new products to the purchase order during receiving in POS.

The functionality that enables adding lines can't be used as a workaround for receiving additional quantities of products already on the purchase order. Over-receiving is managed through the standard [over-receiving](#) settings for the product line on the purchase order.

If **Add lines to Purchase Order during Point of Sale receiving** is enabled and a user is receiving with the **Inbound operation** in POS, if the user scans or keys a product barcode or product number that isn't recognized as an item on the current purchase order, but is recognized as a valid item, the user receives a message about adding the item to the purchase order. If the user adds the item to the purchase order, the quantity entered in **Receiving now** is considered the ordered quantity for the purchase order line.

When the purchase order receipt is complete and submitted to HQ for processing, the added lines are created on the purchase order master document. On the purchase order line in HQ, there will be an **Added by POS** flag on the **General** tab of the purchase order line. The **Added by POS** flag indicates that the purchase order line was added by the POS receiving process and was not a line that was on the purchase order prior to receiving.

## Cancel receiving

You should use the **Cancel receiving** function on the app bar only if you want to back out of the document and don't want to save any changes. For example, you initially selected the wrong document and don't want any of the previous receiving data saved.

## Pause receiving

If you're receiving inventory, you can use the **Pause receiving** function if you want to take a break from the receiving process. For example, you might want to perform another operation from the POS, such as ringing up a customer sale, or delay posting of the receipt.

When you select **Pause receiving**, the document's status is changed to **Paused**. Therefore, users will know that data has been entered for the document, but the document hasn't yet been committed. When you're ready to

resume the receiving process, select the paused document, and then select **Order details**. Any **Receiving now** quantities that were previously saved are retained and can be viewed from the **Full order list** view.

## Review

Before the final commitment of the receipt to Commerce headquarters (HQ), you can use the review functionality to validate the inbound document. The review will alert you to any missing or incorrect data that may cause processing failure and provide you the opportunity to correct problems before submitting the receipt request. To enable the **Review** function on the app bar, enable the **Enable validation in POS inbound and outbound inventory operations** feature through the **Feature management** workspace in Commerce headquarters (HQ).

The **Review** function validates the following issues in an inbound document:

- **Over-receiving** – the receiving now quantity is greater than the ordered quantity. The severity of this issue is determined by the overdelivery configuration in Commerce headquarters (HQ).
- **Under-receiving** – the receiving now quantity is less than the ordered quantity. The severity of this issue is determined by the underdelivery configuration in Commerce headquarters (HQ).
- **Serial number** – the serial number is not provided or validated for a serialized item that requires serial number to be registered in inventory.
- **Location not set** – the location is not specified for a location-controlled item where blank location is not allowed.
- **Deleted lines** – the order has lines deleted by a Commerce headquarters (HQ) user that is not known to POS application.

Set the **Enable automatic validation** parameter to **Yes** in **Commerce parameters > Inventory > Store inventory** to have the validation executed automatically when **Finish receiving** is selected.

## Finish receiving

When you've finished entering all the **Receiving now** quantities for products, you must select **Finish receiving** on the app bar to process the receipt.

When users complete a purchase order receipt, they are prompted to enter a value in the **Receipt number** field, if this functionality is configured. Typically, this value is equivalent to the identifier of the vendor packing slip. The **Receipt number** data will be stored in the Product receipt journal in Commerce Headquarters. Receipt numbers aren't captured for transfer order receipts.

When asynchronous document processing is used, the receipt is submitted through an asynchronous document framework. The time that it takes for the document to be posted depends on the size of the document (the number of lines) and the general processing traffic that is occurring on the server. Typically, this process occurs in a matter of seconds. If document posting fails, the user is notified through the **Inbound operation** document list, where the document status will be updated to **Processing failed**. The user can then select the failed document in POS to view the error messages and the reason for the failure in the **Details** pane. A failed document remains unposted and requires that the user return to the document lines by selecting **Order details** in POS. The user must then update the document with corrections, based on the errors. After a document is corrected, the user can try again to process it by selecting **Finish fulfillment** on the app bar.

## Create an inbound transfer order

From POS, users can create new transfer order documents. To begin the process, select **New** on the app bar while you're in the main **Inbound operation** document list. You're then prompted to select a **Transfer from** warehouse or store that will provide the inventory to your store location. The values are limited to the selection that is defined in the configuration of the store's fulfillment group. In an inbound transfer request, your current store will always be the **Transfer to** warehouse for the transfer order. That value can't be changed.

You can enter values in the **Ship date**, **Receive date**, and **Mode of delivery** fields as you require. You can also



add a note that will be stored together with the transfer order header, as an attachment to the document in Commerce Headquarters.

After the header information is created, you can add products to the transfer order. To start the process of adding items and requested quantities, select **Add product**. In the **Details** pane, you can also add a line-specific note to the journal lines. These notes will be stored as a line attachment.

After lines are entered on the inbound transfer order, you must select **Save** to save the document changes locally or **Submit request** to submit the order details to Commerce Headquarters for further processing. If you select **Save**, the draft document is stored in the channel database, and the outbound warehouse can't run the document until it has been successfully processed via **Submit request**. You should select **Save** only if you aren't ready to commit the request to Commerce Headquarters for processing.

If a document is saved locally, it can be found on the **Drafts** tab of the **Inbound operation** document list. While a document is in **Draft** status, you can edit it by selecting **Edit**. You can update, add, or delete lines as you require. You can also delete the whole document while it's in **Draft** status, by selecting **Delete** on the **Drafts** tab.

After the draft document is successfully submitted to Commerce Headquarters, it appears on the **Active** tab and has a status of **Requested**. At this point, users in the inbound store or warehouse can no longer edit the requested inbound transfer order document. Only users in the outbound warehouse can edit the document, by selecting **Outbound operation** in the POS application. The editing lock ensures that no conflicts occur because an inbound requestor changes the transfer order at the same time that the outbound shipper is actively picking and shipping the order. If changes are required from the inbound store or warehouse after the transfer order has been submitted, the outbound shipper should be contacted and asked to enter the changes.

After the document is in **Requested** status, it's visible on the **Active** tab. However, it can't yet be received by the inbound store or warehouse. After the outbound warehouse has shipped some or all of the transfer order, the inbound store or warehouse can post receipts in POS. When the outbound side processes the transfer order documents, their status is updated from **Requested** to **Shipped** or **Partially Shipped**. After the documents are in **Shipped** or **Partially Shipped** status, the inbound store or warehouse can post receipts against them using the inbound operation receiving process.

## Related topics

[Outbound inventory operation in POS](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Outbound inventory operation in POS

2/18/2021 • 16 minutes to read • [Edit Online](#)

In Microsoft Dynamics 365 Commerce version 10.0.10 and later, inbound and outbound operations in the point of sale (POS) replace the picking and receiving operation.

## NOTE

In version 10.0.10 and later, any new features in the POS application that are related to receiving store inventory against purchase orders and transfer orders will be added to the inbound operations operation. If you're currently using the picking and receiving operation in POS, we recommend that you develop a strategy for moving from that operation to the new inbound and outbound operations. Although the picking and receiving operation won't be removed from the product, there will be no further investments in it, from a functional or performance perspective, after version 10.0.9.

## Prerequisite: Configure an asynchronous document framework

The outbound operation includes performance improvements to ensure that users who have high volumes of receipt postings across many stores or companies, and large inventory documents, can process those documents to Commerce headquarters (HQ) without experiencing time-outs or failures. These improvements require use of an asynchronous document framework.

When an asynchronous document framework is used, you can commit outbound document changes from POS to Commerce headquarters (HQ) and then move on to other tasks while the processing to Commerce headquarters (HQ) occurs in the background. You can check the status of the document through the **Outbound operation** document list page in POS to make sure that posting was successful. In the POS application, you can also use the outbound operation active document list to see any documents that could not be posted to Commerce headquarters (HQ). If a document fails, POS users can make corrections to it and then try again to process it to Commerce headquarters (HQ).

## IMPORTANT

The asynchronous document framework must be configured before a company tries to use the outbound operation in POS.

To configure an asynchronous document framework, complete the following procedures.

### Create and configure a number sequence

1. Go to **Organization administration > Number sequences > Number Sequences**.
2. On the **Number sequences** page, create a number sequence.
3. In the **Number sequence code** and **Name** fields, enter user-defined values.
4. On the **References** FastTab, select **Add**.
5. In the **Area** field, select **Commerce parameters**.
6. In the **Reference** field, select **Retail document operation identifier**.
7. On the **General** FastTab, in the **Setup** section, set the **Continuous** option to **No** to ensure that there are no performance issues.

### Create and schedule two batch jobs for the document processing and monitoring tasks

#### NOTE

In Commerce version 10.0.13 and later, you don't have to configure the batch jobs through the batch job framework. The batch processes can be configured from the **Retail and Commerce > Retail and Commerce IT** menu. Use the **Retail document operation monitor** and **Retail document operation processing** menu options to configure the batch jobs

The batch jobs that you create will be used to process documents that fail or time out. They will also be used when the number of active inventory documents that are being processed from POS exceeds a system-configured value.

1. Go to **System Administration > Inquiries > Batch jobs**.
2. On the **Batch job** page, create two batch jobs:
  - Configure one job to run the **RetailDocumentOperationMonitorBatch** class.
  - Configure the other job to run the **RetailDocumentOperationProcessingBatch** class.
3. Schedule the new batch jobs to run on a recurring basis. For example, set the schedule so that the jobs are run every five minutes.

## Prerequisite: Add Outbound operation to the POS screen layout

Before your organization can use the outbound operation functionality, it must configure the **Outbound operation** POS operation on one or more of your [POS screen layouts](#). Before you deploy the new operation in a production environment, make sure that you thoroughly test it and train your users to use it.

## Overview

The outbound operation lets POS users perform the following tasks:

- Post shipments for transfer order documents in cases where the user's store is the designated outbound warehouse.
- View information about historical transfer order shipments that were posted by the store.
- Create new outbound transfer order requests.

When the outbound operation is started from the POS application, a list page view appears. This view shows open transfer order documents that have inventory lines that the user's current store is intended to ship and fulfill. To find a select a document, users can scroll the list or use the search feature.

The outbound inventory document list has three tabs.

- **Active** – This tab shows transfer orders that have a status of **Requested** or **Partially Shipped**. The orders contain lines or quantities on lines that must be shipped by the user's current store. This tab also shows orders that have a status of **Processing in HQ** (that is, they are waiting for confirmation of successful posting from Commerce headquarters (HQ)) or **Processing failed** (that is, posting to Commerce headquarters (HQ) was unsuccessful, and the user must correct data and try again to submit the orders).
- **Draft** – This tab shows new outbound transfer order requests that the user's store created. However, the documents have only been saved locally. They haven't yet been submitted to Commerce headquarters (HQ) for processing.
- **Complete** – This tab shows a list of transfer order documents that the store has fully shipped during the last seven days. This tab is for informational purposes only. All the information about the documents is read-only data for the store.

When you view documents on any of the tabs, the **Status** field can help you understand the stage that the document is in.

- **Draft** – The transfer order document has only been saved locally to the store's channel database. No information about the transfer order request has yet been submitted to Commerce headquarters (HQ).
- **Requested** – The purchase order or transfer order has been created in Commerce headquarters (HQ) and is fully open. The user's current store has yet processed any shipments against the document.
- **Partially shipped** – The transfer order document has one or more lines or partial line quantities that have been posted as shipped by the outbound warehouse. These shipped lines are available to be received through the inbound operation.
- **Fully shipped** – The transfer order has had all its lines and full line quantities posted as shipped by the outbound warehouse.
- **In progress** – This status is used to inform device users that the document is being actively worked on by another user.
- **Paused** – This status is shown after **Pause receiving** is selected to temporarily stop the receiving process.
- **Processing in HQ** – The document was submitted to Commerce headquarters (HQ) from the POS application, but it hasn't yet been successfully posted to Commerce headquarters (HQ). The document is going through the asynchronous document posting process. After the document is successfully posted to Commerce headquarters (HQ), its status should be updated to **Fully received** or **Partially received**.
- **Processing failed** – The document was posted to Commerce headquarters (HQ) and rejected. The **Details** pane shows the reason for the posting failure. The document must be edited to fix data issues, and then it must be resubmitted to Commerce headquarters (HQ) for processing.

When you select a document line in the list, a **Details** pane appears. This pane shows additional information about the document, such as shipment and date information. A progress bar shows how many items must still be processed. If the document wasn't successfully processed to Commerce headquarters (HQ), the **Details** pane also shows error messages that are related to the failure.

In the document list page view, you can select **Order details** on the app bar to view the document details. You can also activate receipt processing on eligible document lines.

In the document list page view, you can also create a new outbound transfer order for a store.

## Transfer order shipping process

After you select a transfer order document on the **Active** tab, you can select **Order details** to begin the fulfillment process. The **Full order list** view appears. This page shows all the document lines that contain the item. It also shows details of the ordered quantity.

Each scan of a bar code updates the quantity in the **Shipping now** field by one unit. Alternatively, you can enter a shipping quantity by selecting **Ship product** on the app bar, entering an item ID, and then entering the quantity. If the item is location-controlled, you can confirm or set the shipping location for the document line.

In the **Full order list** view, you can manually select a line in the list and then update the **Shipping now** quantity for the selected line in the **Details** pane.

### Over-delivery shipping validations

Validations occur during the receiving process for the document lines. They include validations for over-delivery. If a user tries to receive more inventory than was ordered on a purchase order, but either over-delivery isn't configured or the quantity that is received exceeds the over-delivery tolerance that is configured for the purchase order line, the user receives an error and isn't allowed to receive the excess quantity.

### Underdelivery close lines

In Commerce version 10.0.12, functionality was added that lets POS users close or cancel remaining quantities during outbound order shipment if the outbound warehouse determines that it can't ship the full quantity that was requested. Quantities can also be closed or canceled later. To use this capability, the company must be

configured to allow underdelivery of transfer orders. Additionally, an underdelivery percentage must be defined for the transfer order line.

To configure the company to allow underdelivery of transfer orders, in Commerce headquarters (HQ), go to **Inventory management > Setup > Inventory and warehouse management parameters**. On the **Inventory and warehouse management parameters** page, on the **Transfer orders** tab, turn on the **Accept underdelivery** parameter. Then run the **1070** distribution scheduler job to sync the parameter changes to your store channel.

Underdelivery percentages for a transfer order line can be predefined on products as part of product configuration in Commerce Headquarters. Alternatively, they can be set or overwritten on a specific transfer order line through Commerce headquarters (HQ).

After an organization completes configuring transfer order underdelivery, POS users will see a new **Close remaining quantity** option in the **Details** pane when they select an outbound transfer order line through the **Outbound operation** function. When the user completes the shipment by using the **Finish fulfillment** operation, they can send a request to Commerce headquarters (HQ) to cancel the remaining unshipped quantity. If the user closes the remaining quantity, Commerce performs a validation to verify that the quantity that is being canceled is within the underdelivery percentage tolerance defined on the transfer order line. If the underdelivery tolerance is exceeded, an error message is displayed and the user won't be able to close the remaining quantity until the previously shipped and "ship now" quantity meets or exceeds the underdelivery tolerance.

After the shipment is synced to Commerce headquarters (HQ), the quantities that are defined in the **Ship now** field for the transfer order line in POS are updated to a shipped status in Commerce headquarters (HQ). Any unshipped quantities that previously would have been considered "ship remain" quantities (that is, quantities that will be shipped later) are considered canceled quantities instead. The "ship remain" quantity for the transfer order line is set to **0** (zero), and the line is considered fully shipped.

### **Shipping location-controlled items**

If the items that are being shipped are location-controlled, users can choose the location that they want to issue the inventory from during the shipping process. We recommend that you configure a default issue location for your store warehouse, to make this process more efficient. Even if a default location is configured, users can override the issue location on selected lines as they require.

The operation respects the **Blank receipt allowed** configuration on the **Location** storage dimension and doesn't require that a location dimension be entered if blank receipt is configured. If blank receipt locations aren't allowed for an item, the POS application shows an error and requires that a location be entered before the receipt can be posted.

### **Ship all**

As you require, you can select **Ship all** on the app bar to quickly update the **Shipping now** quantity for all the document lines to the maximum value that is available to be fulfilled for those lines.

### **Cancel fulfillment**

Use the **Cancel fulfillment** function on the app bar only if you want to back out of the document and don't want to save any changes. For example, you initially selected the wrong document and don't want any of the previous shipping data saved.

### **Pause fulfillment**

If you're fulfilling the transfer order, you can use the **Pause fulfillment** function if you want to take a break from the process. For example, you might want to perform another operation from the POS, such as ringing up a customer sale, or delay posting of the shipment to Commerce headquarters (HQ).

When you select **Pause fulfillment**, the document's status is changed to **Paused**. Therefore, user will know that data has been entered in the document, but the document hasn't yet been committed. When you're ready to

resume the fulfillment process, select the paused document, and then select **Order details**. Any **Shipping now** quantities that were previously saved will be retained and can be viewed from the **Full order list** view.

## Review

Before the final commitment of the fulfillment to Commerce headquarters (HQ), you can use the **Review** function to validate the outbound document. This function alerts you to potential missing or incorrect data that may cause a processing failure, and provide you the opportunity to correct issues before submitting the fulfillment request. To enable the **Review** function on the app bar, enable the **Enable validation in POS inbound and outbound inventory operations** feature through Feature management in Commerce headquarters (HQ).

The **Review** function validates the following issues in an outbound document:

- **Over-shipping** – the shipping now quantity is greater than the ordered quantity. The severity of this issue is determined by the overdelivery configuration in Commerce headquarters (HQ).
- **Under-shipping** – the shipping now quantity is less than the ordered quantity. The severity of this issue is determined by the underdelivery configuration in Commerce headquarters (HQ).
- **Serial number** – serial number is not provided or not available for a serialized item that requires a serial number to be registered in inventory.
- **Location not set** – location is not specified for a location-controlled item where location is not allowed to be blank.
- **Deleted lines** – the order has lines deleted by a Commerce headquarters (HQ) user that is not known to the POS application.

If you set the **Enable automatic validation** parameter to **Yes** in **Commerce parameters > Inventory > Store inventory operations**, validation is executed automatically when you select the **Finish fulfillment** function.

## Finish fulfillment

When you've finished entering all the **Shipping now** quantities for products, you must select **Finish fulfillment** on the app bar.

When asynchronous document processing is used, the receipt is submitted through an asynchronous document framework. The time that it takes for the document to be posted depends on the size of the document (the number of lines) and the general processing traffic that is occurring on the server. Typically, this process occurs in a matter of seconds. If document posting fails, the user is notified through the **Outbound operation** document list on the **Active** tab, where the document status will be updated to **Processing failed**. The user can then select the failed document in POS to view the error messages and the reason for the failure in the **Details** pane. A failed document remains unposted and requires that the user return to the document lines by selecting **Order details** in POS. The user must then update the document with corrections, based on the errors. After a document is corrected, the user can try again to process it by selecting **Finish fulfillment** on the app bar.

## Create an outbound transfer order

From POS, users can create new transfer order documents. To begin the process, select **New** on the app bar while you're in the main **Outbound operation** document list. You're then prompted to select a **Transfer to** warehouse or store that your current store will send inventory to. The values are limited to the selection that is defined in the configuration of the store's fulfillment group. In an outbound transfer request, your current store will always be the **Transfer from** warehouse for the transfer order. That value can't be changed.

You can enter values in the **Ship date**, **Receive date**, and **Mode of delivery** fields as you require. You can also add a note that will be stored together with the transfer order header, as an attachment to the document in Commerce headquarters (HQ).

After the header information is created, you can add products to the transfer order. To start the process of

adding items and requested quantities, scan bar codes or select **Add product**.

After lines are entered on the outbound transfer order, you must select **Save** to save the document changes locally or **Submit request** to submit the order details to Commerce headquarters (HQ) for further processing. If you select **Save**, the draft document is stored in the channel database, and the outbound warehouse can't run the document until it has been successfully processed via **Submit request**. Select **Save** only if you aren't ready to commit the request to Commerce headquarters (HQ) for processing.

If a document is saved locally, it can be found on the **Drafts** tab of the **Inbound operation** document list. While a document is in **Draft** status, you can edit it by selecting **Edit**. You can update, add, or delete lines as you require. You can also delete the whole document while it's in **Draft** status, by selecting **Delete** on the **Drafts** tab.

After the draft document is successfully submitted to Commerce headquarters (HQ), it appears on the **Active** tab and has a status of **Requested**. At this point, only users in the outbound warehouse can edit the document, by selecting **Outbound operation** in the POS application. Users in the inbound warehouse can view the transfer order on the **Active** tab of the **Inbound operation** document list, but they can't edit or delete it. The editing lock ensures that no conflicts occur because an inbound requestor changes the transfer order at the same time that the outbound shipper is actively picking and shipping the order. If changes are required from the inbound store or warehouse after the transfer order has been submitted, the outbound shipper should be contacted and asked to enter the changes.

After the document is in **Requested** status, it's ready for fulfillment processing by the outbound warehouse. As the shipment is processed by using the outbound operation, the status of the transfer order documents is updated from **Requested** to **Fully shipped** or **Partially shipped**. After the documents are in **Fully shipped** or **Partially shipped** status, the inbound store or warehouse can post receipts against them by using the inbound operation receiving process.

Fully shipped transfer orders are moved to the **Complete** tab of the **Outbound operation** document list. There, they remain visible to users in the outbound store or warehouse, in read-only mode, for seven days.

## Related topics

[Inbound inventory operation in POS](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Calculate inventory availability for retail channels

2/18/2021 • 13 minutes to read • [Edit Online](#)

This topic describes how a company can use Microsoft Dynamics 365 Commerce to view estimated on-hand availability for products in the online and store channels.

## Accuracy of calculation

Commerce uses multiple servers and databases to ensure scalability and performance. Therefore, it's important that you understand that the available inventory values that are provided through the point of sale (POS) application, the e-Commerce inventory availability application programming interfaces (APIs), and the on-hand inventory pages in Commerce Headquarters might not be 100-percent accurate in real time. If transactions that are created for products in the online or store channel haven't yet been synced to the Commerce Headquarters server and database, the on-hand inventory pages in Commerce Headquarters might not show an accurate real-time inventory value for those products. Conversely, if you configured your company so that users in Commerce Headquarters or other integrated applications can sell, receive, return, or otherwise adjust inventory out of a store or online warehouse, the POS or online channel might not have all the information that is required to show an accurate real-time on-hand value for items.

This topic explains the data synchronization processes that can be run frequently to help limit the latency of data between applications or channels. However, it's critical that you understand that all on-hand availability data that is provided during the operational day is considered an estimated value. Therefore, if you try to compare the on-hand inventory information that the application provides with actual physical inventory on the shelves, or if you try to compare the on-hand values that are shown in POS with the on-hand data that you find for the same warehouse in Commerce Headquarters, the values might differ. This difference during the operational day is expected and should not be considered an issue. If you want to audit data and make sure that the values that are provided in the inventory availability APIs and Commerce Headquarters match the actual physical units that you find on your store or warehouse shelves, the best time to do it is after channel operations have stopped for the day and all transactions have been correctly synced between Commerce Headquarters and the channel.

## Use inventory lookup APIs for e-Commerce inventory availability requests

You can use the following APIs to show inventory availability for a product when your customers are shopping on an e-Commerce site.

- **GetEstimatedAvailability** – Use this API to get inventory availability for the item in the e-Commerce channel warehouse or all warehouses that are linked to the configuration of the fulfillment group for the e-Commerce channel. This API can also be used for warehouses in a specific search area or radius, based on longitude and latitude data.
- **GetEstimatedProductWarehouseAvailability** – Use this API to request inventory for an item from a specific warehouse. For example, you can use it to show inventory availability in scenarios that involve order pickup.

### NOTE

These APIs replace the **GetProductAvailabilities** and **GetAvailableInventoryNearby** APIs in Dynamics 365 Retail version 10.0.7 and earlier.



Both APIs fetch data from the Commerce server and provide an estimate of on-hand inventory for a specific combination of a product or product variant and a warehouse. Although other APIs that are available on the Commerce server can go directly to Commerce Headquarters to fetch on-hand quantities for products, we don't recommend that they be used in an e-Commerce environment because of potential performance issues and the related impact that these frequent requests can have on your Commerce Headquarters servers. Additionally, if the on-hand inventory is calculated through the Commerce server, the calculation is more likely to include inventory that was sold in recent e-Commerce transactions that haven't yet been synced to Commerce Headquarters. Although Commerce Headquarters might not have information about these transactions, the Commerce server and channel database have the data. Therefore, the data will be factored in and can help provide a more accurate estimate of a product's available inventory.

### **Get started with e-Commerce calculated inventory availability**

Before you use the two APIs that were mentioned earlier, you must enable the **Optimized product availability calculation** feature through the **Feature management** workspace in Commerce Headquarters.

Before the APIs can calculate the best estimate of inventory availability for an item, a periodic snapshot of inventory availability from Commerce Headquarters must be processed and sent to the channel database that the e-Commerce Commerce Scale Unit uses. The snapshot represents the information that Commerce Headquarters has about inventory availability for a specific combination of a product or product variant and a warehouse. It can include inventory adjustments or movements that are caused by inventory receipts, or by shipments or other processes that are performed in Commerce Headquarters and that the e-Commerce channel has information about only because of the synchronization process.

The database snapshot that the **Product availability** job creates calculates only the inventory transactions that were processed and posted in Commerce Headquarters at the time when the snapshot was taken. If inventory was sold for a product in a store warehouse through a cash-and-carry or asynchronous customer order sale in the POS application, Commerce Headquarters won't immediately have information about the related inventory issue transaction for the sale. It will have information about the inventory that is sold for these types of store sales only after the P-job uploads the related transaction from the store's channel database into Commerce Headquarters and the related sales order is created through statement posting or the trickle feed posting processes. The process of creating the order in Commerce Headquarters creates the related inventory transactions. For e-Commerce channel orders, Commerce Headquarters has information about the inventory transactions only after the transactions are sent to Commerce Headquarters through the P-job and the order synchronization process is completed. Therefore, it's important that you understand that the inventory snapshot value that is provided in the **Product availability** job might not be 100-percent accurate in real time because of the constant sales processing that occurs across distributed servers.

To take a snapshot of inventory in Commerce Headquarters, follow these steps.

1. Go to **Retail and Commerce > Retail and Commerce IT > Products and inventory > Product availability**.
2. Select **OK** to run the **Product availability** job. You can also schedule this job so that it's run in a batch.

After the **Product availability** job has finished running, the data that was captured must be pushed to the e-Commerce channel databases, so that the latest Commerce Headquarters inventory snapshot can be considered in the calculation of estimated on-hand inventory.

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Run the **1130 (Product availability)** job to sync the snapshot data that the **Product availability** job created from Commerce Headquarters to your channel databases.

When inventory availability is requested from the **GetEstimatedAvailability** or **GetEstimatedProductWarehouseAvailability** API, a calculation is run to try to get the best possible estimate of inventory for the product. The calculation references any e-Commerce customer orders that are in the channel database but that weren't included in the snapshot data that the 1130 job provided. This logic is

performed by tracking the last processed inventory transaction from Commerce Headquarters and comparing it with the transactions in the channel database. It provides a baseline for the channel-side calculation logic, so that the additional inventory movements that occurred for customer order sales transactions in the e-Commerce channel database can be factored into the estimated inventory value that the API provides.

The channel-side calculation logic returns an estimated physically available value and a total available value for the requested product and warehouse. The values can be shown on your e-Commerce site if you want, or they can be used to trigger other business logic on your e-Commerce site. For example, you can show an "out of stock" message instead of the actual on-hand quantity that the API passed.

The calculation logic that the channel-side e-Commerce APIs use for the estimated inventory value can evaluate inventory based only on customer orders that have been created in the channel database but that haven't yet been synced and posted in Commerce Headquarters. If your channel database also contains transactional data for cash-and-carry sales for store-specific warehouses, the cash-and-carry sales aren't factored into the channel-side e-Commerce calculation for those warehouses.

## Configure the inventory lookup operation in the POS channel

In Retail version 10.0.9 and earlier, the **Inventory lookup** operation from POS used a real-time service call to Commerce Headquarters to get inventory information for the selected product, for both the user's current store and any other stores that are configured for the fulfillment group as part of the channel configuration for the store. In Commerce version 10.0.10 and later, you can turn off real-time service calls to Commerce Headquarters. Instead, you can use channel-side calculation on the Commerce server to determine the on-hand inventory that is physically available for the store and any other locations that are defined in the fulfillment group. This channel-calculated inventory configuration is also useful for locations where internet connectivity is unreliable, because you don't have to be online to get inventory lookups from Commerce Headquarters.

When channel-side calculation is correctly configured and managed, it can provide a more reliable estimate of the current store inventory, because it uses the transactional data that is in the Commerce channel database but that Commerce Headquarters might not yet have information about. For example, if you use the existing real-time service call for inventory lookups in POS, Commerce Headquarters probably won't yet have information about a cash-and-carry sale that just occurred for a product. Therefore, the on-hand inventory value that Commerce Headquarters returns for that product will probably exceed the store's actual on-hand inventory by one unit. However, if you use channel-side calculation, the cash-and-carry sale can be factored into the calculation and deducted from the on-hand value that is shown. Although the values that both the channel-side calculation and the real-time service call provide are only estimates of on-hand inventory, the value that the channel-side calculation provides is much more likely to be accurate for the current store.

### Get started with POS channel-side calculated inventory availability

To use the channel-side calculation logic and turn off real-time service calls for inventory lookups from the POS application, you must first enable the **Optimized product availability calculation** feature through the **Feature management** workspace in Commerce Headquarters. In addition to enabling the feature, you must make changes to the **Functionality profile**.

To change the **Functionality profile**, follow these steps:

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**.
2. Select a functionality profile.
3. On the **Functions** FastTab, in the **Invent availability calculation** section, change the value of the **Invent availability calculation mode** field from **Real time service** to **Channel**. By default, all functionality profiles use real-time service calls. Therefore, you must change the value of this field if you want to use channel-side calculation logic. Every retail store that is linked to the selected functionality profile will be affected by this change.

You must then sync the changes to the channel through the distribution schedule process by performing the following steps:

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Run the **1070 (Channel configuration)** job.

After the configuration is completed, the information that is provided about physically available inventory no longer uses a real-time service call when a user in the POS application uses the **Inventory lookup** operation (standard and matrix views). Instead, data about physically available inventory for the current store and all the stores in the fulfillment group is calculated based on the last-known snapshot that was delivered to the channel database from Commerce Headquarters. The snapshot value is further refined by the channel-side calculation to adjust the physically available value, based on additional sales or return transactions that exist for the selected product in the channel database that were not included in the last synchronized snapshot from the 1130 job. If the channel database doesn't contain transactional data for any of the warehouses or stores in the fulfillment group, it contains no additional transactions that can be factored into a recalculation of the value. Therefore, the best estimate of on-hand inventory that can be shown for those warehouses or stores is the data from the last-known Commerce Headquarters snapshot.

The **Order fulfillment** screens of POS also leverage the channel side calculation to show on-hand inventory for items when an order fulfillment line is selected and a user views the **Details** panel for on-hand inventory for the selected item.

## Optimize your inventory data

To ensure the best possible estimate of inventory, it's critical that you use the following Commerce batch jobs and run them frequently:

- **P-job** – The P-job is found on the **Distribution schedules** page and should be run frequently. This job brings e-Commerce orders, asynchronous customer orders that POS created, and cash-and-carry orders that POS created from the channel databases into Commerce Headquarters, so that they can be processed further. Until this data is synced from the channel to Commerce Headquarters, Commerce Headquarters has no information about inventory adjustments to products in the warehouses that result from those transactions.
- **Synchronize orders** – This job processes the raw transactional data in Commerce Headquarters that the P-job provides and converts e-Commerce and asynchronous customer order transactions into sales orders in Commerce Headquarters. Until this job is processed and the sales orders are created, no inventory transactions are created. Therefore, on-hand inventory in Commerce Headquarters won't consider the transactions.
- **Calculate transactional statements in batch** – For cash-and-carry transactions that are created in the store, the trickle feed posting process ensures that inventory that is related to the sales is updated efficiently. To get the most efficient processing of inventory transactions for cash-and-carry orders, make sure that you configure your system to use [trickle feed posting](#).
- **Post transactional statements in batch** – This job is also required for trickle feed posting. It follows the **Calculate transactional statements in batch** job. This job systematically posts the calculated statements, so that sales orders for cash-and-carry sales are created in Commerce Headquarters and Commerce Headquarters more accurately reflects your store's inventory.
- **Product availability** – This job creates the snapshot of inventory from Commerce Headquarters.
- **1130 (Product availability)** – This job is found on the **Distribution schedules** page and should be run immediately after the **Product availability** job. This job transports the inventory snapshot data from Commerce Headquarters to the channel databases.

It's recommended that you don't run those batch jobs too frequently (every few minutes). Frequent runs will overload Commerce headquarters (HQ) and can potentially impact performance. In general, it's good practice to

run product availability and 1130 jobs on an hourly basis, and schedule P-job, synchronize orders, and trickle feed posting-related jobs with the same or higher frequency.

**NOTE**

For performance reasons, when channel-side inventory availability calculations are used to make an inventory availability request using the e-Commerce APIs or the new POS channel-side inventory logic, the calculation uses a cache to determine whether enough time has passed to justify running the calculation logic again. The default cache is set to 60 seconds. For example, you turned on channel-side calculation for your store and viewed the on-hand inventory for a product on the **Inventory lookup** page. If one unit of the product is then sold, the **Inventory lookup** page won't show the reduced inventory until the cache has been cleared. After users post transactions in POS, they should wait 60 seconds before they verify that the on-hand inventory has been reduced.

If your business scenario requires a smaller cache time, contact your product support representative for help.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Purchase order overview

2/18/2021 • 5 minutes to read • [Edit Online](#)

This article provides general information about purchase orders (POs) and links to additional articles that are related to the various stages that a PO goes through.

A purchase order (PO) is a document that represents an agreement with a vendor to buy goods or services. The document also helps keep track of product receipts that are made toward the order and, later, the accounting of vendor invoices that the vendor bills toward the order.

The **Purchase orders** page contains an overview of the available orders and lets you modify those orders. When you open a PO, you can select the **Header** view, which contains information that is specified only one time for each PO, such as the vendor details. Alternatively, you can select the **Lines** view, where you can modify order lines. Typically, you will switch between these two views as you modify POs. Charges aren't listed directly on the **Purchase orders** page, but are accessed via menus on the order header and lines.

There are many reports where you can view information about POs, product receipts, and vendor invoices. These reports are found in the **Procurement and sourcing** and **Accounts payable** modules.

The **Purchase order preparation** and **Purchase order receipt and follow-up** workspaces let you view lists of POs in the various states that they have progressed to. They also provide a summary of the actions that must be taken. The **Purchase order preparation** workspace is focused on PO creation and review, processing of the order through approval, and confirmation with the vendor. The **Purchase order receipt and follow-up** workspace is focused on processing the receipt of goods or services against POs. It includes lists that give insight into receipts that are overdue, or that will soon be due for delivery by the supplier. These workspaces aren't used to perform the related receipt activities that are done in the warehouse. Those activities are performed by using pages in the **Inventory management** and **Warehouse management** modules. Processing of vendor invoices should be done by using the **Vendor invoice entry** workspace, and payments should be done by using the **Vendor payments** workspace.

The following articles provide an overview of the various stages that a PO goes through:

- [Create purchase orders](#)
- [Approve and confirm purchase orders](#)
- [Product receipt against purchase orders](#)
- [Overview of vendor invoices](#)

## Types of purchase orders

There are three types of POs. When you create a PO, you must specify the type. You can set up a default order type for new orders on the **Procurement and sourcing parameters** page.

PO TYPE	DESCRIPTION
Journal	Use this type to create a draft order. This type doesn't affect stock quantities or generate inventory transactions. The PO journal lines aren't included in master scheduling.
Purchase order	Use this type to create POs when orders are confirmed with a vendor, and as the orders are processed through receipt and invoicing before payment is made to the vendor. This type of PO is the most common.

PO TYPE	DESCRIPTION
Returned order	Use this type when you return goods to the vendor. This type of order requires that you specify the return material authorization (RMA) number that the vendor gives you. You specify the RMA number on the <b>General</b> tab of the PO. The order lines must have negative quantities.

## Purchase order statuses

POs include several status fields that indicate the progress of the order. All these fields are visible in the **Header** view of the order, and a few of them are also visible in the grid overview of all orders. The **Status** field shows the status for quantities on the order. The following values are available:

- **Open order** – Orders have been created, and quantities are on order.
- **Received** – Some of the quantities have been received, but they haven't been invoiced yet.
- **Invoiced** – The full quantity on the order has been invoiced. **Note:** If an order has been *partially* invoiced, neither **Received** status nor **Invoiced** status is appropriate. Therefore, the order will still have a status of **Open order**.
- **Canceled** – An order was confirmed but later canceled. Therefore, this status indicates that there are no longer any open quantities on order.

The **Document status** field helps you quickly review the order's progress in terms of documents that have been processed. It shows the status of the most recent document that has been completed for the order. The following values are available:

- **None** – No document has been processed for the order yet.
- **Purchase inquiry** – A purchase inquiry has been generated, and the order is awaiting feedback from the vendor.
- **Purchase order** – Confirmation has been processed on the order.
- **Product receipt** – Product receipt has been processed on the order.
- **Invoice** – An invoice has been accounted with the order.

The **Approval status** field is used when a PO goes through a review process or workflow. The following values are available:

- **Draft, In review, and Rejected** – These statuses are used only when an approval workflow is used for the PO.
- **Approved** – This status is assigned to orders that have completed workflow approval. Orders that are created without using an approval workflow receive a status of **Approved** immediately.
- **In external review** – This status is used in scenarios where a purchase inquiry is sent to the vendor, so that the vendor can confirm terms of the PO. This status is also used in the process that is initiated by the **Confirmation request** action. For this process, the vendor is asked to confirm terms of the PO by connecting to your system and registering whether it confirms or rejects the order.
- **Confirmed** – This status is assigned after the order has been confirmed. Typically, this status is the last approval status that is assigned to an order.

## Additional resources

[Create purchase orders](#)

[Approve and confirm purchase orders](#)

[Product receipt against purchase orders](#)

## Overview of vendor invoices

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create purchase orders

2/18/2021 • 9 minutes to read • [Edit Online](#)

This article describes the process and options when you manually create a purchase order.

When you create a purchase order (PO), general information about the whole order is specified in the PO header, and you then add one or more PO lines. This article describes some of the most frequently used options that are available.

You can also create POs by copying lines from another PO document or a sales order. In this case, you can reverse the sign on the inventory, as you would reverse the sign on an invoice to indicate credit.

Although you can manually create POs, they are more typically generated from other processes. Orders can be automatically created based on other documents, such as requisitions. Alternatively, they can be created as part of the master planning process through planned POs. If you use purchase agreements, POs can be created by the **Release order** action. There are also more advanced methods for automatically creating a PO. For example, orders can be created when you use direct delivery or intercompany order chains.

## Creating a purchase order header

When you create a new PO, a dialog box appears, where you can enter the most common information for the PO header. When you click **OK** to close the dialog box, the order is created, and you can then specify additional information in the header.

The first detail that you must consider when you create a PO is the type of order. The **Purchase order** type is used most often. However, if a credit invoice is required, you can use the **Returned order** type.

You must specify the supplier in the **Vendor account** field. For this field, you can search on either the account or the vendor name. If a vendor delivers from multiple locations but uses a single invoice account, you can select that invoice account in the **Invoice account** field and then use it with different vendor accounts. If you must create a PO for products that won't be ordered repeatedly, you can use the **One-time supplier** option. This option automatically creates a new vendor account that is marked as a one-time account, to support a later clean-up process for one-time accounts in the **Accounts payable** module. When you select a vendor account, many fields in the PO header inherit default values from the information that is associated with the vendor account. For example, the default delivery site and warehouse are copied from the vendor information. However, you can override these default values if the purchase is intended for another location.

If the supplier has provided a reference number for the order, you can record this information in the **Vendor reference** field. For returned orders, you must specify a value in the **RMA** field to reference the supplier's authorization for processing the return.

If a purchase agreement is associated with the order, you must specify this information in the **Purchase agreement** field.

The PO header also contains information about charges that apply to the whole order instead of individual lines. Charges can be automatically added to the order if automatic charges have been set up for the vendor or the vendor's charge group. You can also manually add charges to the order header by clicking **Maintain charges** on the Action Pane.

## Adding purchase order lines

POs can be for physical products or for services. A setting on the inventory model group determines whether a



particular item number applies to a product or a service. Usually, the item that is purchased is specified by an item number. However, if the order is for products or services that are directly consumed, you can also specify the item by using a procurement category.

PO lines contain lots of fields, but many of these fields have a default value or a value that is inherited from the order header. Additional fields are set when you select a product or service. The fields that are most often set manually include the fields for the item number, quantity, and requested delivery date. Information about unit price and discounts is also very important, but the values of those fields are often determined by trade agreements or purchase agreements.

When you select a product, you can search on all or part of the product name instead of using the item number. If the product has several variants, such as different sizes, you can see an overview of the available variants by using the **Add lines** function or by using the lookup that is available in the **Variant number** field.

Often, you will have to specify several dimensions for the item that is selected on each PO line. The dimensions that must be specified depend on the dimension groups that have been assigned to the product master definition. For example, you will often have to specify a site and warehouse to indicate the location that the product should be delivered to. You identify product variants by specifying a variant number, or by entering values for one or more product dimensions, such as color, size, configuration, or style. Tracking dimensions, such as batch and serial number, let you uniquely identify each inventory lot. After you've created an order, you can capture dimension values on the order by using the **Registration** action. For example, you have ordered a quantity of five pieces of an item. Later, you register that three of those pieces will be black, and two of them will be blue. This approach is an alternative to capturing the dimension information during arrival registration.

You can check the details of the inventory transaction status for stocked products. For example, you might want to check the on-hand inventory to help you decide how much to order. Alternatively, you might want to review the inventory status of an ordered quantity to see whether inbound arrival registration has occurred.

A PO line that is being used to return a product to the vendor will have a negative quantity. You can select a specific lot to return by using the **Reservation** action.

Sometimes, you might want to divide the quantity that you've ordered, so that different parts of it are delivered on different dates. You can set up these deliveries by using the **Delivery schedule** action, which is available on the **Purchase order line** menu in the **Lines** view.

Charges can be automatically added to PO lines, if automatic charges have been set up for the vendor or vendor charge group, and for the item or item charge group. However, more typically, charges are added manually at the order line level. To add a charge, open the **Maintain charges** page by using the **Maintain charges** action on the **Financials** menu in the **Lines** view. The advantage of adding charges directly at the order line level is that the charge can be allocated as an inventory cost. To set up charge codes to account product cost, use the **Item** debit option. These types of charges must be allocated from the PO header to the lines before the order can be confirmed. For example, you might want to allocate charges based on the quantity on each line. The charge category also affects how charges are accounted. For example, fixed charges specify a fixed amount, and percent charges are calculated as a percentage of the net amount for the order line. POs can be assigned to a load, and the load might include an estimate of the expected expense for the transportation cost. You can allocate this expense from the load back to the PO lines.

## Purchase order actions

After you've added the header and lines to the PO, you must often complete additional steps before the order is ready to be confirmed. Because so many options are available, you might find it helpful to use [Action search](#) to find the relevant menu item.

You can configure products on the order so that they have supplementary items. Supplementary items are products that must or can be bought together with other products. Supplementary products might be added free of charge as accompanying products, or you may be able to decide whether to add them to the order or

not. You can review the supplementary items after each order line that is added. However, you will probably find it more convenient to review and add relevant supplementary items for all the order lines by using the **Supplementary items** page, which you can open from the Action Pane.

Discounts are usually added to lines as they are created. However, a few discounts apply to the whole order:

- The **Total discount** action calculates a total discount percentage that is applied to the full order. Don't confuse this discount with the cash discount percentage. Cash discounts are applied when the invoice is paid, and they depend on payment settlement by a specific date.
- If a multi-line discount applies, you must use the **Multiline discount** action to calculate and assign it to the order. Multi-line discounts are discounts that can be offered if a mix of products on the order exceeds a joint threshold. Only a few companies use this type of discount.

Charges that have a charge code that uses the **Item** debit type must be assigned to the line level before the order can be confirmed. You might find it convenient to assign these charges at the order header level, so that you can specify the total amount of the charge. However, in this case, the charge must then be allocated down to each line before the order can be confirmed. You can use the **Allocate charges** action to split amounts from charges that are assigned at the header level down to the order lines. Charges can be split according to the net amount of each line, according to the quantity that has been ordered, or evenly across the order lines. After you've allocated charges to the lines, the charge is removed from the order header.

POs can be configured to require that budget funds be allocated to the order before it can be processed. In this case, you can use the **Budget checking** action to allocate the budget.

You might have to delay the completion of a PO. For example, you might require additional information about products or services, or you might have to get authorization for the spend. There are several ways to hold back an order. For example, you can wait to confirm the order. Alternatively, if a change management workflow is being used, don't submit the order for approval. If you must block all orders for a particular vendor, you can also mark the vendor as **On hold** for processing on the vendor master. There are also circumstances that might prevent the order from being processed. For example, processing might be prevented if credit limits have been exceeded, or if required budget funds aren't available.

## Additional resources

[Purchase order overview](#)

[Approve and confirm purchase orders](#)

[Product receipt against purchase orders](#)

[Vendor invoices overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product receipt against purchase orders

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes the various options for registering products as received.

Product receipt is the process of recording that products that were ordered have been received, so that the purchase order (PO) lines can then be processed for invoicing. In some cases, products go through preregistration, where additional information from the supplier is recorded before the products are received. When products arrive, they are first marked as **Registered**. The products might then go through additional processes, such as quality management, before they are finally marked as **Received**.

## Preregistration (ASN)

Suppliers might share information about products that will be shipped. In this case, you can preregister the products to record this information before the products are received. By preregistering products, you reduce the amount of work that is required during item registration and receipt. Suppliers can provide product information electronically through an Advance Shipment Notice (ASN) that is then automatically recorded in the system. The information in the ASN includes the quantity of products that will be shipped and the date when they will be shipped. The ASN might also include information such as batch or serial numbers. Registration of the ASN occurs in the **Transportation management** module.

## Registration

Product receipt registration often occurs at the inbound docks in a warehouse. It's performed either by using a hand-held device or through arrival journals. Alternatively, you can manually register product receipt by using the **Registration** action on the **Purchase order** page. In both cases, the products are marked as **Registered**. Note that the products aren't yet marked as **Received**.

Products that are received in a warehouse might go through quality inspection before they are put away into inventory. Either quality orders or quarantine orders can be used to perform quality inspection. If quality orders are used, you can configure the process to temporarily block products through a reservation while they are inspected. If quarantine orders are used, products are moved to another warehouse for inspection. This warehouse is known as the quarantine warehouse. In both quality inspection processes, some of the goods might be scrapped, either because they don't conform to the quality expectations or because the quality inspection involves destructive testing of a sample of the product.

## Product receipt

Most often, the **Product receipt** action on the **Purchase orders** page is used to mark products as **Received** on the PO. The **Posting product receipt** page has various options for the quantity that is accounted as received. For example, you can set the **Quantity** field to **Ordered quantity** or **Receive now quantity**. Alternatively, if a warehouse arrival process has been used, you will often set this field to **Registered quantity**. You can modify the quantities on each order line that will be marked as **Received**, to account for any discrepancies, such as under-delivery and over-delivery. During product receipt, you must specify a product receipt identifier, which is typically a reference to the packing slip from the supplier. This identifier is required for accounting, because it enables checks or audits of supplier packing slips against what has been received, and the accounted inventory or expense.

POs can be created for products that aren't intended as inventory but are considered an expense. This category includes order lines where the products are marked as **Not stocked** by their inventory model group, and also

lines that use procurement categories. In this case, the items might not go through arrival registration and receipt in the warehouse. Instead, the **Product receipt** action is used to record the receipt directly on the PO, and the receipt is based on the ordered quantity, not a registered quantity.

You can create PO lines where the **New fixed asset** option is enabled. This option indicates that the purchase should be considered a fixed asset instead of inventory. In this case, the fixed asset determination rules that have been configured determine whether the purchase of the product or category exceeds specific thresholds, and must therefore be accounted for as an asset and go through fixed asset management. Purchases can also be made toward an existing fixed asset. In this case, the amount is adjusted as appropriate.

You can select multiple orders and process receipt on all those orders together. This approach isn't used very often, but you might want to use it if a supplier has consolidated shipments for you into a single load. During product receipt on the purchase, there is a function for doing summary updates. Summary updates let you post a single packing slip from the supplier for more than one PO.

POs might be created from a sales order where the **Direct delivery** option was selected. When direct delivery is used, the products never arrive in your warehouse but are shipped directly from the supplier to the customer. In this case, the receipt is usually recorded directly on the PO. The receipt can be done automatically, such as through electronic data interchange (EDI) integration with the supplier. Alternatively, if the PO is an intercompany PO, Supply Chain Management automates the receipt on the intercompany sales order when shipment occurs. When direct delivery is used, products are still accounted as inventory, even though they don't physically arrive at the warehouse. Therefore, when product receipt is registered on the PO, the sales order is automatically updated with a packing slip, so that the overall change to inventory is 0 (zero). In direct delivery scenarios, you should not require preregistration. If you're using warehouses that are enabled for warehouse management, you can get around the requirement for license plate registration by specifying a virtual warehouse instead. You specify this warehouse in the **Direct delivery warehouse** field on the product.

After the product receipt has been processed on the PO, the PO status is set to **Received** to indicate that the invoice can be processed for the order. You can review details about products that have already been received by using the **Product receipt journals** page.

You can access this page from the **Receipt** action group on the **Purchase order** page. The information in the journals includes details about the quantities, dates, and dimensions.

## Additional resources

[Purchase order overview](#)

[Create purchase orders](#)

[Approve and confirm purchase orders](#)

[Overview of vendor invoices](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Approve and confirm purchase orders

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes the statuses that a purchase order (PO) goes through after it has been created, and the effect of enabling change management on POs.

After a purchase order (PO) has been created, it might have to go through an approval process. After the vendor has agreed to the order, the PO is set to a status of **Confirmed**.

## Approval of purchase orders

POs that don't use change management have a status of **Approved** as soon as they are created, whereas POs that use change management have a status of **Draft** when they are first created. A PO that has been created by firming a planned order from master planning is always set to a status of **Approved**, regardless of the change management settings. A PO creates inventory transactions only when it reaches the **Approved** status. Therefore, that inventory doesn't appear as available for reservation or marking until the order is accepted.

You enable change management for POs by setting the **Activate change management** option on the **Procurement and sourcing parameters** page. When change management is enabled, POs must go through an approval workflow after they have been completed. Supply Chain Management has a workflow process editor where you can define a workflow to represent your approval process. This workflow can include rules for automatic approval, rules that determine who will be assigned to approve particular POs, and rules for escalating a workflow that has been waiting for approval for a long time. You can enable the change management process for all vendors or for specific vendors. You can also set up the process so that it can be overridden for individual POs.

When change management is enabled, POs move through six approval statuses, from **Draft** to **Finalized**. After an order has been approved, users who want to modify it must use the **Request change** action.

APPROVAL STATUS	DESCRIPTION	REQUEST CHANGE IS ENABLED
Draft	The PO is a draft and hasn't been submitted for approval in the PO workflow.	No
In review	The PO was submitted for approval in the PO workflow. Approval is pending.	No
Rejected	The PO was rejected during the approval process.	No
Approved	The PO was approved.	Yes
Confirmed	The PO was confirmed. A PO can't be confirmed until it has been approved.	Yes
Finalized	The PO was made final. It's now financially closed and can no longer be changed.	No

## Confirming purchase orders

POs that have an approval status of **Approved** can go through additional steps before they are confirmed. For example, you might have to send a purchase inquiry to the vendor to inquire about prices, discounts, or delivery dates. In this case, you can set the PO to the **In external review** status by using the **Purchase inquiry** action.

Vendors that are set up to use the Vendor portal can review orders on the portal, and approve or reject them. During this review process, the PO has a status of **In external review**. The Vendor portal can be configured so that a confirmation from the vendor automatically confirms the order in Supply Chain Management. Alternatively, you can manually confirm a PO after you receive confirmation from the vendor. If a vendor rejects a PO, the rejection is received together with the reason for the rejection and suggestions for changes. In this case, the status of the PO remains **In external review**.

There is also an option to generate a pro-forma confirmation for an order before the actual confirmation has been processed. This option just creates a report that you can share with the vendor. It doesn't create any journal information.

After the vendor has agreed to the order, the next step is to record the PO as committed. You can complete this step by using either the **Confirmation** action or the **Confirm** action. Both these actions set the approval status of the order to **Confirmed**. Confirmation of an order initiates two additional processes:

- A journal is created to store an exact copy of what was confirmed in the system. Sometimes, orders require changes, and additional journals are created after the updated order is confirmed. These journals let you view the history of the various versions of the order that were confirmed.
- Accounting distributions are created, and order checks and budget checks occur if this functionality has been enabled. If either check fails, you receive an error message that states that changes must be made to the PO before it can be confirmed again.

A vendor might request some type of assurance that payment will be provided for a purchase. There are various methods for providing this guarantee within accounts payable processes. For example, the **Prepayment** action reserves funds for the PO, and this prepayment is recorded on the PO.

## Changing purchase orders

In some situations, you might have to change a PO after it has reached an approval status of **Approved** or **Confirmed**.

If the PO was created by using a change management process, you can make changes by recalling the order or, if the order has already been approved, by using the **Request change** action. In this case, the approval status is changed back to **Draft**, and you can then modify the order. After you've finished making changes, you might have to submit the PO for re-approval. You can configure the types of changes that require re-approval by using a **Re-approval rule for purchase orders** policy rule on the **Purchasing policies** page.

If part of the ordered quantity for a PO line has been delivered, you can't change the ordered quantity when the purchase order is in **Draft**. However, you can change the **Deliver remainder** quantity on the line for the purchase order that is in **Draft** status.

After an order has been confirmed, you can no longer delete it. However, you can cancel the total quantity or any remaining quantity on an order, provided that the quantity hasn't been received or invoiced. You can then use the **Finalize** action to prevent further processing.

## Canceling purchase orders

A PO can be canceled by using the **Cancel** action on the header.

If the quantity has been partially registered, received, or invoiced, you can cancel only the remaining quantity that hasn't been registered, received, or invoiced. The order quantity is then reduced accordingly. When the quantity on the line is updated, the line status is also updated. For example, the original quantity on the line is 5,

and a quantity of 3 is received. In this case, only two can be canceled. The line is then updated to **Received** status.

If a delivery remainder is added to the order line, and it exceeds the quantity on the order line, the **Cancel** action doesn't cancel the excess quantity. Instead, the line remains in **Open order** status, because it has a remaining quantity. For example, the original quantity on the line is 5, and the delivery remainder is 7. If the order is canceled, five are canceled, and a quantity of 2 remains, as you can see in the inventory transactions.

To cancel the whole quantity on a PO line, you should cancel the delivery remainder quantity on the line. The line will then be updated to **Canceled** status.

If a PO is under change management, any change, such as cancellation of the order or the delivery remainder, must be submitted to the workflow system and approved before the process can be completed and the inventory transactions can be updated as canceled.

## Additional resources

[Purchase order overview](#)

[Create purchase orders](#)

[Product receipt against purchase orders](#)

[Vendor invoices overview](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Purchase order approval mobile workspace

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides information about the **Purchase order approval** mobile workspace. This workspace lets you view purchase orders and respond to them through actions. For example, you can approve or reject a purchase order.

## Overview

Purchase orders that requires approval go through an approval workflow. The workflow can include various steps that require that one or more people take action. For example, a person might have to complete a task or approve the purchase order.

The **Purchase order approval** mobile workspace lets you easily view and respond to purchase orders from your mobile device. This workspace also lets you take the same workflow actions that you can take from the web client.

## Prerequisites

The prerequisites vary, depending on the version of Supply Chain Management that has been deployed for your organization.

### Prerequisites if you use Supply Chain Management

If Supply Chain Management has been deployed for your organization, the system administrator must publish the **Purchase order approval** mobile workspace. For instructions, see [Publish a mobile workspace](#).

### Prerequisites if you use Microsoft Dynamics 365 for Operations version 1611 with Platform update 3 or later

If Microsoft Dynamics 365 for Operations version 1611 with Platform update 3 or later has been deployed for your organization, the system administrator must complete the following prerequisites.

PREREQUISITE	ROLE	DESCRIPTION
Implement KB 4017918.	System administrator	KB 4017918 is an X++ update or metadata hotfix that contains the <b>Purchase order approval</b> mobile workspace. To implement KB 4017918, your system administrator must follow these steps. <ol style="list-style-type: none"><li>1. <a href="#">Download the metadata hotfix from Microsoft Dynamics Lifecycle Services (LCS)</a>.</li><li>2. <a href="#">Install the metadata hotfix</a>.</li><li>3. <a href="#">Create a deployable package</a> that contains the <b>SCM Mobile</b> model, and then upload the deployable package to LCS.</li><li>4. <a href="#">Apply the deployable package</a>.</li></ol>
Publish the <b>Purchase order approval</b> mobile workspace.	System administrator	See <a href="#">Publish a mobile workspace</a> .



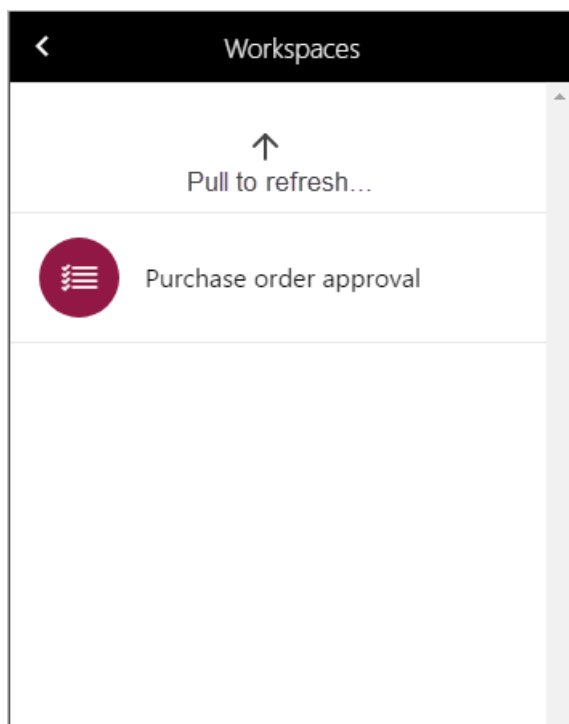
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Microsoft Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



## View orders that are assigned to you

1. On your mobile device, select the **Purchase order approval** workspace.
2. Select **Orders assigned to me** to view all the purchase orders for which you've been asked to take action in the purchase order approval workflow.
3. Select an order. On the **Order details** page, you will see the order header information and lines. You can also find guidelines from the workflow task.
4. Select **Accounting distributions** to open the **Header accounting distributions** page.
5. Return to the **Order details** page, and select a line. From the order line details, you can also explore the line-specific accounting distributions.

## Complete an action on the purchase order

After you've viewed the purchase order that is assigned to you and read the workflow instructions, you should be ready to take action.

1. On your mobile device, select the **Purchase order approval** workspace.
2. Select **Orders assigned to me** to view all the purchase orders for which you've been asked to take

action in the purchase order approval workflow.

3. Select an order, and view the details page.
4. Select **Actions** to show the available actions. The actions that are available depend on the task that has been assigned to you.

TASK ACTION	APPROVAL ACTION
Complete	Approve
Return	Reject
Request change	Request change
Delegate	Delegate

5. Select the appropriate action.
6. On the **Complete task** page, enter a comment. Note that if you select the **Delegate** action, you must select a user to delegate the task to.
7. Select **Done**. After you refresh your workspace, the purchase order will no longer be in your list.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create product packages for purchase orders

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through creating a product package and using it on a purchase order. The purchase order will be used to create an order for a pre-defined set of products. This procedure uses the USRT demo data company.

## Create a product package

1. Go to Retail and Commerce > Inventory management > Replenishment > Product packages.
2. Click New.
3. In the Package number field, type a value.
4. In the Description field, type a value.
5. In the Vendor account field, click the drop-down button to open the lookup.
6. In the list, click the link in the selected row.
7. Click Add.
8. In the Item number field, type '0160'.
9. In the Size field, click the drop-down button to open the lookup.
10. In the list, click the link in the selected row.
11. In the Quantity field, enter a number.
12. Click Add.
13. In the Item number field, type '0160'.
14. In the Variant number field, click the drop-down button to open the lookup.
15. In the list, click the link in the selected row.
16. In the Quantity field, enter a number.
17. Click Add.
18. In the Item number field, type '0175'.
19. In the Quantity field, enter a number.
20. Click Save.
21. Close the page.

## Add package to purchase order

1. Go to Accounts payable > Purchase orders > All purchase orders.
2. Click New.
3. In the Vendor account field, click the drop-down button to open the lookup.
4. In the list, select the same vendor that the product package was previously created for, if a vendor was selected.
5. Toggle the expansion of the General section.
6. In the Site field, click the drop-down button to open the lookup.
7. In the list, click the link in the selected row.
8. In the Warehouse field, click the drop-down button to open the lookup.
9. In the list, click the link in the selected row.
10. Click OK.
11. Toggle the expansion of the Line details section.

12. Click the Product packages tab.
13. Click Purchase order line.
14. Click Create lines from package.
15. In the list, find and select the product package created in previous step.
16. In the Quantity field, enter a number.
17. Click Create.
18. Click Save.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customer orders in Point of Sale (POS)

2/18/2021 • 12 minutes to read • [Edit Online](#)

This topic provides information about how to create and manage customer orders in Point of Sale (POS). Customer orders can be used to capture sales where shoppers want to pick up products on a later date, pick up products from a different location, or have items shipped to them.

In an omni-channel commerce world, many retailers provide the option of customer orders, or special orders, to meet various product and fulfillment requirements. Here are some typical scenarios:

- A customer wants products to be delivered to a specific address on a specific date.
- A customer wants to pick up products from a store or location that differs from the store or location where the customer purchased those products.
- A customer inside a store location wants to order products today and pick them up from the same store location on a later date.

Retailers can use customer orders to minimize lost sales that stock outages might otherwise cause, because the merchandise can be delivered or picked up at a different time or place.

## Set up customer orders

Before you try to use customer order functionality in POS, make sure that you complete all the required configurations in Commerce headquarters.

### Configure modes of delivery

To use customer orders, you must configure modes of delivery that the store channel can use. You must define at least one mode of delivery that can be used when order lines are shipped to a customer from a store. You must also define at least one pickup mode of delivery that can be used when order lines are picked up from the store. Modes of delivery are defined on the **Modes of delivery** page in Commerce headquarters. For more information about how to set up modes of delivery for Commerce channels, see [Define delivery modes](#).

Mode of delivery	Description	Charge group	Expedit
60	Customer pickup		

Retail channels			
✓	Name	Party ID	
<input checked="" type="checkbox"/>	Fabrikam call center	000001741	
<input checked="" type="checkbox"/>	Fabrikam extended online store	000002231	
<input checked="" type="checkbox"/>	Fashion call center	000002255	
<input checked="" type="checkbox"/>	Contoso online store	000001115	
<input checked="" type="checkbox"/>	Fabrikam online store	000001568	
<input checked="" type="checkbox"/>	Outlet	000003666	
<input checked="" type="checkbox"/>	Super	000003667	

Products					
✓	Category	Product	Product variant	Name	Line type
<input checked="" type="checkbox"/>	ALL				Include
<input checked="" type="checkbox"/>	Ties	\$1091		Black Striped Tie	Exclude

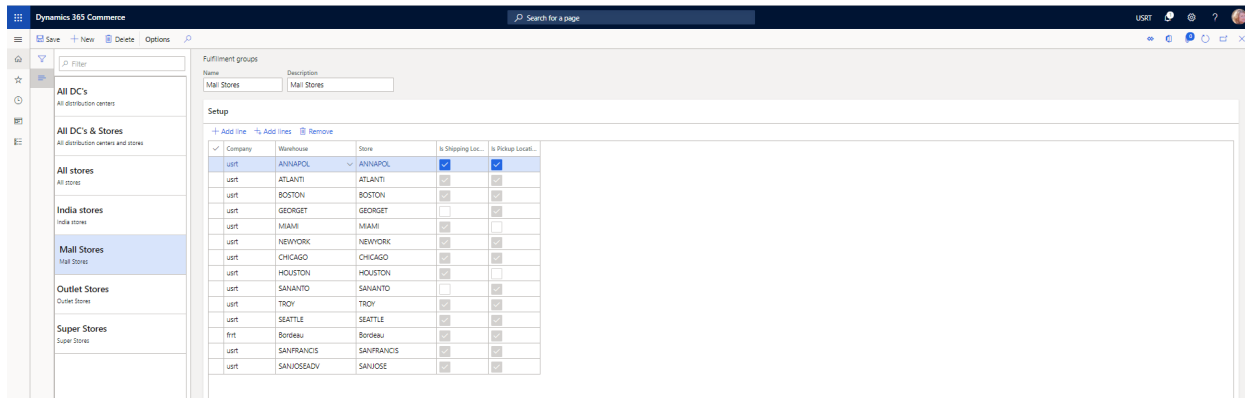
Addresses			
✓	Country/region	State	Line type
<input checked="" type="checkbox"/>	USA		Include

### Set up fulfillment groups

Some stores or warehouse locations might not be able to fulfill customer orders. By configuring fulfillment groups, an organization can specify which stores and warehouse locations are shown as options to users who

create customer orders in POS. Fulfillment groups are configured on the **Fulfillment groups** page. Organizations can create as many fulfillment groups as they require. After a fulfillment group is defined, link it to a store by selecting **Fulfillment group assignment** from the **Set up** tab on the Action Pane of the **Stores** page.

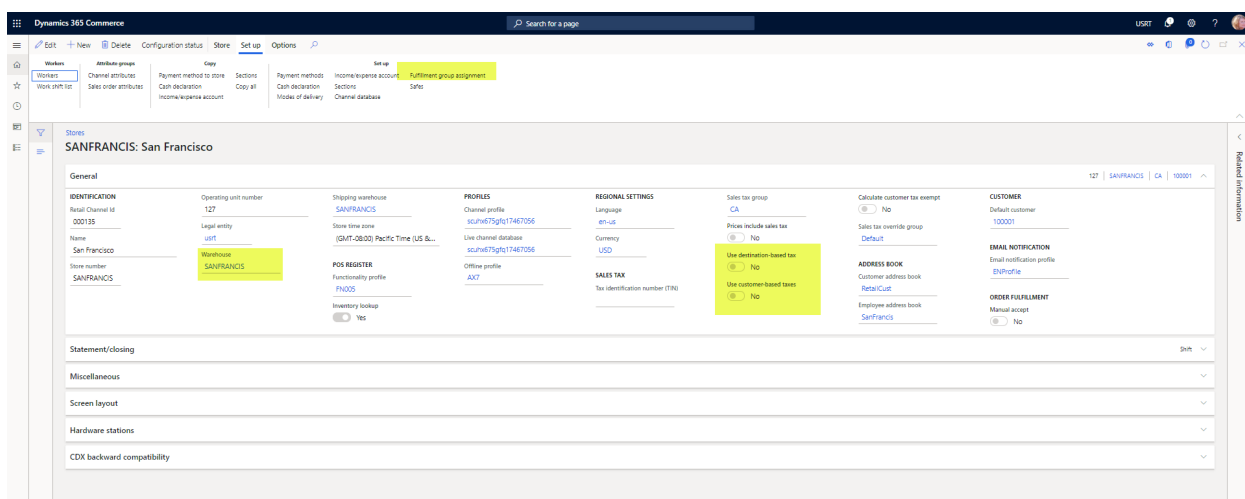
In Commerce version 10.0.12 and later, organizations can define whether the warehouse or warehouse and store combinations that are defined in fulfillment groups can be used for shipping, for pickup, or for both shipping and pickup. This allows for added flexibility for the business to determine which warehouses can be selected when creating a customer order for items to ship vs. which stores can be selected when creating a customer order for items to pick up. To use these configuration options, turn on the **Ability to specify locations as "Shipping" or "Pickup" enabled within Fulfillment group** feature. If a warehouse that's linked to a fulfillment group isn't a store, it can be configured only as a shipping location. It can't be used when orders for pickup are configured in POS.



## Configure channel settings

When you work with customer orders in POS, you must consider some of the settings of the store channel. These settings are found on the **Stores** page in Commerce headquarters.

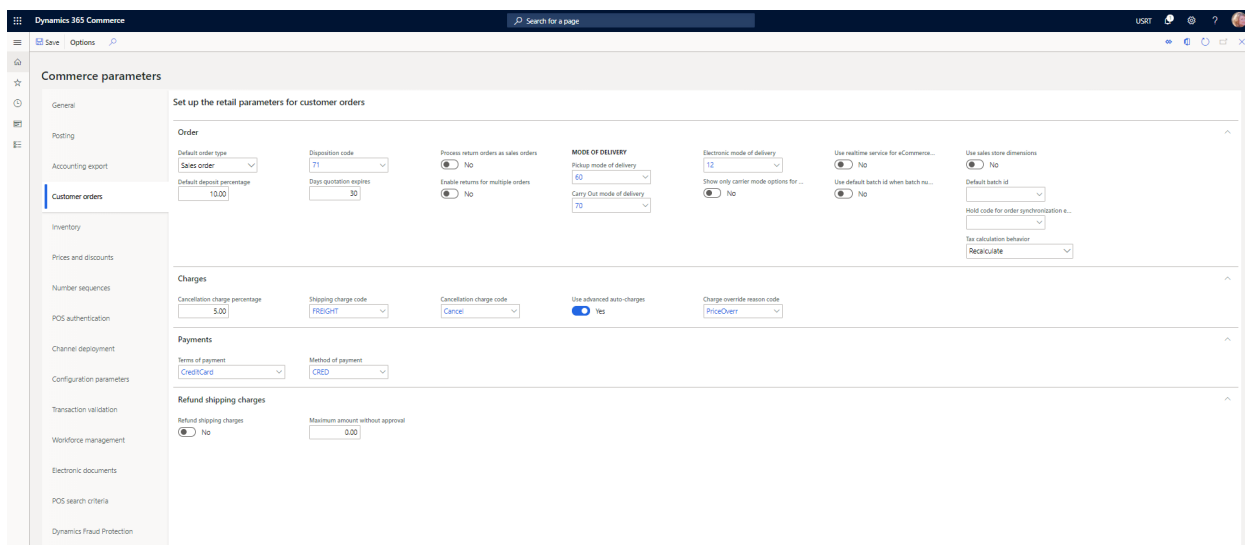
- **Warehouse** – This field indicates the warehouse that is used to fulfill orders that are configured for shipment from the store.
- **Fulfillment group assignment** – Select this button (on the **Set up** tab on the Action Pane) to link the fulfillment groups that are referenced to show options for pickup locations or shipment origins when customer orders are created in POS.
- **Use destination-based tax** – This option indicates whether the shipping address is used to determine the tax group that is applied to order lines that are shipped to the customer's address.
- **Use customer-based tax** – This option indicates whether the tax group that is defined for the customer's delivery address is used to tax customer orders that are created in POS for shipment to the customer's home.



## Set up customer order parameters

Before you try to create customer orders in POS, you must configure the appropriate parameters in Commerce headquarters. These parameters can be found on the **Customer orders** tab of the **Commerce parameters** page.

- **Default order type** – You can specify the order type that is assigned by default to customer orders that are created in POS. These customer orders can be either sales orders or quotation orders. Regardless of the default order type, users can still create both sales orders and customer orders from POS.
- **Default deposit percentage** – Specify the percentage of the order total amount that the customer must pay as a deposit before an order can be confirmed. Depending on their privileges, store associates might be able to override the amount by using the **Deposit override** operation in POS, if that operation is configured for the transaction screen layout.
- **Pickup mode of delivery** – Specify the mode of delivery that should be applied to sales order lines that are configured for pickup in POS.
- **Carryout mode of delivery** – Specify the mode of delivery that should be applied to sales order lines that are considered carryout order lines when a mixed cart is created, where some lines will be picked up or shipped, and other lines will be carried out by the customer immediately.
- **Cancellation charge percentage** – If a charge should be applied when a customer order is canceled, specify the amount of that charge.
- **Cancellation charge code** – Specify the Accounts receivable charge code that should be used when a cancellation charge is applied to canceled customer orders through POS. The charge code defines the financial posting logic for the cancellation charge.
- **Shipping charge code** – If the **Use advanced auto charges** option is set to **Yes**, this parameter setting has no effect. If that option is set to **No**, users will be prompted to manually enter a shipping charge when they create customer orders in POS. Use this parameter to map an Accounts receivable charge code that will be applied to orders when users enter a shipping charge. The charge code defines the financial posting logic for the shipping charge.
- **Use advanced auto charges** – Set this option to **Yes** to use system-calculated auto charges when customer orders are created in POS. These auto charges can be used to calculate shipping fees or other order or item-specific charges. For more information about how to set up and use advanced auto charges, see [Omni-channel advanced auto charges](#).

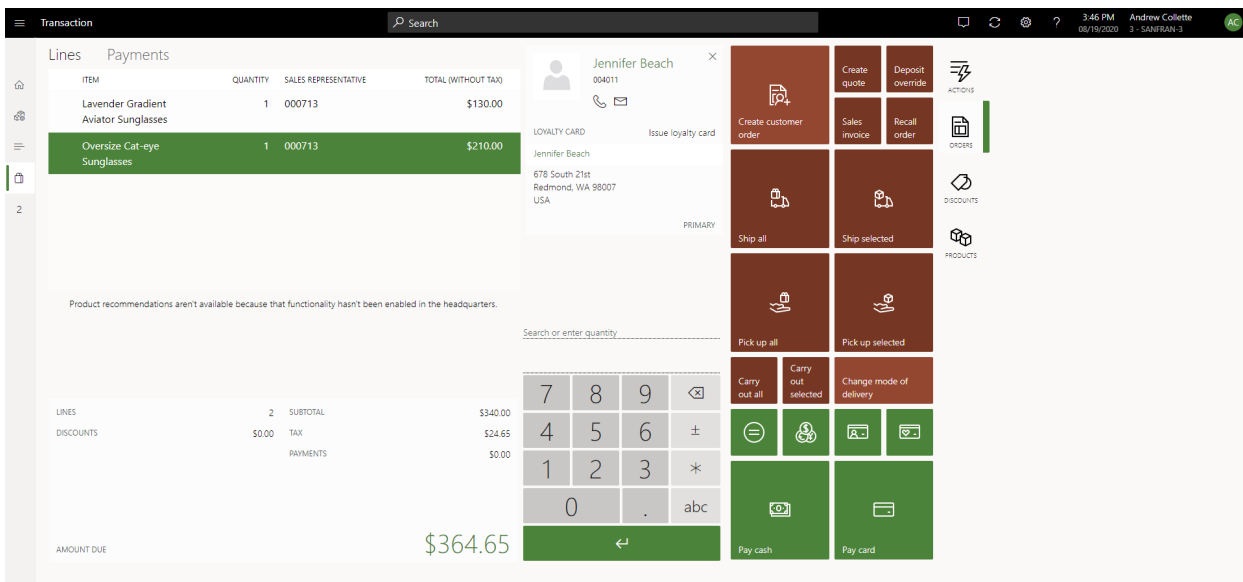


## Update transaction screen layouts in POS

Make sure that the POS [screen layout](#) is configured to support the creation and management of customer orders, and that all required POS operations are configured. Here are some of the POS operations that are recommended to correctly support customer order creation and management:

- **Ship all products** – This operation is used to specify that all lines in the transaction cart will be shipped to a destination.

- **Ship selected products** – This operation is used to specify that selected lines in the transaction cart will be shipped to a destination.
- **Pick up all products** – This operation is used to specify that all lines in the transaction cart will be picked up from a selected store location.
- **Pick up selected products** – This operation is used to specify that selected lines in the transaction cart will be picked up from a selected store location.
- **Carry out all products** – This operation is used to specify that all lines in the transaction cart will be carried out. If this operation is used in POS, the customer order will be converted to a cash-and-carry transaction.
- **Carryout out selected products** – This operation is used to specify that selected lines in the transaction cart are being carried out by the customer at the time of purchase. This operation is useful only in a [hybrid order](#) scenario.
- **Recall order** – This operation is used to search and retrieve customer orders so that POS users can edit, cancel, or perform fulfillment-related operations on them as required.
- **Change mode of delivery** – This operation can be used to quickly change the mode of delivery for lines that are already configured for shipment, without requiring that users go through the "ship all products" or "ship selected products" flow again.
- **Deposit override** – This operation can be used to change the deposit amount that the customer will pay for the selected customer order.



## Work with customer orders in POS

### NOTE

Revenue recognition functionality isn't currently supported for use in Commerce channels (e-commerce, POS, call center). Items configured with revenue recognition shouldn't be added to orders created in Commerce channels.

### Create a customer order for products that will be shipped to the customer

1. On the POS transaction screen, add a customer to the transaction.
2. Add products to the cart.
3. Select **Ship selected** or **Ship all** to ship the products to an address on the customer account.
4. Select the option to create a customer order.
5. Confirm or change the "ship from" location, confirm or change the shipping address, and select a shipping method.
6. Enter the customer's desired order shipment date.
7. Use the payment functions to pay for any calculated amounts that are due, or use the **Deposit override**



operation to change the amounts that are due, and then apply payment.

8. If the full order total wasn't paid, enter a credit card that will be captured for the balance that is due on the order when it's invoiced.

### Create a customer order for products that the customer will pick up

1. On the POS transaction screen, add a customer to the transaction.
2. Add products to the cart.
3. Select **Pick up selected** or **Pick up all** to initiate the order pick up configuration.
4. Select the store location where the customer will pick up the selected products.
5. Select a date when the item will be picked up.
6. Use the payment functions to pay for any calculated amounts that are due, or use the **Deposit override** operation to change the amounts that are due, and then apply payment.
7. If the full order total wasn't paid, select whether the customer will provide payment later (at pick up), or whether a credit card will be tokenized now, and then used and captured at the time of pickup.

### Edit an existing customer order

Retail orders that are created in either the online or store channel can be recalled and edited through POS as required.

#### IMPORTANT

Not all retail orders can be edited through the POS application. Orders that are created in a call center channel can't be edited through POS if the [Enable order completion](#) setting is turned on for the call center channel. To ensure correct payment processing, orders that originated in a call center channel and that use Enable order completion functionality must be edited through the call center application in Commerce headquarters.

In version 10.0.17 and later, users can edit eligible orders through the POS application, even if the order is partially fulfilled. However, orders that are fully invoiced still can't be edited through POS. To enable this capability, turn on the **Edit partially fulfilled orders in Point of Sale** feature in the **Feature management** workspace. If this feature is not enabled, or if you're using version 10.0.16 or earlier, users will only be able to edit customer orders in POS if the order is fully open. Further, if the feature is enabled, you can limit which stores can edit partially fulfilled orders. The option to disable this capability for specific stores can be configured through the **Functionality profile** under the **General** FastTab.

1. Select **Recall order**.
2. Use **Search** to enter filters to find the order, and then select **Apply**.
3. Select the order in the list of results, and then select **Edit**. If the **Edit** button is unavailable, the order is in a state where it can't be edited.
4. From the transaction cart, make any necessary changes to the customer order. Some changes might be prohibited during editing.
5. Complete the editing process by selecting a payment operation.
6. To exit the editing process without saving any changes, you can use the **Void transaction** operation.

### Cancel a customer order

1. Select **Recall order**.
2. Use **Search** to enter filters to find the order, and then select **Apply**.
3. Select the order in the list of results, and then select **Cancel**. If the **Cancel** button is unavailable, the order is in a state where it can no longer be canceled.
4. If cancellation charges are configured, confirm them. You can adjust the cancellation charges before you confirm them, as required.
5. From the transaction cart, complete the cancellation process by selecting a payment operation. If deposits

that were paid exceed the cancellation charge, refund payments might be due.

6. To exit the cancellation process without saving any changes, you can use the **Void transaction** operation.

## Finalizing the customer order shipment or pickup from POS

After an order is created, the items will be picked up by the customer from a store location or shipped, depending on the configuration of the order. For more information about this process, see the [store order fulfillment](#) documentation.

## Asynchronous transaction flow for customer orders

Customer orders can be created in POS in either synchronous mode or asynchronous mode. If you notice performance issues or user delays when you create customer orders in POS, consider turning on asynchronous order creation.

### Enable customer orders to be created in asynchronous mode

1. In Commerce headquarters, on the **Functionality profiles** page, select the functionality profile that corresponds to the store that you want to configure.
2. On the **General** FastTab, set the **Create customer order in async mode** option to **Yes**.

When the **Create customer order in async mode** option is set to **Yes**, customer orders are always created in asynchronous mode, even if Retail Transaction Service (RTS) is available. If you set this option to **No**, customer orders are always created in synchronous mode by using RTS. When customer orders are created in asynchronous mode, they're pulled and created as retail transactions in Commerce headquarters from the Commerce Pull (P) jobs. The corresponding sales orders for the retail transactions are created when **Synchronize orders** is run either manually or through a batch process.

## Additional resources

[Hybrid customer orders](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Hybrid customer orders

2/18/2021 • 2 minutes to read • [Edit Online](#)

A hybrid customer order is a single order, which contains products that can be carried out of the store by the customer, as well as products that will be picked up or shipped later.

In Commerce, you can select either carry out all products or carry out selected products for a customer order. The product lines that are marked as carry out are automatically invoiced after the order is created, similarly this is the same for an order that is to be picked-up after the order is created. The amount due on hybrid orders is determined by adding the deposit percentage on pick and ship product lines with the full amount of the carry out lines. For hybrid orders, the system switches between customer order mode and cash and carry mode as follows:

- If all products in the cart are set to **Carry out delivery**, the order will be handled as a Cash and Carry transaction.
- If any or all lines in the cart are set to either **Pick** or **ship delivery**, the order will be handled as a Customer order transaction.

If a cart line is selected and **Pick selected**, **Ship selected**, or **Carry out selected** is selected, only the specific cart line is set with that delivery method. In that case, the downstream flow of the operation continues as usual. However, if **Pick selected**, **Ship selected**, or **Carry out selected** is selected without a cart line being selected, a new page opens that lists all the cart lines. On that screen, you can select multiple lines at once for setting the delivery method. When you use that method for selecting lines, any previous delivery method that has been assigned to the line will be overridden.

## Additional resources

[Customer orders in Modern POS \(MPOS\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Recall order operation in POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

The **Recall order** operation in the Commerce point of sale (POS) provides updated order search and filtering features and order-specific information. This feature is available in Commerce versions 10.0.15 and later.

To enable this functionality, turn the **Improved Recall order operation in POS** feature on in **Feature management** workspace in Commerce headquarters. After you enable the feature, consider updating your [screen layouts](#) in POS to take advantage of some of the changed capabilities.

The configuration of the **Recall order** operation button allows organizations to deploy the operation with a pre-defined display.

The screenshot shows the 'Configure button' dialog box. The 'Action' dropdown is set to 'Recall order'. The 'Default display' dropdown is open, showing options: 'Orders to pick up' (highlighted), 'None', 'Orders to fulfill', 'Orders to pick up', and 'Orders to ship'. The 'Appearance' section includes fields for 'Button text' (Recall order), 'Image alignment' (Center), 'Size in columns' (2), 'Size in rows' (2), 'Font size' (12), 'Font style' (Regular), and 'Use custom theme' (checked). Color pickers are visible for 'Back color', 'Back color 2', 'Border color', and 'Font color'. A 'Button preview' window shows a red button with a white icon. 'OK' and 'Cancel' buttons are at the bottom.

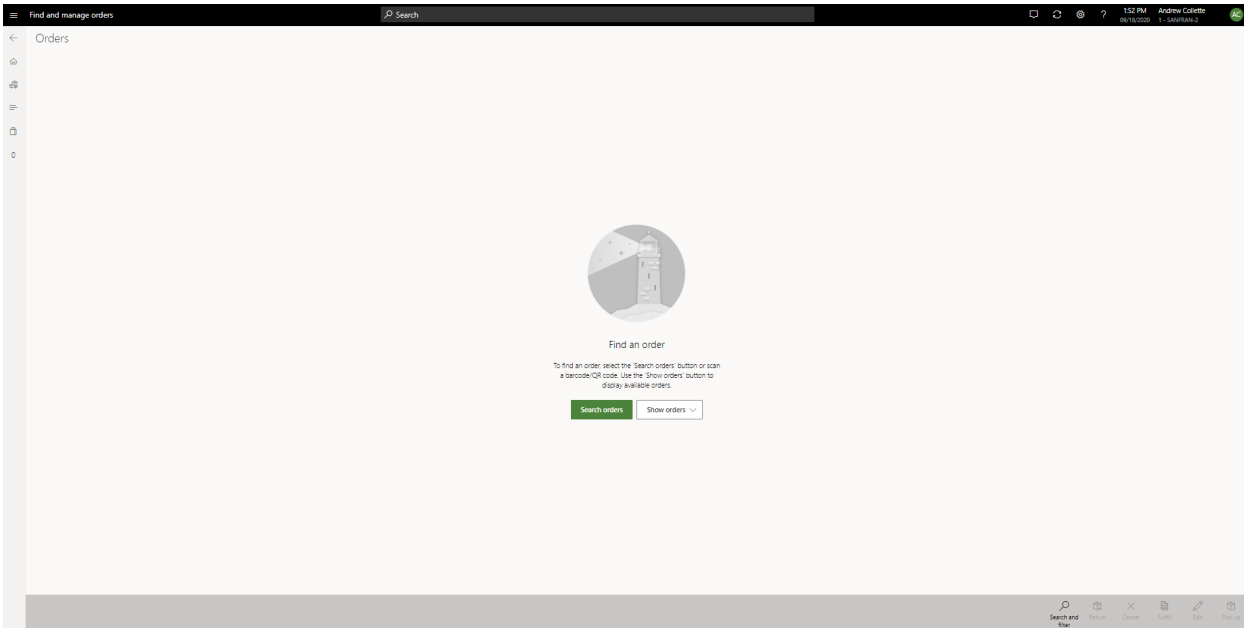
The display options are as follows.

- **None** – This option deploys the operation with no specific display. When a user opens the operation with this configuration, they will be prompted to search and find orders or choose from a pre-defined order filter.
- **Orders to fulfill** – When a user launches the operation, a query will run automatically to search and display a list of orders that are to be fulfilled by the store. These orders are configured for in-store pickup or store shipment and the lines of these orders have not yet been picked or packed.
- **Orders to pick up** – When a user launches the operation, a query will run automatically to search and display a list of orders that are configured for in-store pickup at the user's current store.
- **Orders to ship** - When a user launches the operation, a query will run automatically to search and display a list of orders that are configured for shipment from the user's current store.

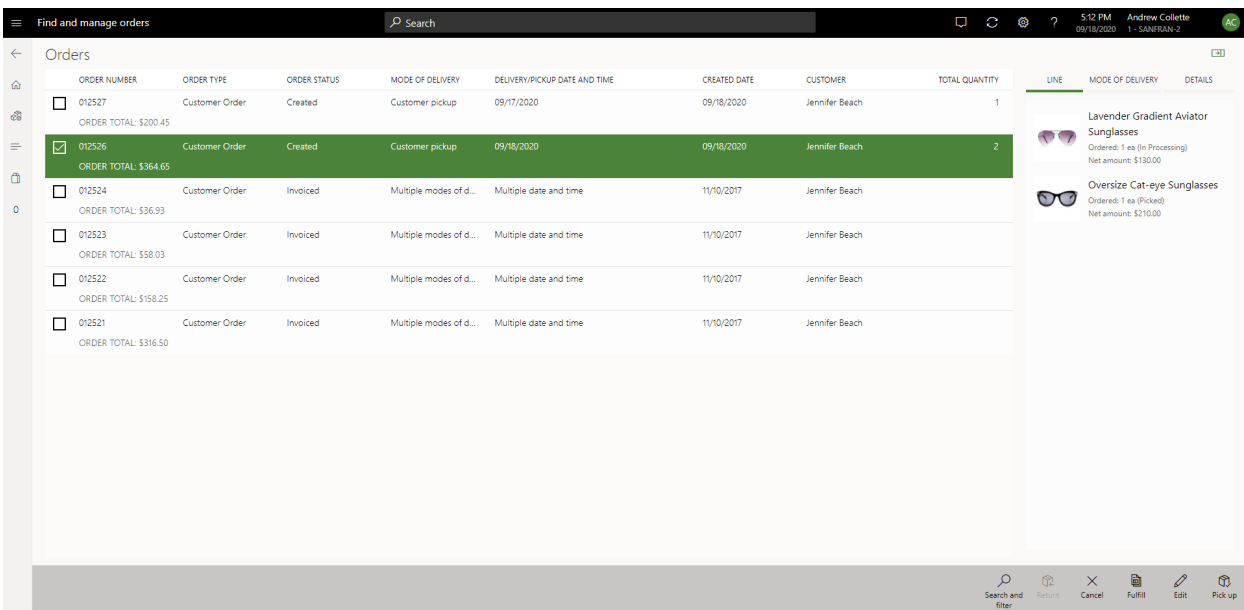
When launching the **Recall order** operation from POS, if the display is configured to **None**, a user will be able to search and retrieve orders in one of the following ways.

- Scan order barcodes. This will search order number, channel reference, and receipt ID fields for matches.
- Select **Search orders** or **Search and filter** icon on the AppBar to use the filtering mechanism to locate orders that meet the filter criteria.

- Choose from a pre-defined filter from the **Show Orders** drop-down menu (orders to fulfill, orders to pick up, or orders to ship).



After search criteria are applied, the application will display a list of matching sales orders.



A user can select an order on the list to view additional details. The information panel on the right side of the screen displays specifics on the selected order, including order line details, delivery details, and fulfillment details.

From the AppBar, a user can select an operation. Depending on the status of the order, certain operations may not be enabled.

- **Return** – Executes a return for one or more invoices related to the selected customer order.
- **Cancel** – Issue a full cancellation of the selected sales order.
- **Fulfill** – Transfers the user to the order fulfillment page, which will be pre-filtered for the selected order. Only order lines that are open for fulfillment by the user's store for the selected order will be displayed.
- **Edit** – Allows users to make changes to the selected customer order.
- **Pick up** – Launches the pickup flow, which allows the user to choose the products to be picked up and creates the pickup sales transaction.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable multiple pickup delivery modes for customer orders

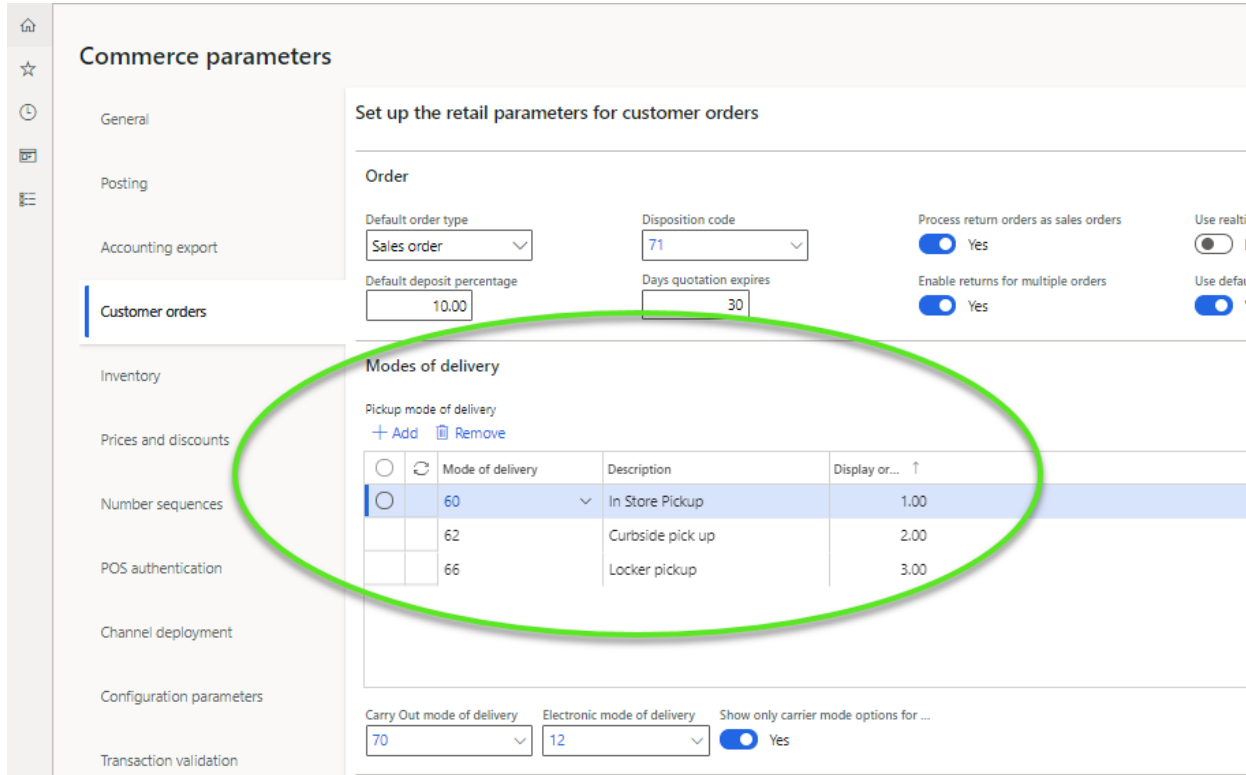
2/18/2021 • 5 minutes to read • [Edit Online](#)

In Microsoft Dynamics 365 Commerce version 10.0.16 and later, organizations can define multiple modes of delivery that shoppers or sales associates can choose among when they create an order that will be picked up at a store. In this way, organizations can provide multiple pickup options to their shoppers. For example, many retailers now offer shoppers the choice of in-store pickup or curbside pickup for their orders. Commerce supports the configuration of these different pickup delivery modes. Users can then take advantage of them when they create customer orders in any supported Commerce channel (e-commerce, call center, or store).

## Enable and configure pickup delivery modes

To use this functionality, turn on the **Support for multiple pickup delivery modes** feature in the **Feature management** workspace in Commerce headquarters. After you turn on the feature, additional configuration is required.

In Commerce version 10.0.15 and earlier, organizations can define only one mode of delivery as the designated pickup delivery mode. This definition is done on the **Commerce parameters** page. In version 10.0.16 and later, when you turn on the **Support for multiple pickup delivery modes** feature, the mode of delivery that was previously defined as the pickup delivery mode on the **Commerce parameters** page is automatically copied into the new configuration for pickup delivery modes.



The screenshot displays the 'Commerce parameters' page in Microsoft Dynamics 365 Commerce. The left-hand navigation pane includes sections for General, Posting, Accounting export, Customer orders (highlighted), Inventory, Prices and discounts, Number sequences, POS authentication, Channel deployment, Configuration parameters, and Transaction validation. The main content area is titled 'Set up the retail parameters for customer orders' and is divided into several sections:

- Order:** Contains fields for Default order type (Sales order), Disposition code (71), Default deposit percentage (10.00), and Days quotation expires (30). It also features four toggle switches: 'Process return orders as sales orders' (Yes), 'Use realti' (off), 'Enable returns for multiple orders' (Yes), and 'Use defat' (Yes).
- Modes of delivery:** This section is circled in green. It includes a 'Pickup mode of delivery' section with '+ Add' and '- Remove' buttons. Below is a table with columns for 'Mode of delivery', 'Description', and 'Display or...'. The table contains three rows:

Mode of delivery	Description	Display or...
60	In Store Pickup	1.00
62	Curbside pick up	2.00
66	Locker pickup	3.00
- Bottom section:** Contains 'Carry Out mode of delivery' (70), 'Electronic mode of delivery' (12), and a 'Show only carrier mode options for ...' toggle switch (Yes).

After you turn on the **Support for multiple pickup delivery modes** feature, you can define multiple pickup delivery modes in the **Pickup mode of delivery** grid on the **Modes of delivery** FastTab on the **Customer orders** tab of the **Commerce parameters** page.

The **Carry Out mode of delivery** and **Electronic mode of delivery** fields, and the **Show only carrier mode options for ship orders** option, have been relocated to this FastTab.

Before you configure additional pickup delivery modes, you must define the modes of delivery. On the **Modes of delivery** page in Commerce headquarters, add the modes of delivery that should be considered pickup delivery modes. Make sure that all configuration is completed. For example, make sure that the mode of delivery is linked to appropriate channels and items. When you've finished, run the **Process delivery modes** job to create the relationships among the mode of delivery, channels, and items. When the job has finished running, open the **Distribution schedule** page in Commerce headquarters, and run the **1120** distribution job to ensure that the relevant Commerce channel databases are updated with your new delivery mode configuration.

Modes of delivery

Mode of delivery	Description	Charges group	Expedite
62	Curbside pick up		

---

**Retail channels**

[Add line](#) [Remove](#)

<input type="radio"/> Name	Party ID
<input checked="" type="radio"/> Houston	000001099
<input type="radio"/> Fabrikam online store	000001568
<input type="radio"/> Fabrikam call center	000001741
<input type="radio"/> San Francisco	000002230
<input type="radio"/> Fabrikam extended online store	000002231
<input type="radio"/> Fashion call center	000002255

---

**Products**

[+ Add line](#) [+ Add products](#) [Remove](#)

<input type="radio"/> Category	Product	Product variant	Name	Line type
<input checked="" type="radio"/> ALL				Include
<input type="radio"/> ALL	Digital Gift Card		Digital Gift Card	Exclude
<input type="radio"/> Gift Cards	Digital Gift Card		Digital Gift Card	Exclude
<input type="radio"/> Gift Cards	E-mail Gift Card		E-mail Gift Card	Exclude

---

**Addresses**

[+ Add line](#) [Remove](#)

<input type="radio"/> Country/region	State	Line type
<input checked="" type="radio"/> USA		Include

After you define the additional pickup delivery modes, add them to the **Pickup mode of delivery** grid on the **Commerce parameters** page. Then run the appropriate distribution jobs to update the relevant Commerce channel databases with the configuration change.

**NOTE**

Apart from the existing pickup delivery mode that is copied to the **Pickup mode of delivery** grid when you turn on the **Support for multiple pickup delivery modes** feature, for every additional pickup delivery mode configuration that you create, you should configure new modes of delivery. When you add modes of delivery to the **Pickup mode of delivery** grid, Commerce validates whether any active open sales lines already use them. If any open sales lines are found, you receive an error message. Modes of delivery aren't considered pickup delivery modes until all open sales lines that use them have been closed (either invoiced or canceled).



### IMPORTANT

After you define more than one pickup delivery mode on the **Commerce parameters** page, the **Support for multiple pickup delivery modes** feature becomes mandatory and can no longer be turned off. If you must turn off the feature, remove all but one pickup delivery mode from the **Pickup mode of delivery** grid. When only one pickup delivery mode is defined, the feature is longer considered mandatory and can be turned off.

### E-commerce site configurations

When the **Support for multiple pickup delivery modes** feature is turned on, the following modules on e-commerce pages show the new pickup delivery modes as configured:

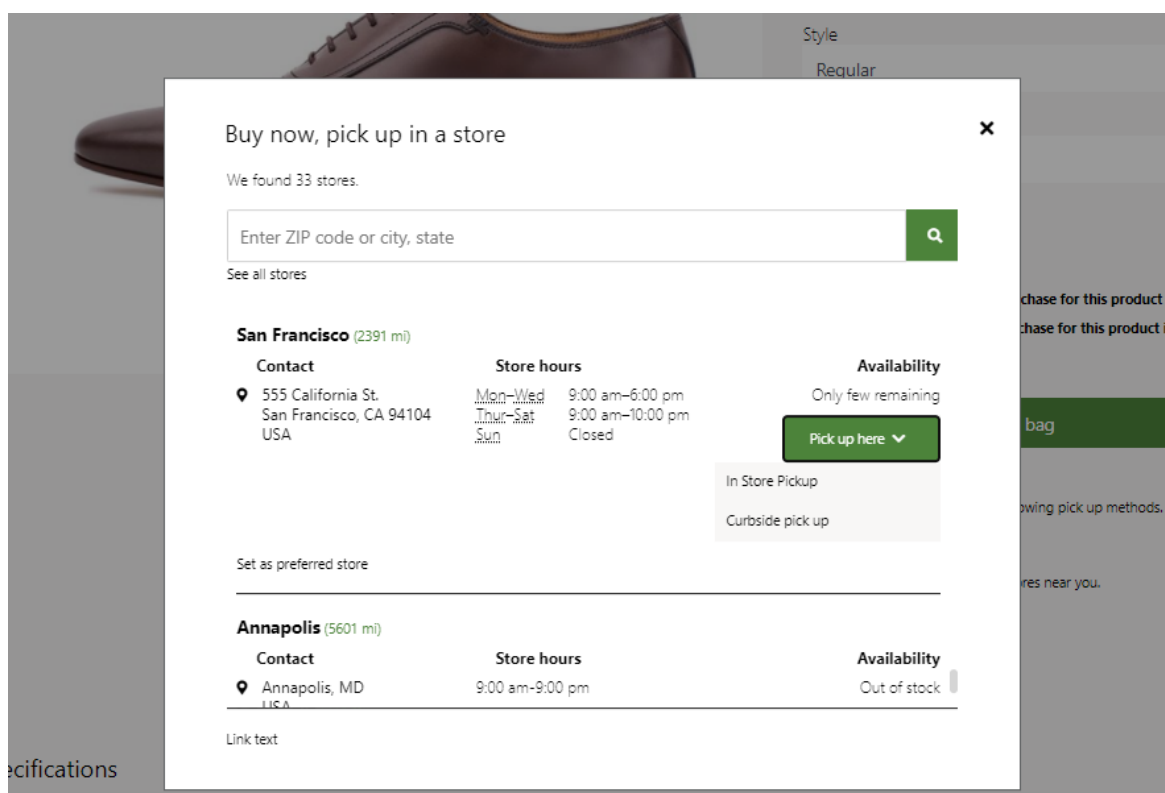
- Buy box
- Store selector
- Cart
- Pickup information
- Order confirmation
- Order details

No additional steps are required on e-commerce pages to make the pickup delivery modes available.

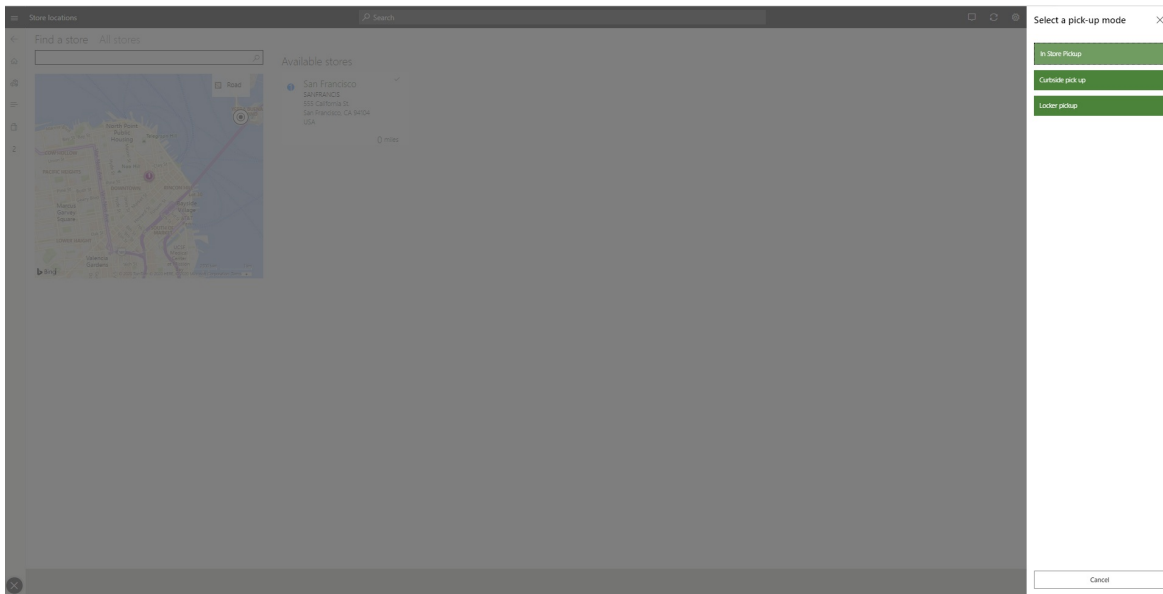
## Work with multiple pickup delivery modes

When multiple pickup delivery modes are available for a channel, an enhanced experience is provided to customers when they shop for products that will be picked up.

- In e-commerce channels, shoppers can select any valid pickup delivery mode that is available. For example, a retailer defines two pickup delivery modes (in-store pickup and curbside pickup), both are configured in the **Pickup mode of delivery** grid, and both are valid for the order fulfillment channel and the product that a shopper is currently purchasing. In this case, the shopper can select their preferred pickup delivery mode. The selected pickup delivery mode then becomes the mode of delivery that is linked to the sales order line when the order is created in Commerce headquarters.



- In store channels, if a customer order for pickup is created through the point of sale (POS) application, the sales associate is prompted to choose among the available pickup delivery modes, if any have been configured. If only one valid pickup delivery mode is available for the channel and item, the sales associate isn't prompted to select it. Instead, the available pickup delivery mode is automatically applied to the order lines.



- In call center channels, when users create pickup orders, they can manually select any defined pickup delivery mode that is linked to the call center channel. The system then validates that the selected pickup delivery mode can be used when the item that is being linked to it is ordered. When a pickup delivery mode is selected in call center channels, the sales order lines must be linked to a valid store warehouse. If a non-store warehouse is defined on a call center sales line, a pickup delivery mode can't be set on that sales line.
- Sales associates can use the **Order recall** or **Order fulfillment** operation in the POS application to retrieve a list of orders or order lines for pickup. If a sales associate uses a predefined search filter to show all orders that will be picked up at the current store, the queries are modified to ensure that the search results include all eligible orders that use any pickup delivery mode. POS users can also use existing filters to narrow down the list of orders to a specific pickup delivery mode. For example, they can show only orders for curbside pickup.

ORDER NUMBER	ORDER TYPE	ORDER STATUS	MODE OF DELIVERY	DELIVERY/OUT DATE AND TIME	CREATED DATE	CUSTOMER	TOTAL QTY.	PAYMENT STATUS	ORDER TOTAL
02806	Customer Order	In processing	Curbside pick up	11/05/2020	11/05/2020	Default Online Customer	1	Authorized	\$279.50
028074	Customer Order	In processing	Multiple modes of delivery	Multiple date and time	11/06/2020	Default Online Customer	12	Authorized	\$2,498.97
027979	Customer Order	In processing	In Store Pickup	11/06/2020, 8:00 AM - 9:00 AM	11/05/2020	Mary Kay Andersen	1	Not submitted	\$97.34
027978	Customer Order	In processing	In Store Pickup	11/06/2020, 8:00 AM - 9:00 AM	11/05/2020	Demo Berg	2	Paid	\$74.93
027977	Customer Order	In processing	Curbside pick up	11/07/2020	11/05/2020	Jennifer Beach	1	Paid	\$70.00
028046	Customer Order	In processing	Multiple modes of delivery	Multiple date and time	11/05/2020	Hamita Test-Faizkam Test	4	Authorized	\$1,061.86
028007	Customer Order	In processing	Multiple modes of delivery	Multiple date and time	11/05/2020	testmsta jin	2	Authorized	\$676.64
027942	Customer Order	In processing	Locker pickup	11/05/2020	11/05/2020	Jennifer Beach	1	Paid	\$70.00
027941	Customer Order	In processing	Curbside pick up	11/05/2020	11/05/2020	Jennifer Beach	1	Paid	\$70.00
027976	Customer Order	In processing	Curbside pick up	11/05/2020	11/05/2020	Jennifer Beach	1	Paid	\$260.00
027926	Customer Order	In processing	In Store Pickup	11/06/2020, 8:00 AM - 9:00 AM	11/05/2020	Karen Berg	1	Paid	\$82.00
027836	Customer Order	In processing	In Store Pickup	11/06/2020, 3:00 PM - 4:00 PM	11/02/2020	okoronardo da vinici Sharma	3	Authorized	\$866.97
025838	Customer Order	In processing	In Store Pickup	10/21/2020, 10:00 AM - 11:00 AM	10/26/2020	Mary Kay Andersen	1	Not submitted	\$26.94
028827	Customer Order	In processing	Multiple modes of delivery	Multiple date and time	10/26/2020	okoronardo da vinici Sharma	3	Authorized	\$484.00
028826	Customer Order	In processing	In Store Pickup	10/21/2020, 9:00 AM - 10:00 AM	10/26/2020	Mary Kay Andersen	1	Not submitted	\$98.84
024999	Customer Order	In processing	In Store Pickup	10/24/2020, 12:00 PM - 1:00 PM	10/21/2020	okoronardo da vinici Sharma	1	Authorized	\$332.00
024823	Customer Order	In processing	In Store Pickup	10/20/2020	10/20/2020	okoronardo da vinici Sharma	1	Paid	\$78.50
024782	Customer Order	In processing	In Store Pickup	10/20/2020, 1:00 PM - 2:00 PM	10/20/2020	Default Online Customer	2	Authorized	\$336.00
024775	Customer Order	In processing	In Store Pickup	10/21/2020, 3:00 PM - 4:00 PM	10/20/2020	mkrish Lather Singh	2	Paid	\$84.10
024744	Customer Order	In processing	Multiple modes of delivery	Multiple date and time	10/20/2020	Philoj Nolasco	2	Authorized	\$243.50
024424	Customer Order	In processing	In Store Pickup	10/16/2020, 9:00 AM - 10:00 AM	10/15/2020	Mary Kay Andersen	1	Not submitted	\$64.34
024422	Customer Order	In processing	In Store Pickup	10/15/2020, 9:00 AM - 10:00 AM	10/15/2020	Demo Berg	1	Not submitted	\$64.34
024423	Customer Order	In processing	In Store Pickup	10/15/2020, 9:00 AM - 10:00 AM	10/15/2020	Mary Kay Andersen	1	Not submitted	\$26.94
024421	Customer Order	In processing	In Store Pickup	10/16/2020, 8:00 AM - 10:00 AM	10/16/2020	Demo Berg	1	Not submitted	\$33.54
024419	Customer Order	In processing	In Store Pickup	10/14/2020, 8:00 AM - 9:00 AM	10/15/2020	Corosco Retail San Diego	1	Not submitted	\$33.54
024420	Customer Order	In processing	In Store Pickup	10/16/2020, 12:00 PM - 1:00 PM	10/15/2020	Demo Berg	1	Not submitted	\$207.35
024418	Customer Order	In processing	In Store Pickup	10/16/2020, 8:00 AM - 9:00 AM	10/15/2020	Mary Kay Andersen	6	Not submitted	\$103.28

## Considerations for distributed order management

The [distributed order management \(DOM\)](#) features in Commerce ignore any sales lines that are marked for store pickup. These features have been updated to ensure that sales lines that are linked to configured pickup delivery modes bypass the DOM logic and won't be reallocated to a new fulfillment warehouse.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Process customer order pickups in POS

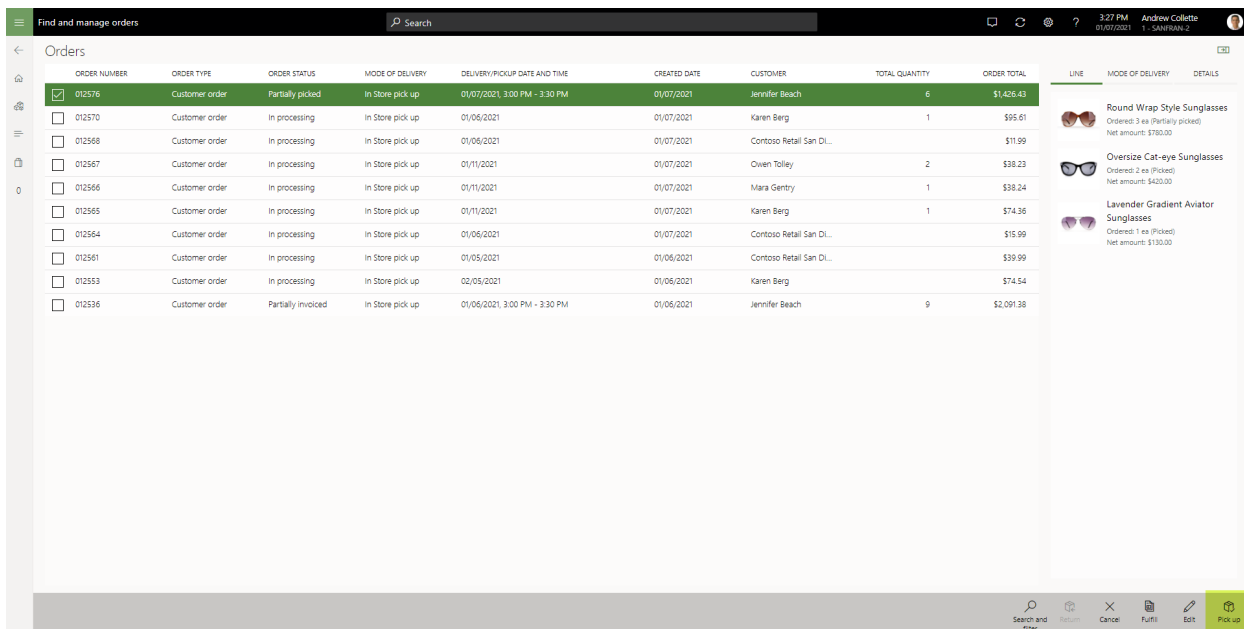
2/18/2021 • 3 minutes to read • [Edit Online](#)

When a [customer order](#) is created for store pickup, a store user can use the point of sale (POS) application to start the pickup of inventory. POS will run the final payment capture as required. It will also complete the inventory and financial posting for the quantities that are picked up.

If you're a store user, you can perform the pickup by using either the **Recall order** operation or the **Order fulfillment** operation in POS. To make the **Pick up** operation available, you must first follow one of these steps:

- To use the **Recall order** operation, search for and select the order that will be picked up.
- To use the **Order fulfillment** operation, search for and select one or more order lines.

If the selected order or order lines aren't configured for pickup at that specific store, or if the order has already been fully picked up, the **Pick up** operation will be unavailable.



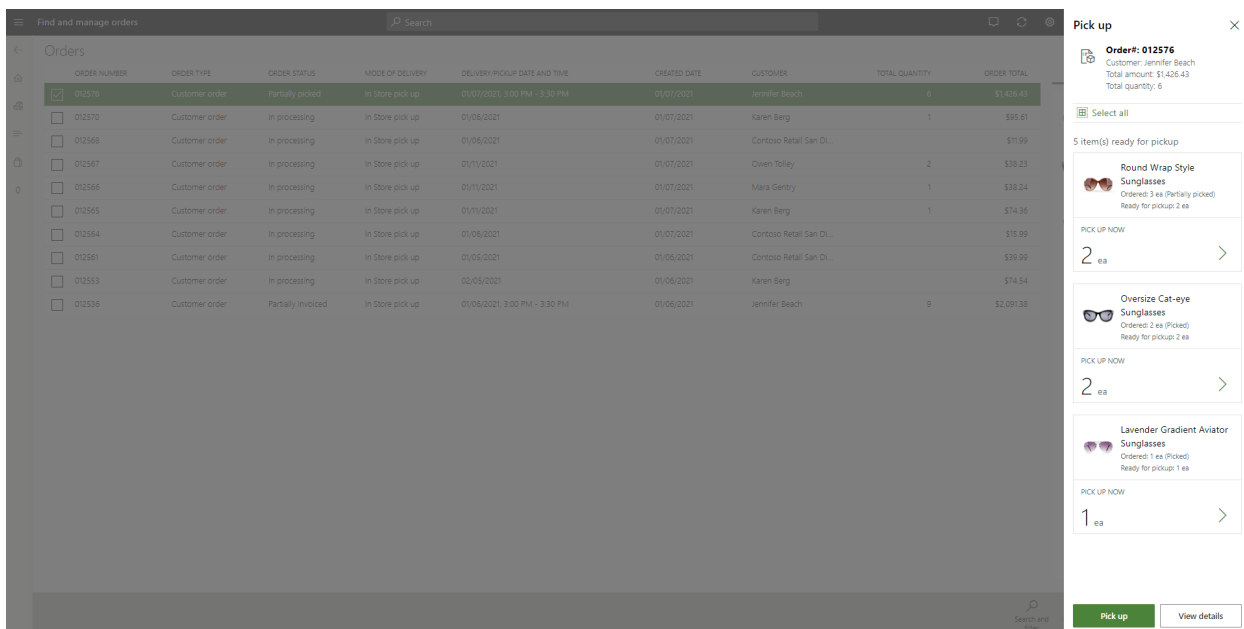
The screenshot displays the 'Find and manage orders' screen in the POS application. It features a table of orders with columns for Order Number, Order Type, Order Status, Mode of Delivery, Delivery/Pickup Date and Time, Created Date, Customer, Total Quantity, and Order Total. The first order, 012576, is highlighted in green and marked as 'Partially picked'. To the right, a detailed view of this order line is shown, listing 'Round Wrap Style Sunglasses' with a quantity of 3 (1 partially picked) and a net amount of \$780.00. Below this, 'Oversize Cat-eye Sunglasses' (2 ea, \$420.00) and 'Lavender Gradient Aviator Sunglasses' (1 ea, \$130.00) are also listed. The bottom navigation bar includes icons for Search and Filter, Return, Cancel, Fulfill, Edit, and Pick up.

ORDER NUMBER	ORDER TYPE	ORDER STATUS	MODE OF DELIVERY	DELIVERY/PICKUP DATE AND TIME	CREATED DATE	CUSTOMER	TOTAL QUANTITY	ORDER TOTAL
<input checked="" type="checkbox"/> 012576	Customer order	Partially picked	In Store pick up	01/07/2021 3:00 PM - 3:30 PM	01/07/2021	Jennifer Beach	6	\$1,426.43
<input type="checkbox"/> 012570	Customer order	In processing	In Store pick up	01/06/2021	01/07/2021	Karen Berg	1	\$95.61
<input type="checkbox"/> 012568	Customer order	In processing	In Store pick up	01/06/2021	01/07/2021	Cortoso Retail San Di...		\$11.99
<input type="checkbox"/> 012567	Customer order	In processing	In Store pick up	01/11/2021	01/07/2021	Owen Tolley	2	\$38.23
<input type="checkbox"/> 012566	Customer order	In processing	In Store pick up	01/11/2021	01/07/2021	Mara Gentry	1	\$38.24
<input type="checkbox"/> 012565	Customer order	In processing	In Store pick up	01/11/2021	01/07/2021	Karen Berg	1	\$74.36
<input type="checkbox"/> 012564	Customer order	In processing	In Store pick up	01/06/2021	01/07/2021	Cortoso Retail San Di...		\$15.99
<input type="checkbox"/> 012561	Customer order	In processing	In Store pick up	01/05/2021	01/06/2021	Cortoso Retail San Di...		\$39.99
<input type="checkbox"/> 012553	Customer order	In processing	In Store pick up	02/05/2021	01/06/2021	Karen Berg		\$74.54
<input type="checkbox"/> 012535	Customer order	Partially invoiced	In Store pick up	01/06/2021 3:00 PM - 3:30 PM	01/06/2021	Jennifer Beach	9	\$2,091.38

In Microsoft Dynamics 365 Commerce version 10.0.17 and later, the **Improved user experience for pick up order processing in Point of Sale** feature can be turned on through Feature management in Commerce headquarters. If this feature is turned off, users can't select pickup quantities. By default, the full quantity that was ordered for the line is the quantity that will be picked up. This experience can be problematic, because users might forget to select some items for pickup when they perform the pickup through order fulfillment.

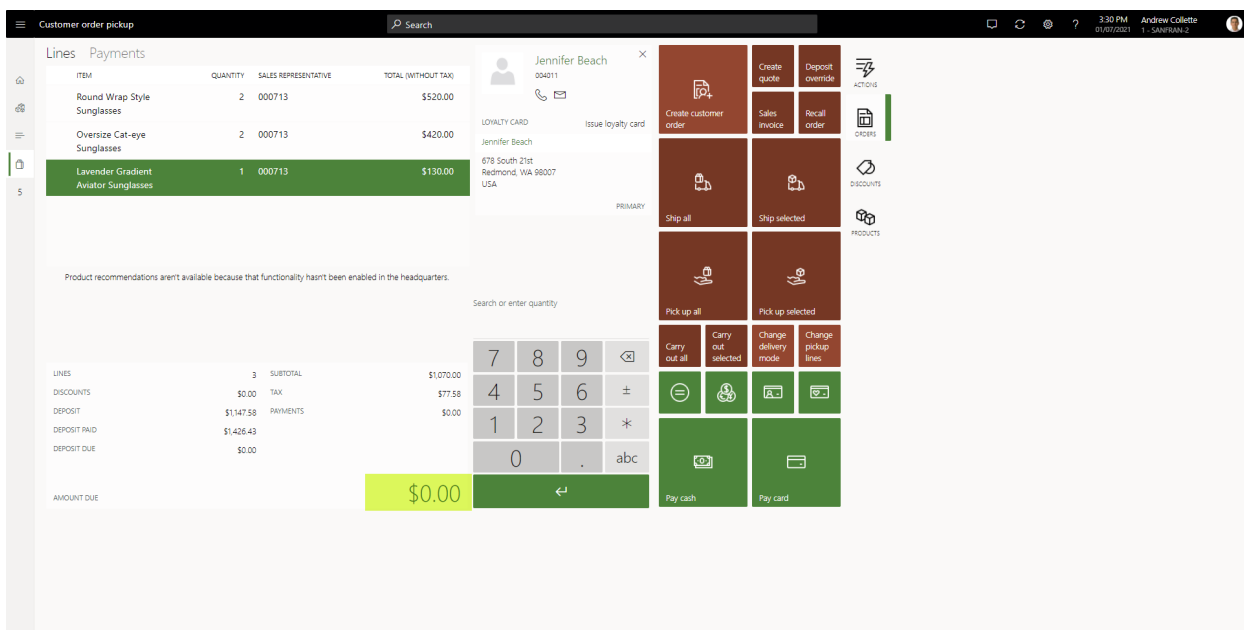
The **Improved user experience for pick up order processing in Point of Sale** feature gives users more control over the selection of products that will be picked up and the quantity of those products that will be picked up. Users don't have to select every line of the sales order on the order fulfillment page before they select **Pick up**. All the items that can be picked up will be shown. Users can specify multiple lines for pickup even if only one product line is selected.

When the **Improve user experience for pick up order processing in Point of Sale** feature is turned on, and you select the **Pick up** operation, the **Pick up** dialog box appears. There, you can select the items and quantities that will be picked up. By default, any ordered quantity that has inventory in a picked or packed state is considered eligible for pickup. By default, that quantity is set as the pickup quantity. You can change the quantity that was entered, provided that the quantity isn't 0 (zero) and doesn't exceed the total open (that is, non-invoiced) quantity for the selected line.



After you select the quantities that will be picked up and then select **Pick up**, the transaction page appears. If the [omni-channel payments](#) feature is turned on, and there are pre-authorized credit card payments on file, you must apply the payment.

On the transaction page, the system calculates the amounts that are due by calculating the total that is due for the selected pickup items and then subtracting any previously applied deposits or authorized credit card payments. You must process payment to complete the pickup transaction. If the [screen layout](#) of the transaction page is configured so that it includes the **Conclude transaction** operation, and no amount is due, you can complete the transaction without selecting a payment method. If the **Conclude transaction** operation isn't available, you can select the **\$0.00 amount due** link in the **Totals** pane to conclude the transaction without having to select a payment method.



## Changing pickup lines or quantities

If you must change the pickup quantity after you've selected the items that will be picked up, you can select **Set quantity**. You can't set the pickup quantity to 0 (zero) or increase it to a value that exceeds the non-invoiced quantity that remains for the ordered line. To remove a pickup line from the transaction cart, select **Void transaction**. The current transaction will be stopped, and the flow for the **Pick up** operation will be restarted.

If the **Improve user experience for pick up order processing in Point of Sale** feature is turned on,

organizations can add a button for the **Change pickup lines** operation to the screen layout of the transaction page. After you create the pickup transaction cart in POS and select items, you can select **Change pickup lines** if you must change the pickup items but don't want to void the whole transaction. In the **Change pickup lines** dialog box that appears, you can change the pickup items and quantities. The transaction cart is then updated to reflect your changes.

The screenshot displays a POS interface for a 'Customer order pickup'. The main screen shows a list of items with columns for ITEM, QUANTITY, SALES REPRESENTATIVE, and TOTAL (WITHOUT TAX). The items listed are:

ITEM	QUANTITY	SALES REPRESENTATIVE	TOTAL (WITHOUT TAX)
Round Wrap Style Sunglasses	2	000713	\$520.00
Oversize Cat-eye Sunglasses	2	000713	\$420.00
Lavender Gradient Aviator Sunglasses	1	000713	\$130.00

The interface also includes a customer profile for Jennifer Beach, a numeric keypad, and a 'Change pickup items' dialog box on the right. The dialog box shows 5 items ready for pickup:

- Round Wrap Style Sunglasses (Ordered: 3 ea (partially picked), Ready for pickup: 2 ea)
- Oversize Cat-eye Sunglasses (Ordered: 2 ea (picked), Ready for pickup: 2 ea)
- Lavender Gradient Aviator Sunglasses (Ordered: 1 ea (picked), Ready for pickup: 1 ea)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create call center orders

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through looking up a customer, creating a new order, searching for a product, and collecting payment from the customer. This procedure uses demo data company USRT and is intended for the Sales Order Clerk. Pre-requisites: The user who completes the procedure is set up as a Call center user and the Fabrikam Semi-Annual Catalog is published with at least one Source code on it.

1. Go to **Retail and Commerce > Customers > Customer service**.
2. For **SearchText**, enter the search criteria to look up the customer.
  - For this example procedure, enter "Karen" and select **Tab**.
3. Select **Search**.
  - Since there is only one customer named "Karen" in demo data, the result will be automatically selected.
4. Select **New sales order**.
5. Expand or collapse the **Sales order** header section.
6. Select the source code for the catalog.
  - If there are no active source codes you can skip this step.
7. Select **Add line**.
8. For **Item number**, enter the item search term.
  - For this sample procedure, enter a partial item number of '8111' and press tab. This action will bring up the item search window.
9. Select the product to add to the sales order.
10. Enter the sales quantity.
11. Select **Create**.
12. Select **Complete** to capture the customer payment.
13. Select **Add**.
  - The Add link is in the Payments tab. Expand the Payments tab if it is collapsed.
14. Select the payment method.
  - For this procedure, select the cash payment method.
15. Close the page.
16. Enter the amount.
  - For this procedure, enter an amount equal to the order balance that can be seen in the Sales order summary page to the left of the amount field. This action will allow you to complete the order as fully paid.
17. Select **OK**.
18. Select **Submit**.

## Additional resources

[Customize transactional emails by mode of delivery](#)

[Change mode of delivery in POS](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Customize transactional emails by mode of delivery

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to set up custom email templates for specific notification types and modes of delivery in Microsoft Dynamics 365 Commerce.

Transactional emails can now be customized for a combination of a notification type (for example, **Order created**, **Order packed**, or **Order invoiced**) and a mode of delivery (for example, overnight, in-store pickup, or curbside pickup). Custom transactional emails let retailers provide their customers order with fulfillment experiences that are tailored to the order's mode of delivery. For example, the "order packed" event can be customized so that it provides curbside pickup instructions for customers who choose curbside pickup. Alternatively, it can provide shipping carrier and delivery information for customers who choose to have their order shipped.

## NOTE

To use the functionality for customized transactional emails, you first must turn on the **Customize transactional email templates by mode of delivery** feature by going to **Workspaces > Feature management** in Commerce headquarters.

Emails can be customized by mode of delivery for the following notification types:

- **Order cancellation** – This email notification type is new.
- **Order created**
- **Order confirmed**
- **Order invoiced** – This email notification type is new. It can be used instead of the **Order shipped** notification type that will send a notification for any invoice event that has a shipped mode of delivery (not a pickup, carry out, or electronic mode of delivery).
- **Order picked**
- **Order packed**
- **Order ready for pickup** – This notification type can be customized by mode of delivery only if the **Support for multiple pickup delivery modes** feature is turned on. In that case, this notification type is functionally equivalent to the **Order packed** notification type.
- **Payment failed**
- **Replacement order created**

## Configure email templates for specific modes of delivery

For this procedure, the assumption is that you've already created your new, custom email templates and added them to the **Organization email templates** page. For information about how to create and upload email templates, see [Create email templates for transactional events](#).

To configure email templates for specific modes of delivery in Commerce headquarters, follow these steps.

1. Go to **Commerce email notification profile**.
2. Under **Retail event notification settings**, select an existing notification type.
3. While the notification type is still selected, select **Configure modes of delivery**.
4. In the **Modes of delivery** dialog box, select **New**.
5. In the new row, in the **Mode of delivery** field, select a mode of delivery.

6. In the **Email ID** field, select the email template to map to the mode of delivery.
7. Select the **Active** check box.
8. Repeat steps 4 through 7 to add more modes of delivery.
9. When you've finished, select **OK**.

#### **NOTE**

- When more than one mode of delivery is present across lines in a sales order, the default template will be used. The default template is the template that is mapped to the notification type on the **Commerce email notification profile** page.
- If a sales order has a mode of delivery that hasn't been configured for a custom email template, the default email template will be used.

## Additional resources

[Create call center orders](#)

[Change mode of delivery in POS](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Change mode of delivery in POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to set up and use the "change mode of delivery" functionality in your point of sale (POS) environment.

In Dynamics 365 Commerce versions 10.0.10 and later, the **Change mode of delivery** operation (647) is available to add to your POS screen layouts.

The change mode of delivery feature provides you with the option to change the mode of delivery for one or more shipment-configured sales lines on the POS transaction. In previous versions of Commerce, you had to go through the full **Ship all** or **Ship selected** configuration flows if you wanted to change the mode of delivery on an existing line that was configured for shipment. This process was time consuming and could result in accidental changes to the delivery origin or delivery dates for the line. The new functionality provides an alternative method for efficiently updating the mode of delivery on these sales lines.

For more information about how to add an operation to a button on your POS button grid, see [Screen layouts for the point of sale](#).

After this feature is configured in POS, when you select **Change mode of delivery**, you will be presented with a list page that allows you to choose the lines of the transaction that you want to change the mode of delivery for. You can choose some or all of the lines, or exit without making any changes. The sales lines that were previously configured for shipment are the only lines in the list that you can change. If you want to change a line designated for pickup or carryout to ship, you must use the **Ship all** or **Ship selected** operations. Conversely, if you want to change a line designated as a shipment to a pickup or carryout, you must use the **Pickup all**, **Pickup selected**, **Carryout all**, or **Carryout selected** operations.

After you select the lines that you want to change, click **Change mode of delivery** to be prompted to select the delivery mode options. If you selected multiple lines to change, POS will only display modes of delivery that have been configured as allowable for all of the selected products. Modes of delivery can be configured to support specific products and delivery addresses. If there is a mode of delivery that is acceptable for one product and address combination but is not acceptable for another selected product and address combination, the mode of delivery is not available. You may need to select lines one by one and change the mode of delivery for each line separately if you want to select a mode of delivery for one product that is not supported by another product.

After you select the new mode of delivery, the transaction page is displayed. To review your new delivery mode selections, select the **Delivery** tab on the transaction list.

## Additional resources

[Create call center orders](#)

[Customize transactional emails by mode of delivery](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Distributed order management (DOM)

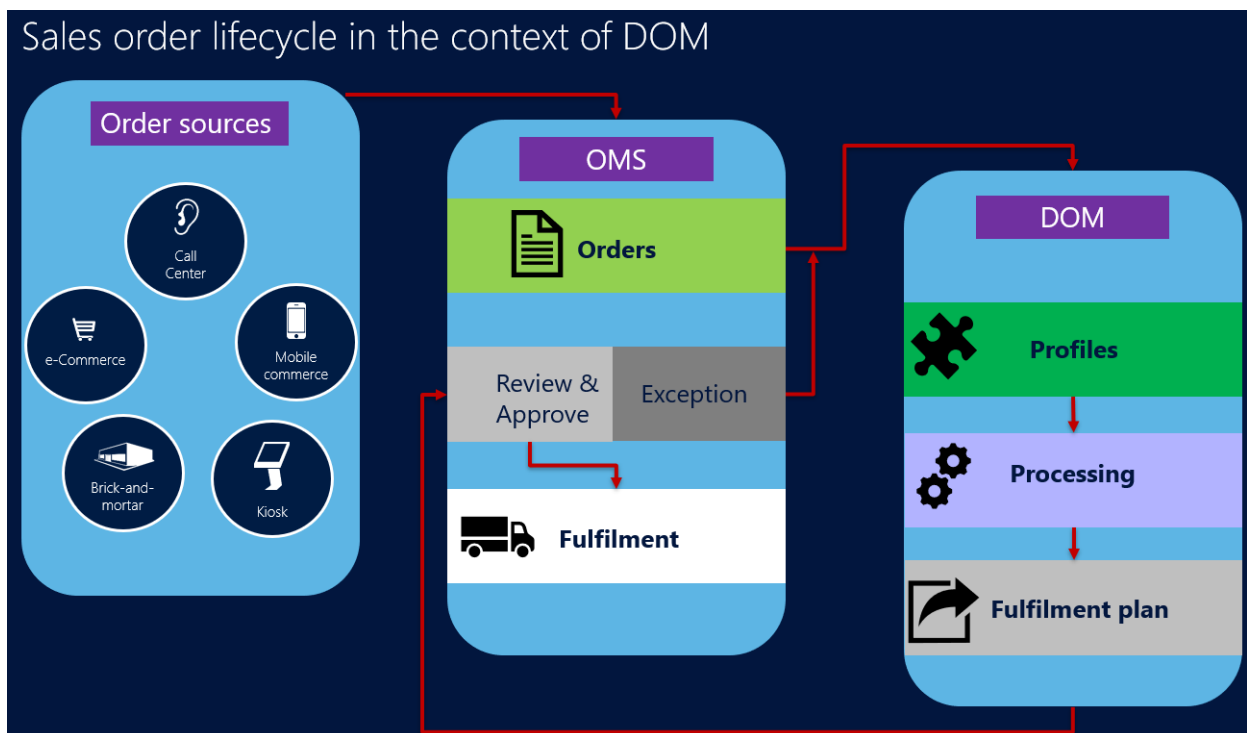
2/18/2021 • 20 minutes to read • [Edit Online](#)

In the new paradigm for commerce operations, retailers strive to provide personalized customer engagement, omni-channel experiences, and frictionless interactions. Because so many choices are available, consumers will shop wherever they can have the most favorable experience. In many cases, prices and products are no longer the top deciding factors for consumers.

To help improve the customer experience, retailers must have visibility into their inventory in real time, across all their channels. A single, holistic view of all the inventory can help optimize order fulfillment, allocation, and distribution. Therefore, adoption and implementation of a distributed order management (DOM) system are becoming more imperative for retailers.

DOM optimizes order fulfillment across a complex network of systems and processes. It relies on a single, global view of inventory across the whole organization to intelligently manage orders, so that they are fulfilled accurately and in a more cost-effective manner. By improving the efficiency of a retailer's supply chain, DOM helps the retailer better meet customer expectations.

The following illustration shows the lifecycle of a sales order in a DOM system.



## Set up DOM

1. Go to **System administration > Setup > License configuration**.
2. On the **Configuration keys** tab, expand the **Commerce** node, and then select the **Distributed Order Management** check box.
3. Go to **Retail and Commerce > Distributed order management > Setup > DOM parameters**.
4. On the **General** tab, set the following values:
  - **Enable distributed order management** – Set this option to **Yes**.

- **Confirm Bing Maps usage for DOM** – Set this option to **Yes**.

**NOTE**

You can set this option to **Yes** only if the **Enable Bing Maps** option on the **Bing Maps** tab of the **Commerce shared parameters** page (**Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**) is also set to **Yes**, and if a valid key is entered in the **Bing Maps key** field.

The [Bing Maps Dev Center](#) portal allows you to restrict access on your Bing Maps API keys to a set of domains that you specify. With this feature, customers can define a strict set of referrer values or IP address ranges that the key will be validated against. Requests originating from your allow list will process normally, while requests from outside of your list will return an access denied response. Adding domain security to your API key is optional and keys left as-is will continue to function. The allow list for a key is independent from all of your other keys, enabling you to have distinct rules for each of your keys. Distributed Order Management does not support the setting up of domain-referred properties.

- **Retention period in days** – Specify how long the fulfillment plans that DOM runs generate are kept in the system. The **DOM fulfillment data deletion job setup** batch job will delete any fulfillment plan that is older than the number of days that you specify here.
- **Rejection period (in days)** – Specify how much time must pass before a rejected order line can be assigned to the same location.

5. On the **Solver** tab, set the following values:

- **Max auto-fulfillment attempts** – Specify how many times the DOM engine will try to broker an order line to a location. If the DOM engine can't broker an order line to a location in the specified number of attempts, it will flag the order line as an exception. It will then skip that line in future runs until the status is manually reset.
- **Local store region radius** – Enter a value. This field helps determine how locations are grouped and considered equal in terms of distance. For example, if you enter **100**, every store or distribution center within a 100-mile radius of the fulfillment address is considered equal in terms of distance.
- **Solver type** – Select a value. Two solver types are released with Commerce: **Production Solver** and **Simplified Solver**. For all machines that will run DOM (that is, all servers that are part of the DOMBatch group), **Production Solver** must be selected. The Production Solver requires the special license key that, by default, is licensed and deployed in production environments. For non-production environments, this license key must be manually deployed. To manually deploy the license key, follow these steps:
  - a. In Microsoft Dynamics Lifecycle Services, open the Shared asset library, select **Model** as the asset type, and download the **DOM license** file.
  - b. Start Microsoft Internet Information Services (IIS) Manager, right-click **AOSService website**, and then select **Explore**. A Windows Explorer window is opened at **<AOS service root>\webroot**. Make a note of the **<AOS Service root>** path, because you will use it in the next step.
  - c. Copy the configuration file in the **<AOS Service root>\PackagesLocalDirectory\DOM\bin** directory.
  - d. Go to the Headquarters client, and open the **DOM parameters** page. On the **Solver** tab, in the **Solver type** field, select **Production solver**, and confirm that no error messages appear.

#### NOTE

The Simplified Solver is provided so that retailers can try out the DOM feature without having to deploy the special license. Organizations should not use the Simplified Solver in production environments.

The Production Solver improves performance (such as the number of orders and order lines that can be handled in a run) and convergence of results (since a batch of orders might not yield the best result in some scenarios). Some rules like the **Partial orders** rule and the **Maximum number of locations** rule require Production Solver.

6. Go back to **Retail and Commerce > Distributed order management > Setup > DOM parameters**.

7. On the **Number sequences** tab, assign the required number sequences to the various DOM entities.

#### NOTE

Before the number sequences can be assigned to the entities, they must be defined on the **Number sequences** page (**Organization administration > Number sequences > Number sequences**).

8. The DOM feature supports the definition of various types of DOM rules, and organizations can configure multiple rules, depending on their business needs. DOM rules can be defined for a group of locations or individual locations, and for a specific product category, product, or variant. To create the grouping of locations that must be used for the DOM rules, follow these steps:

- a. Go to **Retail and Commerce > Channel setup > Fulfillment groups**.
- b. Select **New**, and enter a name and description for the new group.
- c. Select **Save**.
- d. Select **Add line** to add a single location to the group. Alternatively, select **Add lines** to add multiple locations.

#### NOTE

In Commerce version 10.0.12 and higher, **Ability to specify locations as 'Shipping' or 'Pickup' enabled within Fulfillment group** must be enabled in the **Feature Management** workspace.

This feature add new configurations on the **Fulfillment group** page so you can define if the warehouse can be used for shipping or if the warehouse/store combination can be used for shipping, pickup, or both.

If you enable the feature, the options available for location selection when you create pickup or shipment orders in POS will be updated.

Enabling the feature also results in updated pages in POS when the "ship all" or "ship selected" operations are selected.

9. To define rules, go to **Retail and Commerce > Distributed order management > Setup > Manage rules**. The following DOM rules are currently supported:

- **Minimum inventory rule** – This rule type lets organizations "ring fence" a specific quantity of a product for purposes other than order fulfillment. For example, organizations might not want DOM to consider all the inventory that is available in a store for order fulfillment. Instead, they might want to reserve some inventory for walk-in customers. When this rule type is used, you can define the minimum inventory to keep for a category of products, an individual product, or a product variant per location or group of locations.
- **Fulfillment location priority rule** – This rule type lets organizations define a hierarchy of

locations to establish the priority that the DOM engine considers when it tries to identify fulfillment locations for specific products. The valid range of priorities is 1 through 10, where 1 is the highest priority and 10 is the lowest priority. Locations that have higher priority are considered before locations that have lower priority. If the rule is defined as a hard constraint rule, orders are brokered only to locations that priorities are defined for.

- **Partial orders rule** – This rule lets organizations define whether an order or order lines can be partially fulfilled. The following parameters are available:
  - **Fulfill partial orders?** – If this option is set to **Yes**, DOM can fulfill only part of the quantity on an order line. This partial fulfillment is achieved by splitting the order line.
  - **Fulfill partial lines?** – If this option is set to **Yes**, DOM can fulfill a partial quantity of order lines. This partial fulfillment is achieved by splitting the order line.
  - **Fulfill order from one location only** – If this option is set to **Yes**, DOM makes sure that all lines on an order are fulfilled from a single location.

The following table explains the behavior when a combination of these parameters is defined.

COMBINATION NUMBER	FULFILL PARTIAL ORDERS	FULFILL PARTIAL LINES	FULFILL ORDER FROM ONE LOCATION ONLY	DESCRIPTION
1	Yes	Yes	Yes	A few lines of the order can be fulfilled, and individual lines can be partially fulfilled, but all the lines must be from the same location in an instance of the DOM run. (This combination isn't currently supported.)
2	Yes	No	Yes	A few lines of the order can be fulfilled, but individual lines can't be partially fulfilled, and all the fulfilled lines must be from the same location in an instance of the DOM run. (This combination isn't currently supported.)

COMBINATION NUMBER	FULFILL PARTIAL ORDERS	FULFILL PARTIAL LINES	FULFILL ORDER FROM ONE LOCATION ONLY	DESCRIPTION
3	Yes	Yes	No	A few lines of the order can be fulfilled, individual lines can be partially fulfilled, and each line can be fulfilled from more than one location in an instance of the DOM run.
4*	No	Not applicable	No	All order lines must be fulfilled, individual lines can't be partially fulfilled, and each order line can be fulfilled from a different location.
5*	No	Not applicable	Yes	All order lines must be fulfilled, individual lines can't be partially fulfilled, and all the order lines can be delivered from one location only.
6*	No	Not applicable	No	This combination works like combination 4, because <b>Fulfill partial lines</b> can't be set to <b>Yes</b> when <b>Fulfill partial orders</b> is set to <b>No</b> .
7*	No	Not applicable	Yes	This combination works like combination 5, because <b>Fulfill partial lines</b> can't be <b>Yes</b> when <b>Fulfill partial orders</b> is <b>No</b> .



COMBINATION NUMBER	FULFILL PARTIAL ORDERS	FULFILL PARTIAL LINES	FULFILL ORDER FROM ONE LOCATION ONLY	DESCRIPTION
8	Yes	No	No	A few lines of the order can be fulfilled, but individual lines can't be partially fulfilled, and the various order lines can be fulfilled from more than one location in an instance of the DOM run.
9*	No	Not applicable	Yes	All order lines must be fulfilled, and all the order lines must be fulfilled from one location only.

\* If **Fulfill partial orders** is set to **No**, **Fulfill partial lines** is always considered to be set to **No**, regardless of how it's actually set.

**NOTE**

In Retail version 10.0.5, the parameter **Fulfill order from one location only** was changed to **Maximum fulfilling locations**. Instead of allowing a user to configure whether orders can be fulfilled from one location only or fulfilled from as many locations as it can be, users can now specify whether the fulfillment can be from a definite set of locations (up to 5) or from as many locations as it can be. This provides more flexibility in terms of how many locations the order can be fulfilled from. This rule only works with Production Solver.

- **Offline fulfillment location rule** – This rule lets organizations specify a location or group of locations as offline or unavailable to DOM, so that orders can't be assigned to those locations for fulfillment.
- **Maximum rejects rule** – This rule lets organizations define a threshold for rejections. When the threshold is reached, the DOM processor will mark an order or order line as an exception, and exclude it from further processing.

After order lines are assigned to a location, the location can reject an assigned order line, because it might not be able to fulfill that line for some reasons. Rejected lines are marked as an exception and put back into the pool for processing in the next run. During the next run, DOM will try to assign the rejected line to a different location. The new location can also reject the assigned order line. This cycle of assignment and rejection can occur multiple times. When the rejection count hits the threshold that is defined, DOM will mark the order line as a permanent exception and won't pick the line for assignment again. DOM will consider the order line again for reassignment only if a user manually resets the status of the order line.

- **Maximum distance rule** – This rule lets organizations define the maximum distance that a location or group of locations can be to fulfill the order. If overlapping maximum distance rules are defined for a location, DOM will apply the lowest maximum distance that is defined for that location.

- **Maximum orders rule** – This rule lets organizations define the maximum number of orders that a location or group of locations can process during a calendar day. If the maximum number of orders is assigned to a location in a single day, DOM won't assign any more orders to that location for the rest of that calendar day.

Here are some of the common attributes that can be defined for all the preceding rule types:

- **Start date and End date** – Every rule can be made date-effective by using the these fields.
- **Disabled** – Only rules that have a value of **No** for this field are considered in a DOM run.
- **Hard constraint** – A rule can be defined as either a hard constraint or not a hard constraint. Every DOM run goes through two iterations. In the first iteration, every rule is treated as a hard constraint rule, regardless of the setting of this field. In other words, every rule is applied. The only exception is the **Location priority** rule. In the second iteration, the rules that weren't defined as hard constraint rules are removed, and the order or order lines that weren't assigned to locations when all the rules were applied are assigned to locations.

10. Fulfillment profiles are used to group a collection of rules, legal entities, sales order origins, and modes of delivery. Every DOM run is for a specific fulfillment profile. In this way, organizations can define and run a set of rules for a set of legal entities, on orders that have specific sales order origins and modes of delivery. Therefore, if different set of rules must be run for different sets of sales order origins or modes of delivery, the fulfillment profiles can be defined accordingly. To set up fulfillment profiles, follow these steps:

- Go to **Retail and Commerce > Distributed order management > Setup > Fulfillment profiles**.
- Select **New**.
- Enter values in the **Profile** and **Description** fields.
- Set the **Auto apply result** option. If you set this option to **Yes**, the results of the DOM run for the profile will be automatically applied to the sales order lines. If you set it to **No**, the results can only be viewed in the fulfillment plan. They won't be applied to the sales order lines.
- If you want the DOM profile to be run for orders that have every sales order origin, including orders where the sales order origin is undefined, set the **Process orders with empty sales origin** option to **Yes**. To run the profile for only a few sales order origins, you can define them on the **Sales origins** page, as explained later.

#### NOTE

In Commerce version 10.0.12 and higher, **Ability to assign Fulfillment group to a Fulfillment Profile** must be enabled in the **Feature Management** workspace.

This feature adds a new configuration on the **Fulfillment profile** page that can be associated to a single fulfillment group.

If you select the fulfillment group, the DOM rules for that fulfillment profile will efficiently run against the "shipping" warehouses included in the fulfillment group.

To effectively use this feature, ensure that there is one fulfillment group that contains all the shipping warehouses, and then associate that fulfillment group to the fulfillment profile.

- On the **Legal entities** FastTab, select **Add**, and then select a legal entity.
- On the **Rules** FastTab, select **Add**, and then select the rule to link to the profile.
- Repeat the previous two steps until all the required rules are associated with the profile.
- Select **Save**.
- On the Action Pane, on the **Setup** tab, select **Modes of delivery**.
- On the **Modes of delivery** page, select **New**.

- l. In the **Company** field, select the legal entity. The list of companies is limited to the legal entities that you added earlier.
- m. In the **Mode of delivery** field, select the mode of delivery to associate with this profile. A mode of delivery can't be associated with multiple active profiles.
- n. Repeat the previous two steps until all the required modes of delivery are associated with the profile.
- o. Close the **Modes of delivery** page.
- p. On the Action Pane, on the **Setup** tab, select **Sales order origins**.
- q. On the **Sales origins** page, select **New**.
- r. In the **Company** field, select the legal entity. The list of companies is limited to the legal entities that you added earlier.
- s. In the **Sales origin** field, select the sales origin to associate with this profile. A sales origin can't be associated with multiple active profiles.
- t. Repeat the previous two steps until all the required sales origins are associated with the profile.
- u. Close the **Sales origins** page.
- v. Set the **Enable profile** option to **Yes**. If there are any errors in the setup, you receive a warning message.

## DOM processing

DOM will run only in a batch job. To configure the batch job for DOM runs, follow these steps.

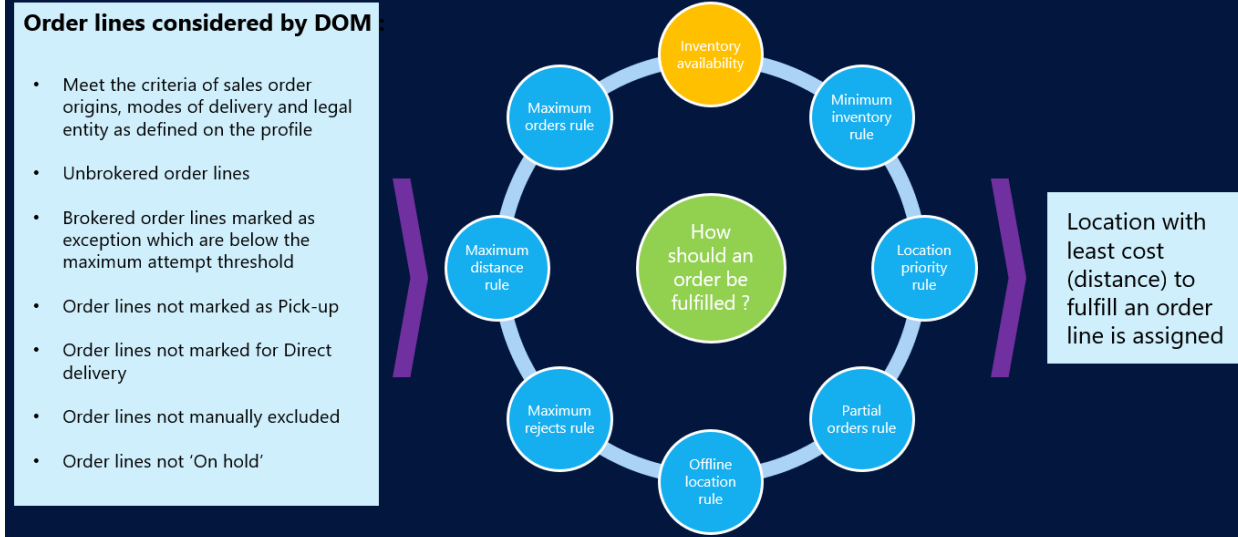
1. Go to **Retail and Commerce > Distributed order management > Batch processing > DOM processor job setup**.
2. On the **Parameters** FastTab, in the **Fulfillment profile** field, select a profile that DOM must be run for.
3. On the **Run in the background** FastTab, in the **Batch group** field, select a configured batch group.
4. In the **Task description** field, enter a name for the batch job.
5. Select **Recurrence**, and define the recurrence of the batch job.
6. Select **OK**.

At the time of processing, DOM will consider the order and order lines as described here:

- Order lines that meet the criteria for sales order origins, modes of delivery, and legal entity as defined in the DOM profile, and that also meet any of these criteria:
  - They are created from commerce channels.
  - They have never been brokered by DOM.
  - They have been brokered by DOM before, but they are marked as exceptions and are below the maximum threshold for attempts.
  - The mode of delivery isn't pick-up or electronic delivery.
  - They aren't marked for delivery.
  - They aren't manually excluded.
- Orders that aren't on hold

After it applies the rules, inventory constraints, and optimization, DOM picks the location that is closest to the customer's delivery address.

## Distributed order management – Order brokering



## Results of DOM runs

If the fulfillment profile is set to **Auto apply**, the results of the run will be automatically applied to the sales order lines, and the fulfillment plan can be viewed separately. However, if the fulfillment profile isn't set to **Auto apply**, the results of the run can be seen only from the fulfillment plan view.

To view all the fulfillment plans that are generated, follow these steps.

1. Go to **Retail and Commerce > Distributed order management > Distributed order management**.
2. In the **Distributed order management** workspace, select the **Fulfillment Plans** tile.
3. Select the ID of the relevant order fulfillment plan to view the fulfillment plan.

The order details section of the fulfillment plan shows the original sales order lines that were part of the run. Besides the standard sales order line fields, the order details section also includes the following three DOM-related fields:

- **Fulfillment type** – This field indicates whether the sales order line is fully brokered, partially brokered, or not brokered at all to a location.
- **Split** – This field indicates whether one sales order line has been split and brokered to different locations.
- **Number of fulfillment locations** – This field indicates how many fulfillment lines were created for one sales order line (based on the number of locations that the original sales order line was brokered to).

The order fulfillment details section shows the assignment of the original sales order lines to different locations, together with their quantities.

## Order line actions and statuses

The following describes settings on the order line. To open the order line, go to **Retail and Commerce > Customers > All sales orders**.

- If you set the **Exclude from DOM processing** option on the **General** tab of the sales order line to **Yes**, the order or order line will be excluded from DOM processing.
- The **DOM status** field on the **General** tab of the sales order line can be set to one of the following

values:

- **None** – The order line has never been brokered.
- **Complete** – The order line has been successfully brokered and assigned to a location.
- **Exception** – The order line has been brokered but can't be assigned to a location. Exceptions have multiple subtypes that can be viewed from the DOM workspace:
  - **No quantity available** – There is no available inventory to assign the order to in the locations.
  - **Maximum rejections** – The order line has reached the maximum threshold for rejections.
  - **Data modification conflict** – The sales order line has been changed since the order was brokered. Therefore, the fulfillment plan can't be applied to the order.
  - **Order line specific exception** – There are multiple exceptions on the order line.
- During the sales order entry, DOM can be run in an interactive mode. While you're entering an order line, after you specify the product and quantity, you can select **Update line** and then, under **DOM**, select **Suggest fulfillment location**. You then see a list of locations that is based on DOM rules that can fulfill the quantity on the order line. This list is sorted by distance. Select a location to set the relevant site and warehouse on the sales order line. For this functionality to work, there must be an existing, active fulfillment profile that matches the sales origin and delivery mode on the sales line.
- To view the DOM run logs for a sales order line, select **Sales order line**, and then, under **DOM**, select **View DOM Logs**. The DOM logs show all the events and logs that were generated by the DOM run. The logs can help you understand why a specific location was assigned to the order line, and what rules and constraints were considered as a part of the assignment. On the **Manage** tab, the DOM logs are also available at the level of the sales order header.

## Run a clean-up job for DOM fulfillment plans

As DOM processing is run, fulfillment plans are created. Over time, the system will keep numerous fulfillment plans. To manage the number of fulfillment plans that the system keeps, you can configure a batch job that deletes older fulfillment plans, based on the **Retention period in days** value.

1. Go to **Retail and Commerce > Distributed order management > Batch processing > DOM fulfillment data deletion job setup**.
2. In the **Batch group** field, select a configured batch group.
3. Select **Recurrence**, and define the recurrence of the batch job.
4. Select **OK**.

## More information

Here are some things to consider when you use the DOM feature:

- Currently, DOM looks only at orders that are created from commerce channels. Sales orders are identified as sales orders when the **Commerce sale** option is set to **Yes**.
- Microsoft hasn't tested DOM with advanced warehouse management features. Customers and partners must be careful to determine whether DOM is compatible with the advanced warehouse management capabilities and processes that are relevant to them.
- DOM is available only on the cloud version of Commerce. It isn't supported in on-premises deployments.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cost configuration for distributed order management (DOM)

2/18/2021 • 7 minutes to read • [Edit Online](#)

Organizations consider multiple cost components to determine the optimal location to fulfill an order from. Some of these cost components are shipping cost, handling cost, and packaging cost. A combination of these costs is calculated to determine the fulfillment location.

When the first iteration of distributed order management (DOM) in Dynamics 365 Commerce optimized the assignment of orders to fulfillment locations, it factored in distance only. Although distance can be correlated with cost, it isn't the same as cost. For example, an overnight shipping method costs more than three-day shipping or seven-day shipping over the same distance.

The cost configuration feature lets retailers define and configure additional cost components that will be calculated and factored in to determine the optimal location to fulfill order lines from.

When cost components are configured, the DOM solver uses only those cost definitions to determine the optimal location for order fulfillment. It doesn't consider the distance component as a cost. However, if no cost components are configured, the DOM solver does use the distance component as a cost to determine the optimal location for order fulfillment.

## Set up cost components

Two major cost component types can be defined in the system: **Shipping** and **Other**.

Both cost component types support multiple calculation bases, as shown in the following table.

COST COMPONENT TYPE	CALCULATION BASIS
Shipping	<ul style="list-style-type: none"><li>• Simple</li><li>• Tiered</li></ul>
Other	<ul style="list-style-type: none"><li>• Sales order</li><li>• Sales line</li><li>• Location</li></ul>

### Shipping cost component type

This section explains how to set up each combination of the **Shipping** cost component type and a calculation basis for shipping costs. It also explains how the DOM solver uses each combination.

#### Cost component type = Shipping and Calculation basis = Simple

If a combination of the **Shipping** cost component type and the **Simple** calculation basis is used, the shipping cost for a mode of delivery is based on either a flat cost or distance.

You must set up the following fields for this combination:

- **Cost factor** – Enter a unique identifier for the cost factor.
- **Description** – Enter the name and description of the cost factor.
- **Start date** and **End date** – You can use these fields to limit the cost factor for a specific date range. If you

leave these fields blank, the cost factor is valid for an indefinite period.

- **Active** – Indicate whether the cost factor is active. The DOM considers only active cost factors that are associated with the fulfillment profile.
- **Company** – Specify the legal entity that the cost factor is configured for. All lines of the calculation criteria must be for the same legal entity.
- **Modes of delivery** – Specify the modes of delivery that the cost is configured for.
- **Calculation type** – Specify how the cost should be calculated for a specific mode of delivery. Two calculation types are supported:
  - **Fixed** – A flat cost is used for the mode of delivery. If you select this calculation type, the **Cost** field defines the flat cost.
  - **Per-distance unit** – The cost for the mode of delivery is calculated as the cost value that is specified in the **Cost** field times the distance between the delivery address and the locations.
- **Cost** – Specify the cost value that is used in conjunction with the **Calculation type** field to compute the cost for a mode of delivery.

**Cost component type = Shipping and Calculation basis = Tiered**

If a combination of the **Shipping** cost component type and the **Tiered** calculation basis is used, the shipping cost for a mode of delivery is based on either a flat cost or distance. However, in this combination, the distance is based on a tiered range of distances.

You must set up the following fields for this combination:

- **Cost factor** – Enter a unique identifier for the cost factor.
- **Description** – Enter the name and description of the cost factor.
- **Default cost** – Specify the cost that should be used for a mode of delivery if the distance between the delivery address and the location doesn't fall into any of the tiered distances for the mode of delivery.
- **Start date** and **End date** – You can use these fields to limit the cost factor for a specific date range. If you leave these fields blank, the cost factor is valid for an indefinite period.
- **Active** – Indicate whether the cost factor is active. The DOM considers only active cost factors that are associated with the fulfillment profile.
- **Company** – Specify the legal entity that the cost factor is configured for. All lines of the calculation criteria must be for the same legal entity.
- **Modes of delivery** – Specify the modes of delivery that the cost is configured for.
- **Distance type** – Specify whether the tiered distance definition is an aerial distance or a road distance.
- **Distance units** – Specify the unit that the tiered distance is measured in.
- **Distance from** – Specify the start range for the tiered distance.
- **Distance to** – Specify the end range for the tiered distance.
- **Calculation type** – Specify how the cost should be calculated for a specific mode of delivery and tiered distance. Two calculation types are supported:
  - **Fixed** – A flat cost is used for the mode of delivery. If you select this calculation type, the **Cost** field defines the flat cost.
  - **Per distance unit** – The cost for the mode of delivery and tiered distance is calculated as the cost value that is specified in the **Cost** field times the distance between the delivery address and the



locations.

- **Cost** – Specify the cost value that is used in conjunction with the **Calculation type** field to compute the cost for a mode of delivery.

#### NOTE

- When you define tiered distances, the system validates that there are no missing or overlapping distances.
- The distance type that is used for a mode of delivery must be the same across all the tiered distances.

### Other cost component type

This section explains how to set up each combination of the **Other** cost component type and an other cost type for non-shipping costs. It also explains how the DOM solver uses each combination.

#### Cost component type = Other and Other cost type = Sales order

A combination of the **Other** cost component type and the **Sales order** other cost type is used to define non-shipping costs at the sales order level.

You must set up the following fields for this combination:

- **Cost factor** – Enter a unique identifier for the cost factor.
- **Description** – Enter the name and description of the cost factor.
- **Start date** and **End date** – You can use these fields to limit the cost factor for a specific date range. If you leave these fields blank, the cost factor is valid for an indefinite period.
- **Active** – Indicate whether the cost factor is active. The DOM considers only active cost factors that are associated with the fulfillment profile.
- **Cost** – Specify the cost value for a non-shipping cost at the sales order level.

#### Cost component type = Other and Other cost type = Sales line

A combination of the **Other** cost component type and the **Sales line** other cost type is used to define non-shipping costs at the sales order line level.

You must set up the following fields for this combination:

- **Cost factor** – Enter a unique identifier for the cost factor.
- **Description** – Enter the name and description of the cost factor.
- **Start date** and **End date** – You can use these fields to limit the cost factor for a specific date range. If you leave these fields blank, the cost factor is valid for an indefinite period.
- **Active** – Indicate whether the cost factor is active. The DOM considers only active cost factors that are associated with the fulfillment profile.
- **Cost** – Specify the cost value for a non-shipping cost at the sales order line level.

#### Cost component type = Other and Other cost type = Location

A combination of the **Other** cost component type and the **Location** other cost type is used to define non-shipping costs for a group of locations or an individual location.

You must set up the following fields for this combination:

- **Cost factor** – Enter a unique identifier for the cost factor.
- **Description** – Enter the name and description of the cost factor.
- **Start date** and **End date** – You can use these fields to limit the cost factor for a specific date range. If you leave these fields blank, the cost factor is valid for an indefinite period.
- **Active** – Indicate whether the cost factor is active. The DOM considers only active cost factors that are associated with the fulfillment profile.

- **Fulfillment group** – Specify the group of locations that the non-shipping cost is defined for.
- **Fulfillment location** – Specify the location that the non-shipping cost is defined for.

**NOTE**

You can't specify a fulfillment group and a fulfillment location on the same line for location-based calculation criteria.

- **Cost** – Specify the cost value for a non-shipping cost at the fulfillment group level or fulfillment location level.

**IMPORTANT**

For DOM to consider these costs when it's run, you must add the cost factor to the relevant fulfillment profile.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Store order fulfillment

2/18/2021 • 14 minutes to read • [Edit Online](#)

Many retailers would like to optimize order fulfillment by enabling stores to fill orders. Order fulfillment at the store level can help to ease overstock scenarios for a specific store, or may be needed from a logistical standpoint in cases where a store has extra capacity or is located within closer shipping distance to the customer. To address this need, a unified order fulfillment operation is available at the point of sale.

Orders for fulfillment at a specific store have the store's warehouse designated on the header or lines of the order.

The order fulfillment operation in the point of sale provides a single work area in the point of sale that can be used to process orders. This includes everything from accepting the order, to marking it as shipped, or initiating store pickup.

## Access unified order fulfillment in the point of sale

Order fulfillment, [Operation ID 928](#), can be used to access the store order fulfillment work area in the point of sale.

The order fulfillment operation does not have its own permission out-of-the-box, but in the future, users will be able to use the **Allow retrieve order** permission to invoke the operation from the point of sale.

At the store level, a configuration setting is available to determine whether an order line must be accepted manually from within the point of sale. If that configuration option is not set, order lines will be accepted by default. If that configuration option is turned on, users at the point of sale will need to select **Allow accept order** permission to accept orders from within the point of sale.

Order lines can also be rejected from the point of sale. Rejecting an order line signifies that it will not be fulfilled at that store and sends the order line back for reassignment to another store or warehouse. Order line rejection permission is granted through the **Allow order reject** permission.

## Order fulfillment operation parameters

Order fulfillment provides out-of-the-box parameters that can be applied to the operation when it is called in the point of sale. When the **All orders** parameter is configured, all orders are shown when the operation is used. The **Orders to ship** parameter shows only orders that must be shipped from the store and **Orders for pick up** shows orders that will be picked up in-store.

## Orders for fulfillment

The order fulfillment operation shows only orders that will either be picked up in or shipped from the current store. Orders for other stores to fulfill are not listed when using the order fulfillment operation.

## Line selection

Lines can be selected using the **Select** function in the Action Pane. When **Select** is enabled, multiple lines can be selected for processing. You can clear selected lines by clicking the same line again.

## Line details

Line details can be shown using the line details flyout menu. When this menu is used, three tabs are provided to

show additional information for the selected line. The first tab, **Line details** shows details for the line itself such as quantity ordered and remaining. Additional details are provided including quantity picked, packed, and invoiced as well as mode of delivery and delivery address. The **Order details** tab provides order header information including customer, customer ID, order number, order total, and balance. The **Inventory** tab shows information for the selected line in terms of physical available inventory, reserved inventory and ordered inventory.

If multiple lines are selected, the order line details flyout menu will only indicate that multiple lines are selected. To show details for a single line, clear the lines until only one line remains.

## Pending order lines

Unified order fulfillment includes the ability to manually accept orders. By default, orders for fulfillment at the store are already accepted. However, if business processes dictate that a worker at the store level must accept orders, manual acceptance can be turned on at the retail store level. To enable order acceptance, go to **Retail and Commerce > Channels > Stores > All stores**. Open the desired store and on the **General** tab, locate the **Order fulfillment** sub header. This sub header has a **Manual accept** option that is set to **No** by default. By setting this option to **Yes** and synchronizing the changes to the channel database, order lines can go through the acceptance process.

Workers with the **Allow accept order** permission can open order fulfillment and select lines for acceptance. Once lines have been accepted, their state changes from **Pending** to **Accepted** and the rest of the order fulfillment process can proceed. When **Manual accept** is turned on, orders will not be processed until they have been accepted.

Orders for store pickup never have the **Pending** state. This is done to avoid a scenario in which a customer arrives at the store and the order line cannot be processed because a worker with the proper privilege is not available.

## Accepted order lines

Orders with the line state **Accepted** can proceed through the rest of the order fulfillment process at the point of sale. After an order has been accepted, any remaining action can be taken against the order line.

For example, an accepted order line can be selected and then picked up directly without going through picking and packing.

## Line actions

### Pick

The **Pick** category of actions is provided to aid in the process of picking order lines from shelves. The picking action can only be performed on order lines that have been previously accepted.

#### Action: Picking

- **Resulting POS status:** Picking
- **Resulting back office status:** No change

After an order has been accepted, lines can be selected and marked as **Picking**. Marking a line as **Picking** is a way to indicate that the picking work is already being performed on a line. This prevents two workers from attempting to pick the same order lines at the same time.

#### Action: Print picking list

- **Resulting status:** Picking
- **Resulting back office status:** No change

Picking lists can be printed at the point of sale to assist workers performing the picking process. A printed picking list can be carried with the worker performing picking and as products are picked, the worker would manually mark them as picked on the picking list.

The picking list format is configured in Commerce and added to the receipt profile. For more information about setting up receipt profiles, see [Receipt templates and printing](#).

If lines are selected and a picking list is printed for those lines, they are automatically updated with the **Picking** status.

#### **Action: Mark as picked**

- **Resulting status:** Picked or partially picked
- **Resulting back office status:** Picked or partially picked

After the physical picking process has been performed, lines can be marked as **Picked**. Selecting a line and marking it as **Picked** performs a real-time call to update the order line. After the line has been marked as **Picked** at the point of sale, the status in the back office is also updated to **Picked** and inventory transactions reflect that the specified quantity has been decremented.

When orders are processed over time, partial quantities can be processed for a specific line. If a line is selected and the action **Mark as picked** is taken, and the quantity is greater than one, the user is prompted for the quantity. The remaining quantity to be picked is auto-filled. If less than the remaining quantity is specified, the status of the line becomes **Partially picked**. When the order line is updated in the back office, it will also reflect the partially picked status and the quantity entered by the user is used for the inventory update.

If an order line is picked in error, the unpick process must be performed on the order line in the back office. There is currently no unpick action supported at the point of sale.

Orders lines from different orders can be selected and marked as **Picking**, printed on the same pick list, or marked as **Picked**.

#### **Pack**

Order lines can be packed at any point after the order line has been accepted.

#### **Action: Print packing slip**

- **Resulting status:** Packed or partially packed
- **Resulting back office status:** Delivered or partially delivered

This action marks lines as packed or partially packed and prints a packing slip. A packing slip can be printed to validate the products that have been packed together. The packing slip format is configured in Commerce and added to the receipt profile. For more information about setting up receipt profiles, see [Receipt templates and printing](#).

#### **Action: Mark as packed**

- **Resulting status:** Packed or partially packed
- **Resulting back office status:** Delivered or partially delivered

The **Mark as packed** action can be used to indicate that lines are packed without printing a packing slip. Both **Print packing slip** and **Mark as packed** result in inventory transactions in the back office. Packing lines in the point of sale will result in packing slip journals being generated in the back office.

If an order line is packed in error, the packing slip journal must be corrected at the back office.

Only lines on the same order and with the same mode of delivery can be packed at the same time.

Currently, the option to mark store pickup lines as **Packed** is disabled. This capability will be added in a future

release. The packing slip creation process will be enhanced to support injection of third-party shipping information into the packing slip process.

### **Pick up**

Orders for store pickup can be directly picked up once they are retrieved in the point of sale. Store pick up orders are not subject to acceptance.

#### **Action: Pick up**

- **Resulting status:** Invoiced or partially invoiced
- **Resulting back office status:** Invoiced or partially invoiced

If a line is selected for pick up from unified order fulfillment, the entire order is loaded into the point of sale and the full quantity for the selected line is marked. Other lines on the order are also loaded into the transaction view of the point of sale, but with quantity marked as zero.

After lines for pickup have been loaded into the transaction view, the transaction can be tendered as usual.

Lines that have been fully invoiced through pick up will no longer show up in unified order processing. Lines that have been partially picked up will continue to appear in unified order fulfillment until they have been picked up in full.

If a line is picked up in error, a return must be performed to correct the error.

Only lines on the same order and with the same mode of delivery can be picked up at the same time.

### **Shipping**

Order lines to be shipped from the store can be processed through unified order fulfillment using the **Ship** action. If manual order line acceptance is configured at the channel level, orders must be accepted prior to shipping. After an order line has been accepted and has a **Pending** or other status, lines can be shipped.

#### **Action: Ship**

- **Resulting status:** Invoiced or partially invoiced
- **Resulting back office status:** Invoiced or partially invoiced

Lines shipped from unified order fulfillment are invoiced from the back office similar to if the order is invoiced directly from the back office. Lines being shipped from unified order fulfillment are not loaded into the transaction view and there is no tendering performed at the time the lines are shipped.

Order lines that have been fully shipped no longer appear in unified order fulfillment. Partially shipped lines will continue to appear in unified order fulfillment until they have been shipped in full.

Only lines from the same order can be shipped at the same time. If the lines from the same order have different modes of deliver, they can still be selected for shipping at the same time.

### **Reject**

Lines or partial lines can be rejected. This allows them to be reassigned from the back office to another store or warehouse. Lines can only be rejected if they have not yet been picked or packed. To reject a line that has already been picked or packed, that line must be unpicked or unpacked from the back office.

#### **Action: Reject**

- **Resulting status:** Rejected
- **Resulting back office status:** No change

The rejected order lines can be viewed from the **Sales order processing and inquiry** workspace. Clear the person filter on the workspace to view all the rejected order lines across the stores. The **Rejected order lines** tab under the **Orders and favorites** section display the order line details. Additionally, the users can click the

**Rejected order lines** button under the **Summary** section to navigate to a sales order view. This shows all the orders that have one or more rejected order lines. If Distributed Order Management (DOM) is enabled, then these rejected orders will be automatically reassigned to the appropriate stores for fulfillment, however, these order lines can be manually reassigned as well. To do so, select the line that shows the **Fulfillment status as Rejected** and change the site/warehouse as needed. Click the **Update line** drop-down menu and click **Reset fulfillment status** to change the fulfillment status from **Rejected** to **Accepted** or **Pending** depending on the order fulfillment set up. After the fulfillment status is reset, then the store workers will be able to view the order lines in POS.

## Line quantity tracking

A single order line of quantity greater than one can be processed over time, resulting in multiple sub states for order lines. For example, if a builder has a project that required 500 boards, but the builder will only pick up or have a few boards delivered at a time over the course of the project, there could be quantities that are being picked, packed, and shipped at the same time.

Any time a line is selected, the remaining amount for the line will be auto-filled to assume that the remaining quantity is being processed. Using the above example, if 200 boards have already been picked and the line for boards is selected for picking, the remaining quantity of 300 will be automatically filled in the quantity. The same is true if 200 boards have already been invoiced. In that case, only the remaining quantity will be auto filled.

Continuing with the above example, if 200 boards are marked as packed and shipping is selected, the full amount of 500 will be autofilled. If only 200 boards are shipped, the system will assume that the previously packed boards are being shipped and the packed quantity will be decremented. If 201 boards are shipped, the packed boards are first decremented with the remaining single board being decremented from the quantity remaining.

## Line statuses

Order lines in the point of sale have several statuses to reflect the state of the order line. Statuses in the point of sale and back office do not match in all cases. Order line status can be viewed through the point of sale using the order fulfillment operations. In the back office, order lines can be viewed from the order details. Order details can be accessed through **Retail and Commerce > Customers > All customer orders**. Select the **Order ID** to view order details. From order details select the **Sales order** tab, then select **Detailed status** under the **View** subheader.

- **Pending** – Order lines that have been assigned to a store, but not yet accepted have the **Pending** status when viewed at the point of sale. Lines pending acceptance in the point of sale will have the **Order processing** status in the back office.
- **Accepted** – Order lines that have been manually accepted or automatically accepted will have the status of **Accepted** when viewed at the point of sale. Lines with the **Accepted** status will show as **Order processing** in the back office.
- **Picking** – Lines that are currently being picked at the store level have the status of **Picking**. Those same lines, when viewed at the back office, will show as **Order processing**.
- **Picked and Partially picked** – Lines that have been picked or partially picked at the point of sale will have the status **Picked** or **Partially picked**. The same lines in the back office will also show as **Picked** or **Partially picked**.
- **Packed and Partially packed** – Lines that have been packed or partially packed at the point of sale will have the status **Packed** or **Partially packed**. The same lines in the back office will also show as **Delivered** or **Partially delivered**.
- **Partially invoiced** – Lines that have been partially picked up or partially shipped will have the status **Partially invoiced** at the point of sale and in the back office.
- **Invoiced** – Lines that have been fully invoiced at the point of sale will no longer show up for fulfillment. In

the back office the status for those lines is **Invoiced**.

## Order fulfillment filtering

Order fulfillment at the point of sale includes filtering to help the user easily find what they need. Filters can be changed through the Action Pane at the bottom of the **Point of sale** screen. By default, a **Delivery type** filter is applied, based on how the operation is set up. If the operation is set up with the **All orders** parameter, then that filter is applied when accessing order fulfillment. The same applies for the **Store pickup** and **Ship from store** parameters. Other filters that can be applied to the order fulfillment view include:

- Customer number
- Customer name
- Customer email
- Order number
- Mode of delivery
- Receipt number
- Channel reference ID
- Originating store number
- Line status
- Created date
- Delivery date
- Receipt date

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Set up order fulfillment for stores

2/18/2021 • 8 minutes to read • [Edit Online](#)

## Overview

Many retailers would like to optimize order fulfillment by enabling stores to fill orders. Order fulfillment at the store level can help to ease overstock scenarios for a specific store, or may be needed from a logistical standpoint in cases where a store has extra capacity or is located within closer shipping distance to the customer. To address this need, a unified order fulfillment operation is available at the point of sale.

Orders for fulfillment at a specific store has the store's warehouse designated on the header or lines of the order.

The order fulfillment operation in the point of sale provides a single work area in the point of sale that can be used to process orders. This includes everything from accepting the order, to marking it as shipped, or initiating store pickup.

## Set up the order fulfillment operation

Order fulfillment, [Operation ID 928](#), can be used to access the store order fulfillment work area in the point of sale.

Follow the steps in [Add the operation to a button grid](#) to specify which parameter to use when invoking order fulfillment at the point of sale. By default, after specifying the order fulfillment operations, the **All orders** is selected. When configured with this parameter, the operation will list all order lines for fulfillment at the current store. Also available is **Orders to ship**, which can be assigned to a button and utilized when the user only wants to see orders that will ship out of the store. Finally, there is **Orders for pick up**. When invoked at the point of sale, this only lists orders to be picked up at the store. The different parameters can be assigned to different buttons to give the user a variety of ways to view order fulfillment.

### Enable users to access order fulfillment at the point of sale

The order fulfillment operation does not have its own permission out-of-the-box, but in the future, users may require the **Allow retrieve order** permission to invoke the operation from the point of sale.

At the store level, a configuration setting is available to determine whether an order line must be accepted manually from within the point of sale. If that configuration option is not set, order lines will be accepted by default. If that configuration option is turned on, users at the point of sale will need to have the **Allow accept order** permission to accept orders from within the point of sale.

### Enable manual order acceptance

By default, order lines assigned to a store are marked as **Accepted**. This means that it is assumed they will be fulfilled from the assigned store and will not be subject to further assignment. In certain cases, retailers may want to manually accept orders before they can be fulfilled. For example, if a store is short staffed and is unable to fulfill orders, a store manager will only accept as many orders for processing as they feel can adequately be processed in a given day. Until an order is accepted, it may be reassigned by the back office to a different store. In this way, order acceptance also provides a way to indicate that an order has been acknowledged by a store and will be fulfilled.

Order lines for store pickup are marked as **Pending** and are not subject to acceptance.

To turn on manual acceptance for order lines, navigate to **Retail and Commerce > Channels > Stores > All stores**. Select the store and click in the store ID to view the store's details. Click **Edit**. On the **General** FastTab, locate the **Order fulfillment** subheader and change **Manual accept** from **No** to **Yes**.

## Enable reject order line capability

Order lines can also be rejected from the point of sale. Rejecting an order line signifies that it will not be fulfilled at that store and sends the order line back for reassignment to another store or warehouse. Order line rejection permission is granted through the **Allow reject order** permission in the POS permission group associated with the worker. While rejecting a line, the retailers can mandate their workers to provide a reason for rejection. This can be achieved by using info codes of **Info code activity** type **Order fulfillment** and assigning the info code to **Reject order line** in the functionality profile associated with the channel.

### NOTE

Only the info codes of **Info code activity** type **Order fulfillment** can be assigned to the **Reject order line** action.

## Synchronize changes to the channel database

After the operation has been assigned to a button grid, the proper permissions have been assigned, and the channel is configured, the changes must be synchronized to the channel database. To do so, navigate to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**. Select schedule "1090-Registers" to sync button grid changes and then click **Run now**. Next, sync permissions changes by selecting "1060-Staff" and then click **Run now**. Next, sync channel changes by selecting "1070-Channel configuration" and click **Run now**. Finally, sync the newly created info code for reject reason by selecting the "1110-Global configuration" and click **Run now**.

## Use order fulfillment at the point of sale

Open the point of sale and select the order fulfillment operation. Depending on how it is configured, either all lines, order lines for pickup, or order lines to ship will be listed.

### Order fulfillment view

The order fulfillment view lists order lines for fulfillment at the store and includes the following columns:

- Order number
- Product number
- Description
- Quantity ordered
- Requested delivery date
- Customer name
- Fulfillment status

Additional information for a specific order line can be viewed by selecting the order line and then opening the flyout menu located just below the logged in user/shift information shown in the point of sale header. This menu includes 2 tabs: one for line details and another for order details. The line details tab includes the following information:

- Quantity ordered
- Quantity remaining to be shipped/picked up
- Quantity picked
- Quantity packed
- Quantity invoiced (already picked up or shipped)
- Mode of delivery
- Delivery address

The details flyout menu also has a tab that provides more order level details including:

- Customer name

- Customer ID
- Order number
- Order total
- Order balance

At the bottom of the order fulfillment view is an Action Pane. This contains all of the actions that can be taken against an order line. If an action is not available based on a line's status, that action will be unavailable.

By default, orders will have a status of **Accepted**. Order status can be viewed as a column in the list of order lines. If **Manual accept** is configured at the channel level, all lines to be shipped will show as **Pending** and must be accepted before they can be fulfilled. Orders for store pickup are **Pending** by default and do not need to be accepted.

### Order fulfillment line actions

- **Edit** – If an order status is pending, it can be edited at the point of sale. Orders that have already been partially picked, packed, or invoiced cannot be edited from the order fulfillment view.
- **Accept** – If **Manual accept** is configured at the channel level, lines must be first accepted before they can move through the order fulfillment process.
- **Pick** – The pick option supports several actions. First, **Picking** updates the status of the order line so others in the store do not attempt to pick the same line. Next, **Print picking list** prints a picking list for the selected line or lines and also updates their status to **Picking**. Picking list formats are controlled as part of receipt formats. For more information about how to set up receipt formats, see [Receipt templates and printing](#). Finally, **Mark as picked** indicates the line has been picked. **Mark as picked** initiates corresponding inventory transactions in the back office. Picking actions can be performed at the same time for multiple order lines across orders and for all modes of delivery.
- **Reject** – Lines or partial lines can be rejected. This allows them to be reassigned from the back office to another store or warehouse. Lines can only be rejected if they have not yet been picked or packed. To reject a line that has already been picked or packed, that line must be unpicked or unpacked from the back office.
- **Pack** – The pack option supports two actions: **Print packing slip** will print a packing slip for the selected lines and **Mark as packed** will mark the lines as packed and mark the lines as delivered in the back office. Only order lines that belong to the same order and have the same mode of delivery can be packed at the same time. Packing slip formats are controlled as part of receipt formats. For more information about how to set up receipt formats, see [Receipt templates and printing](#).
- **Ship** – The ship action will mark selected lines as **Delivered** in the back office. After a line has been fully shipped, it will no longer appear in the order fulfillment view.
- **Pickup** – The pickup action adds the lines to the transaction view for pickup. If there are other lines on the order that aren't currently being picked up, they will be added to the transaction view with quantity zero. After a line has been fully picked up, it will no longer appear in the order fulfillment view.

### Order fulfillment filtering

Order fulfillment at the point of sale includes filtering to help the user easily find what they need. Filters can be changed on the Action Pane at the bottom of the **Point of sale** screen. By default, a **Delivery type** filter is applied, based on how the operation is set up. If the operation is set up with the parameter **All orders**, then that filter is applied when accessing order fulfillment. The same applies for the **Store pickup** and **Ship from store** parameters. Other filters that can be applied to the order fulfillment view include:

- Customer number
- Customer name
- Customer email
- Order number
- Mode of delivery
- Receipt number

- Channel Reference ID
- Originating store number
- Line status
- Created date
- Delivery date
- Receipt date

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Show order notifications in the point of sale (POS)

2/18/2021 • 5 minutes to read • [Edit Online](#)

In the modern retail environment, store associates are assigned various tasks, such as helping customers, entering transactions, doing stock counts, and receiving orders in the store. The point of sale (POS) client provides a single application where associates can perform all these tasks and many others. Because various tasks must be performed during the day, associates might have to be notified when something requires their attention. The notification framework in the POS helps by letting retailers configure role-based notifications. As of Dynamics 365 for Retail with application update 5, these notifications can be configured only for POS operations.

Currently, the system can show notifications only for order fulfillment operations. However, because the framework is designed to be extensible, developers will eventually be able to write a notification handler for any operation and show the notifications for that operation in the POS.

## Enable notifications for order fulfillment operations

To enable notifications for order fulfillment operations, follow these steps.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS > Operations**.
2. Search for the **Order fulfillment** operation, and select the **Enable notifications** check box for it to specify that the notification framework should listen to the handler for this operation. If the handler is implemented, notifications for this operation will then be shown in the POS.
3. Go to **Retail and Commerce > Employees > Workers >**, under Commerce tab, open the POS permissions associated with the worker. Expand the **Notifications** FastTab, add the **Order fulfillment** operation, and set the **Display order** field to **1**. If more than one notification is configured, this field is used to arrange the notifications. Notifications that have a lower **Display order** value appear above notifications that have a higher value. Notifications that have a **Display order** value of **1** are at the top.

Notifications are shown only for operations that are added on the **Notifications** FastTab, and you can add operations there only if the **Enable notifications** check box for those operations has been selected on the **POS operations** page. Additionally, notifications for an operation are shown to workers only if the operation is added to the POS permissions for those workers.

### NOTE

Notifications can be overridden at the user level. Open the worker's record, select **POS permissions**, and then edit the user's notification subscription.

4. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. In the **Notification interval** field, specify how often notifications should be pulled. For some notifications, the POS must make real-time calls to the back-office application. These calls consume the compute capacity of your back-office application. Therefore, when you set the notification interval, you should consider both your business requirements and the impact of real-time calls to the back-office application. A value of **0** (zero) turns off notifications.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**. Select the **1060 (Staff)** schedule to synchronize notification subscription settings, and then select **Run now**. Next, select the **1070 (Channel configuration)** schedule to synchronize the permission interval, and then select

Run now.

## View notifications in the POS

After you complete the preceding steps, the workers will be able to view the notifications in the POS. To view notifications, press the notification icon in the top right corner of the POS. A notification center appears and shows notifications for the order fulfillment operation. The notification center should show the following groups in the order fulfillment operation:

- **Store pickup** – This group shows the count of orders that have a delivery mode of **Pickup**, and that are scheduled for pickup from the current store. You can press the number on the group to open the **Order fulfillment** page. In this case, the page will be filtered so that it shows only the active orders that are set up for pickup from the current store.
- **Ship from store** – This group shows the count of orders that have the delivery mode of **Shipping**, and that are scheduled for shipment from the current store. You can press the number on the group to open the **Order fulfillment** page. In this case, the page will be filtered so that it shows only the active orders that are set up for shipment from the current store.

When new orders are assigned to the store for fulfillment, the notification icon changes to indicate that there are new notifications, and the count for the appropriate groups is updated. Even though the groups are refreshed at regular intervals however, POS users can manually refresh the groups at any time by selecting the **Refresh** button next to the group. Lastly, if a group has a new item, that the current worker hasn't viewed, then the group shows a burst symbol to indicate new content.

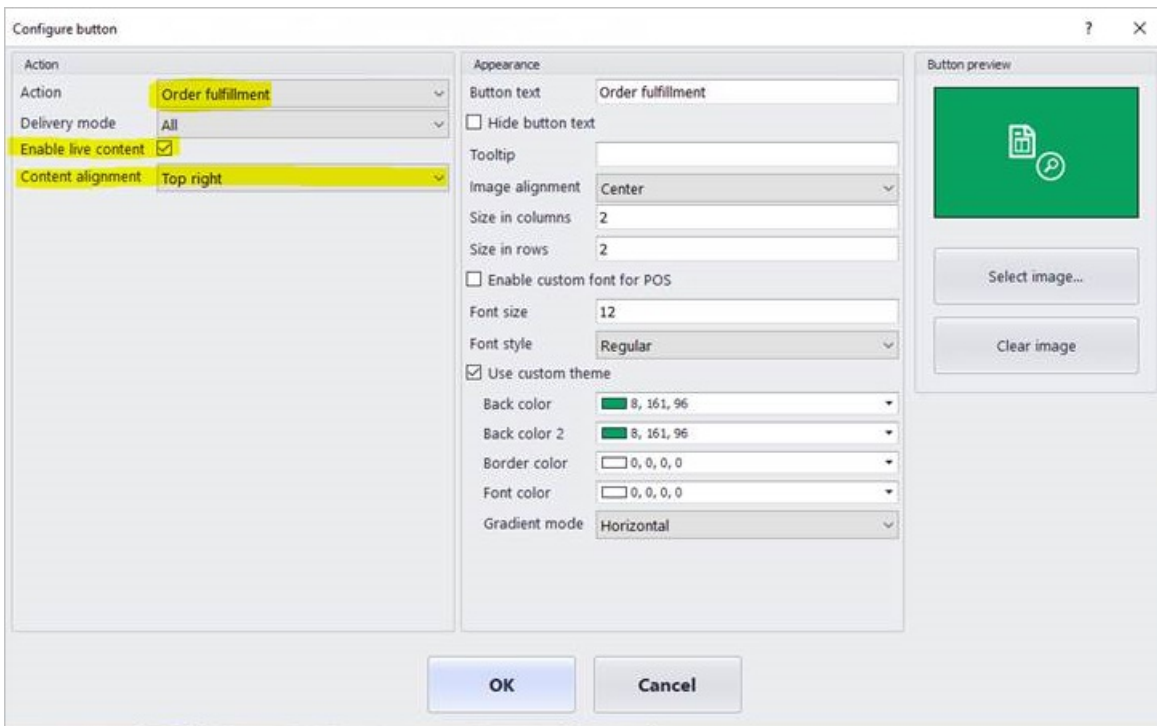
## Enable live content on POS buttons

POS buttons can now show a count to help workers easily determine which tasks require their immediate attention. To show this number on a POS button, you must complete the notification setup that is described earlier in this topic (that is, you must enable notifications for an operation, set up a notification interval, and update the POS permission group for the worker). Additionally, you must open the button grid designer, view the button's properties, and select the **Enable live content** check box. In the **Content alignment** field, you can select whether the count appears in the upper-right corner of the button (**Top right**) or in the center (**Center**).

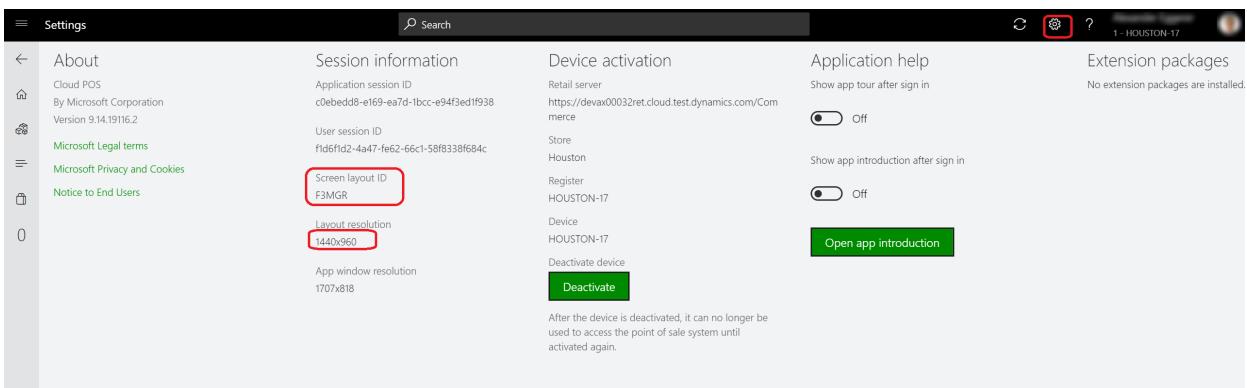
### NOTE

The live content can be enabled for operations only if the **Enable notifications** check box has been selected for them on the **POS operations** page, as described earlier in this topic.

The following illustration shows the live content settings in the button grid designer.



To show the notification count on a button, you need to ensure that the correct screen layout is being updated. To determine the screen layout that is being used by the POS, select the **Settings** icon in upper-right corner and note the **Screen layout ID** and **Layout resolution**. Now using Edge browser, go to the **Screen layout** page, find the **Screen layout ID** and **Layout resolution** identified above and select the **Enable live content** check box. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule** and run the 1090 (Registers) job to synchronize layout changes.



The following illustration shows the effect of selecting **Top right** versus **Center** in the **Content alignment** field for buttons of various sizes.

The image shows a Windows Start menu on the left and a Notifications window on the right. The Start menu features several tiles: 'Current transaction' (shopping cart icon), 'Return transaction' (shopping cart with arrow icon), 'Order fulfillment - ALL' (two tiles, each with a document icon and a count of 1), 'Order fulfillment - Pick Up' (calendar icon, count 1), 'Order fulfillment - Ship' (calendar with checkmark icon, count 0), and 'Select hardware station' (two small tiles, each with a count of 1). The Notifications window is titled 'Order fulfillment' and shows 'Last updated: 01/17/2018 5:43 PM'. It contains two sections: 'Store pickup' with a count of 1, and 'Ship from store' with a count of 0.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Using continuity program

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through selling a continuity program and processing related sales orders. To complete this procedure, the user has to be set up as a call center user. This procedure uses the USRT demo data company.

1. Go to Retail and Commerce > Customers > Customer service.
2. In the SearchText field, type 'Karen' and then press the Tab key.
  - The advanced search dialog should pop up. If it doesn't, click Search to the right of this field.
3. In the list, mark the selected row.
  - There should be only one row with Karen Berg showing. Select the row by clicking on the checkmark column on the far left of the grid.
4. Click Select.
5. Click New sales order.
  - It's a good idea to note the sales order number. You'll need it later in this procedure.
6. In the Item number field, type '88000' and then press the Tab key.
  - This is a continuity item in the USRT demo data.
7. Click Complete.
8. In the Payment method field, enter 'Visa'.
9. Click Add credit card.
  - Enter the required credit card information on this page.
10. Click OK.
11. Expand the Payment section.
  - To submit a call center order, payments have to be entered for the order.
12. Click OK.
13. Click Submit.
  - You're done creating a new continuity order. Next, you'll run two batch processes that are used to process the continuity orders.
14. Close the page.
15. Go to Retail and Commerce > Continuity > Process continuity payments.
16. In the Continuity item field, type '88000' and then press the Tab key.
17. Click OK.
18. Go to Retail and Commerce > Continuity > Create continuity child orders.
  - This process will create new sales orders based on the settings of your continuity programs.
19. In the Continuity item field, type '88000' and then press the Tab key.
  - Item '88000' is a continuity item in the USRT demo data.
20. In the Sales order field, enter or select a value.
  - Enter the sales order number that you noted earlier in the procedure. This will keep the processing time to a minimal for this procedure. The Sales order field field is optional--you could process all orders for any one program.
21. Click OK.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Linked refunds of previously approved and confirmed transactions

2/18/2021 • 4 minutes to read • [Edit Online](#)

Returns are an important operation for retailers. The ability to accept returns for sales and refund payments to customers gives retailers a way to service the needs of customers and to help resolve their issues.

This topic provides information about how to configure and use linked refunds. A linked refund is a refund of a transaction that was previously approved and confirmed. The refund can be either a full refund or a partial refund of the transaction, and it can't exceed the full amount of the original authorization. The functionality for linked refunds is available in Microsoft Dynamics 365 Retail version 10.0.1.

In Microsoft Dynamics 365 Retail version 10.0 and earlier, retailers can process refunds to cards, but cashiers must manually specify these refunds. Cashiers can process refunds to the original mode of payment only if the customer provides that mode of payment. Therefore, by providing new card details, customers can use the return process to move balances from one card to another and therefore do unauthorized card balance transfers.

By using linked refunds, retailers can greatly reduce risk by making sure that refunds are processed only to the card that was authorized during the original transaction. To help prevent unauthorized card balance transfers, the system can prompt cashiers to use the confirmed and approved card token to process refunds. By using the original mode of payment for refunds, retailers can help reduce their card authorization costs.

## Prerequisites

[Payment method setup](#)

[Omni channel payments setup](#)

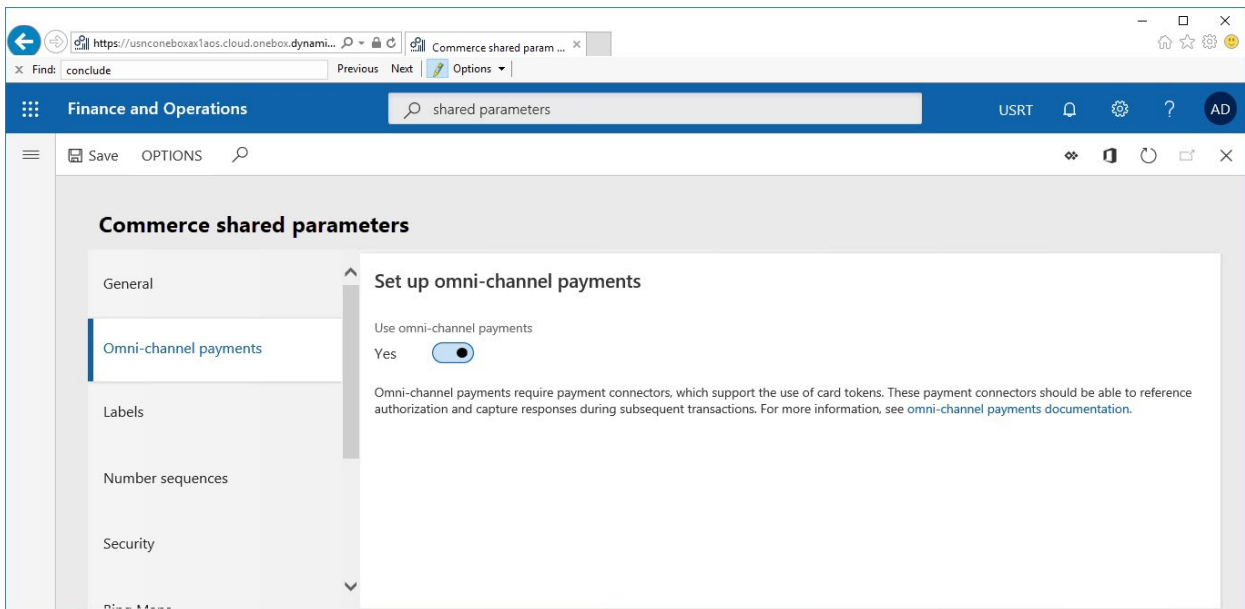
### **Additional setup**

Customers who aren't using the out-of-box implementation of the Adyen Connector must set up the connector that supports tokenization of credit cards. All the scenarios that are described in this topic can be implemented by using the standard Payments software development kit (SDK) that is provided with Commerce. The [Dynamics 365 Payment Connector for Adyen](#) provides an out-of-box implementation of every scenario that is described here.

## Turn on the linked refunds functionality

The linked refunds functionality works with the omni-channel payments functionality that is available in Microsoft Dynamics 365 Retail 8.1.3 and later.

To turn on the linked refunds functionality, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**. On the **Omni-channel payments** tab, set the **Use omni-channel payments** option to **Yes**.



When you turn on the omni-channel payments functionality, you change the business process flow for calculating shipping charges and other charges, and for adding those charges to point of sale (POS) sales. Therefore, make sure that you test and train your employees before you turn on this functionality.

When the omni-channel payments functionality is turned on, the card payment tokens that are used in one channel (for example, a call center or Modern POS (MPOS)) will be available in all channels that are set up for the retailer. For POS applications, the linked refunds functionality will also be turned on. For call center, MPOS, and e-Commerce applications, customers can still manually enter card numbers for payment.

### Supported flows

Cashiers can process a refund to the card that was used during the original transaction, even if the card isn't presented for the return.

- Linked refunds for cash-and-carry transactions that use credit or debit cards
- Linked refunds for customer orders that use credit or debit cards

### Unsupported flows

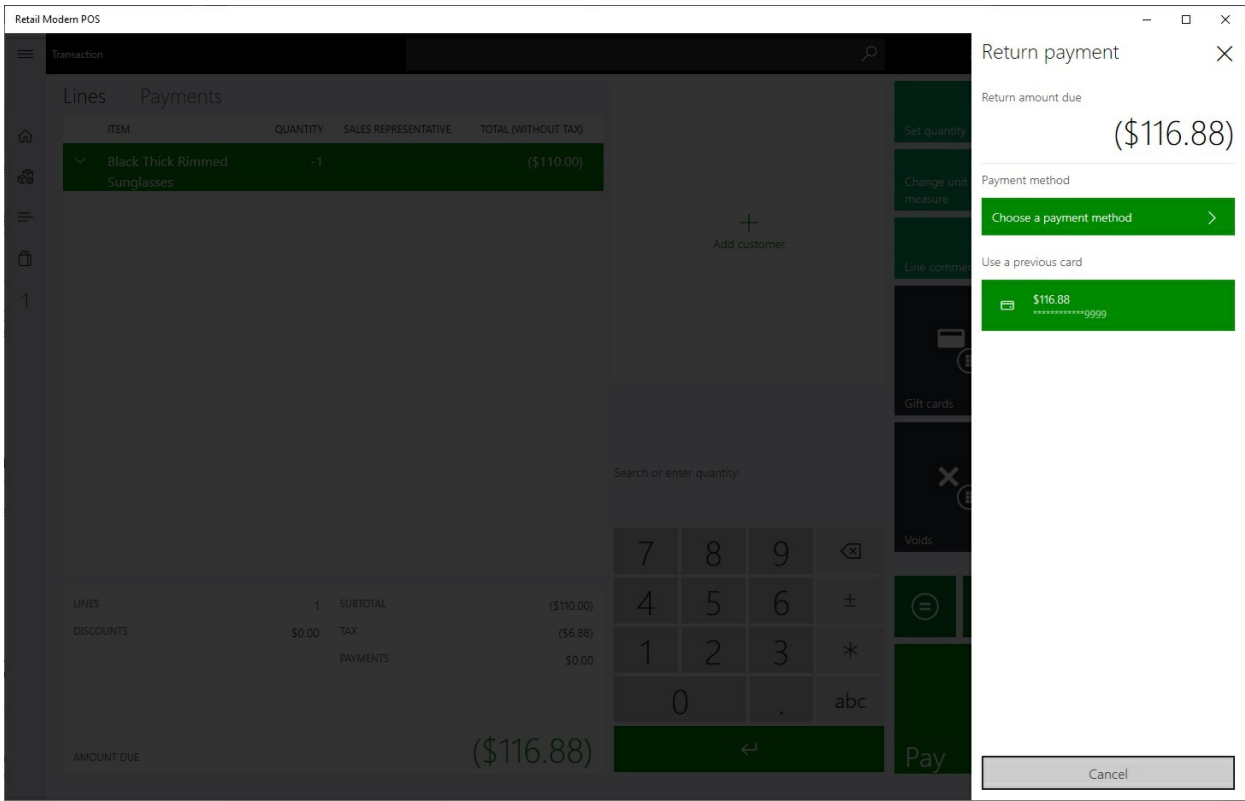
- Linked refunds for transactions that use gift cards
- Linked refunds for transactions that use loyalty cards
- Linked refunds for exchange orders
- Multiple return orders in the same transaction
- Returns without a receipt or customer account details

## Use case examples

This section presents examples of use cases to help you understand the configuration and use of linked refunds and payment authorizations in the context of a customer order or a return where a receipt is presented. These examples show the behavior of the application when the **Omni-channel payments** parameter has been turned on.

### Customer account-based or receipt-based return that has a single card authorization

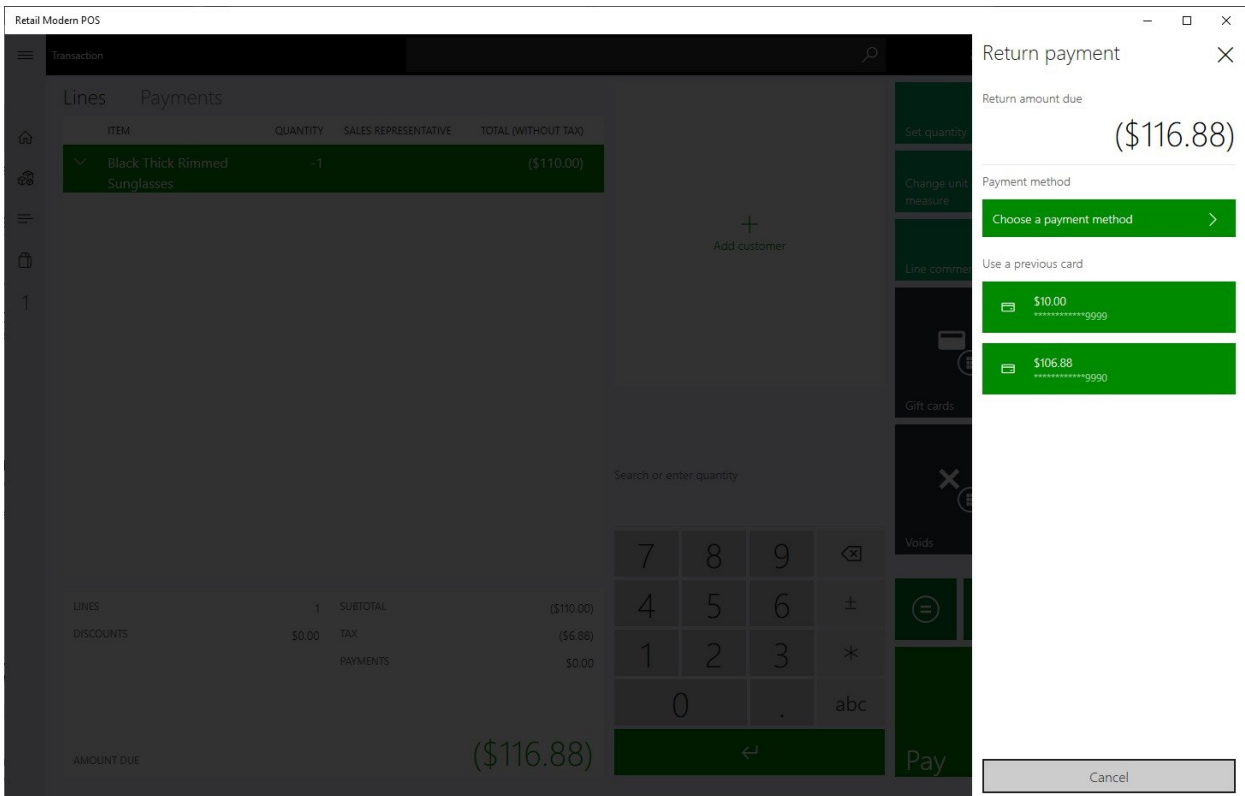
A customer comes to return an item that was purchased by using a single credit card. The customer provides a receipt, and the return is being made within the allowed period for returns. When the cashier scans the receipt, the item for return is processed. When the cashier processes the payment refund by selecting the button for any payment method, the existing credit card authorization is shown.



When the cashier selects the credit card authorization, the payment refund is processed, and the **Transaction end** screen appears. If a receipt printing is configured, the cashier is prompted to print a receipt.

**Customer account-based or receipt-based return that has multiple card authorizations**

A customer comes to return an item that was purchased by using multiple credit cards. When the cashier scans the receipt, the item for return is processed. When the cashier processes the payment refund by selecting the button for any payment method, all the existing credit card authorizations are shown.



When the cashier selects a credit card authorization, the payment refund is processed. If more must be refunded, the current transaction screen shows the remaining amount. When the cashier processes the payment refund for this amount, the remaining credit card authorizations are shown. This process continues until there is no

remaining amount that must be refunded.

After the full amount is successfully refunded, the cashier can complete the transaction, and a receipt can be printed as configured.

## Related topics

- [Payments FAQ](#)
- [Dynamics 365 Payment Connector for Adyen](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Sell and return products that aren't part of a store's assortment

2/18/2021 • 2 minutes to read • [Edit Online](#)

A common scenario for any retailer is to sell products to their customers or accept returns from their customers even if they don't carry the specific products in their store (in other words, the products are not assorted to the store).

Here are some typical scenarios:

- A retailer doesn't carry all its products in a specific store. The remaining products are stored in the warehouse. The store associate can assist the customer by searching or browsing for the products in the warehouse, add them to the cart, and complete the checkout by selecting a delivery method, such as shipping to an address from the warehouse or letting the customer pick up the product from the current store or from another store.
- A retailer doesn't carry specific products in the store or doesn't have them in stock at the store the customer visited, but the products are available in other stores. The store associate can assist the customer by searching or browsing the products in the other store, add them to the cart, and complete the checkout by selecting a delivery method.
- A retailer has many stores in and around a specific city or zip code and doesn't want to force the customers to return products to the same store they were purchased in. Instead, customers can return products to any store.

Those common scenarios are available for retailers using Commerce. With Commerce, you can:

- Search or browse products at other stores.
- Search or browse all released products.
- Create cash-and-carry transactions or customer orders.
- Select delivery options for customer orders.
- Pick up products at the current store or another store.
- Cancel an order at the current store or another store.
- Return an order with or without the receipt at the current store or another store.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and process an exchange on a return order

2/18/2021 • 2 minutes to read • [Edit Online](#)

In previous versions of Dynamics 365 Commerce, returns against customer orders were processed by using the return order document in Headquarters. However, the return order document can be used to process only products that are being returned. The returned products are indicated by a negative quantity on the return order lines. By contrast, sales are indicated by a positive quantity. However, the return order document doesn't support positive quantities. Because of this limitation, previous versions of the app didn't support scenarios where product exchanges are done by using the return order document.

However, functionality has been added to support scenarios where exchanges are done on return orders. Commerce now uses the sales order document instead of the return order document to process these types of transactions.

## Configure Commerce to support exchanges on return orders

Follow these steps to configure the system to support exchanges on return orders.

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**. On the **Customer orders** FastTab, set the **Process return orders as sales orders** option to **Yes**.
2. Run the **Global configuration distribution schedule** job (1110).

## Make an exchange

After the system is configured as described in the previous section, the point of sale (POS) user will still select a sales order or sales invoice to process a return, as in previous versions of the app. However, after the return items are added to the cart, the user will be able to add new sales lines to the cart.

For these new sales lines, the user must define all the attributes that are required in order to process a customer order line. These attributes include the delivery method and fulfillment location. The payment that is due for the transaction will be a net of the return order lines and sales order lines. When payment is tendered for the transaction, the return order will be posted as a sales order document in Headquarters, and the system will immediately invoice the return lines.

To provide better visibility into the various amounts for the cart, three new amount fields have been added to the cart. You can use the screen designer to make these new fields available in the POS user interface (UI).

- **Deposit applied** – The deposit amount that is applied on a transaction when the user does a customer order pickup. If there is no deposit override, and a 10-percent deposit is configured, the amount in this field is 90 percent of the total amount of the customer order.
- **Carry out amount** – The total amount for lines where the delivery mode was set to **Carry out** when the customer order was created or edited, or during a customer order exchange. The amount in this field includes taxes and charges.
- **Return amount** – The total amount for lines that have negative quantities during the customer order exchange. The amount in this field includes taxes and charges.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Restrict payment methods for returns without a receipt

2/18/2021 • 2 minutes to read • [Edit Online](#)

Each payment type that a retailer accepts must be configured when the system is set up. This topic describes how certain payment types can be restricted for refund if the returns are made without a receipt.

## Set up payment methods

To set up payment methods, the following tasks must be completed.

1. Create the payment methods that are accepted by the entire organization.
2. Create organization-wide card types and card numbers. If credit cards or debit cards are accepted, you must create one payment method for cards, and then create the organization-wide card types and card numbers.
3. Set up store payment methods. Associate payment methods with each store, and then enter the store-specific settings for each payment method.
4. Set up card payment methods for stores. For any card payment methods that the store accepts, complete the card setup.

The screenshot shows the Dynamics 365 Finance and Operations interface. The breadcrumb navigation is 'Commerce > Channels > Stores > All stores'. The main navigation bar includes 'Edit', '+ New', 'Delete', 'Configuration status', 'STORE', 'SET UP', and 'OPTIONS'. The 'SET UP' section is expanded, showing 'Payment methods' highlighted in yellow. Other options include 'Income/expense account', 'Channel database', 'Cash declaration', 'Sections', and 'Fulfillment group assignment'. Below the navigation, there is a message: 'Click the edit button to make changes.' The 'STORES' section contains a search filter and a table with the following data:

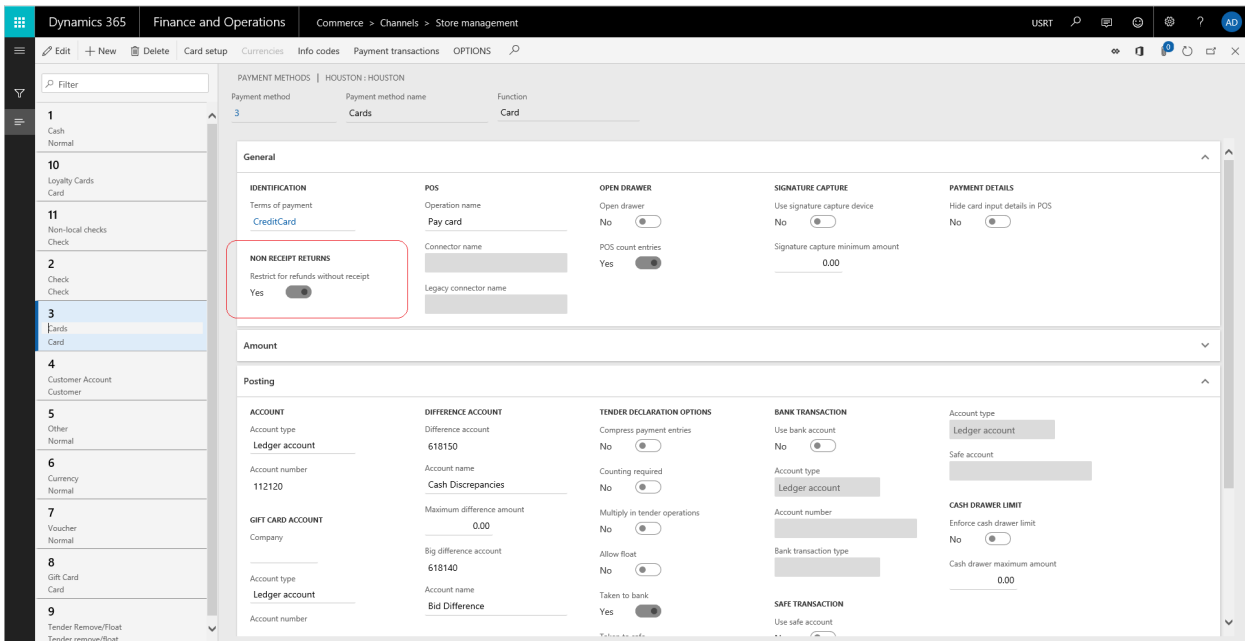
Channel Id	Name	Channel type	Operating unit ...	Store number	Warehouse	Sales tax
000001	Annapolis	Retail store	036	ANNAPOL	ANNAPOL	MD

## Restrict payment methods for returns without a receipt

For each store payment method, on the **Store management** page, under **Non receipt returns**, set **Restrict for refunds without receipt** to **Yes**.

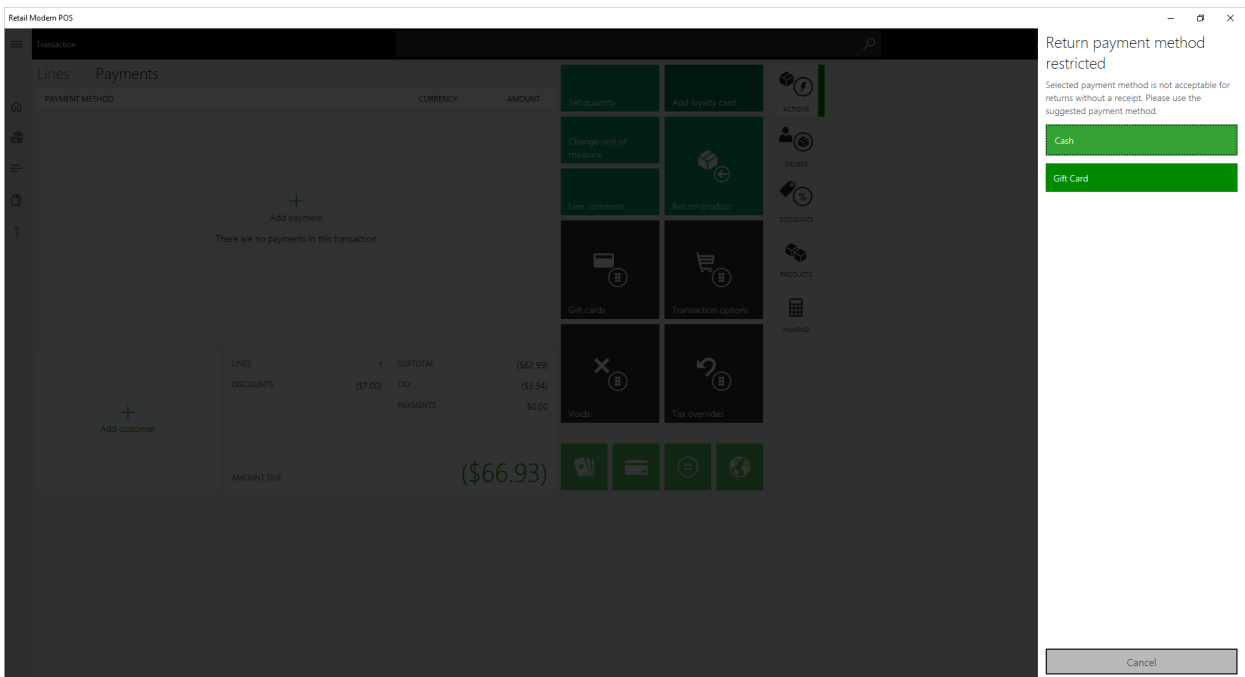
The default value of the toggle is **No**, which ensures that the payment method is allowed for refunds.

When **Restrict for refunds without receipt** is set to **Yes**, the selected payment method will not be allowed for refunds.



**NOTE**

When a cashier selects a payment method that is restricted for refund without a receipt, a message displays to verify the acceptable payment methods.



If a transaction has both a receipted return and a return without a receipt, the restriction conditions will not be enforced because the transaction will be a return workflow with a receipt.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Return items across multiple customer orders and invoices

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes two features that optimize customer order returns over multiple invoices.

## Enable refunds over multiple captures

This feature enables multiple linked refunds against the same customer order.

1. Go to the **Feature management** workspace and search for **Enable refunds over multiple captures**.
2. Select **Enable refunds over multiple orders** and then click **Enable**.

## Enable proper tax calculation for returns with partial quantity

This feature ensures that when an order is returned using multiple invoices, the taxes will ultimately be equal to the tax amount originally charged.

1. Go to the **Feature management** workspace and search for **Enable proper tax calculation for returns with partial quantity**.
2. Select **Enable proper tax calculation for returns with partial quantity** and then click **Enable**.

## Process returns

After these features are turned on and the changes are synchronized to the stores, the cashier in the store can select multiple sales orders for a customer for their return.

When the orders are selected, a list of all the returnable products across all the invoices for the orders will display. The cashier can then select the products to return. A single return order will be created for all the selected products.

If the order is fully returned, the amount of taxes returned to the customer will be equal to the amount of tax originally charged.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create and update a returns and refunds policy for a channel

2/18/2021 • 3 minutes to read • [Edit Online](#)

## Overview

The channel return policy in Dynamics 365 Commerce enables retailers to set enforcements on which payment tenders can be allowed for processing a return on a point of sale (POS) device.

This topic describes the steps to set up a returns and refunds policy for a channel.

The scope of the policy is currently limited to setting the payment tenders that can be allowed for a channel. The "allowed" list is based on the payment methods used to make the purchase. For example:

- If a purchase was made using a gift card, the store policy is to process refunds only to a new gift card or to give store credit.
- If a sale is made using cash, the options allowed for refund are cash, gift card, and customer account, but not credit card.

## Enable return policy

To enable the channel return policy functionality, do the following:

1. Go to the **Feature Management** workspace in Dynamics 365 Commerce.
2. Search for the **Enable channel return policies** feature in the list of feature names.
3. Select **Enable now**.

## Configure return policy

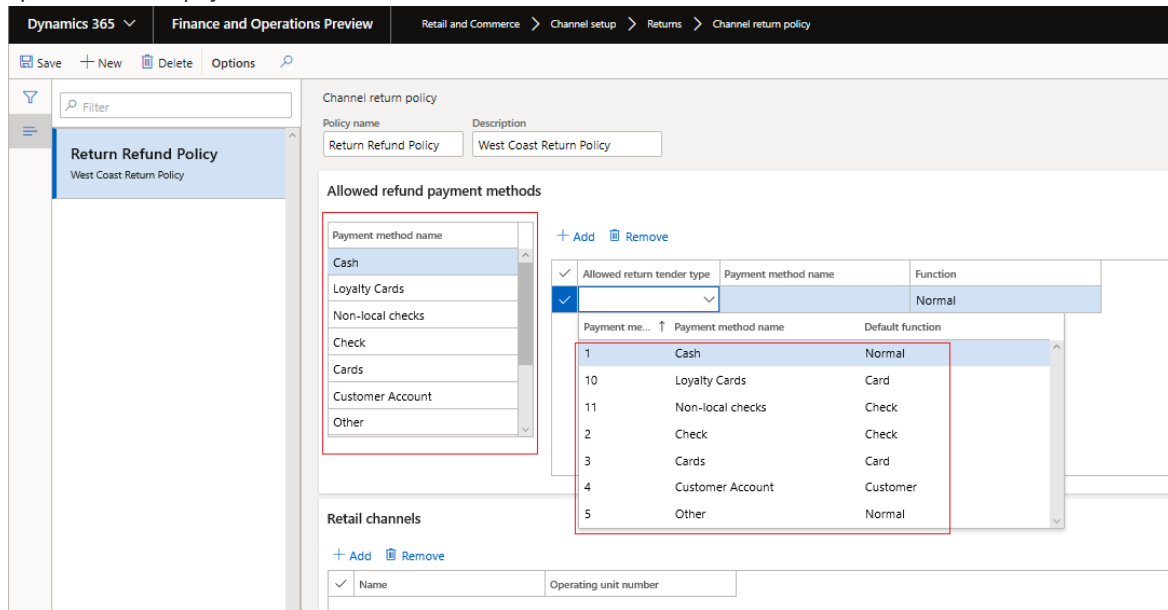
Follow these steps to configure a return policy for a retail store or online retail channel.

1. Go to **Retail and Commerce > Channel Setup > Returns > Channel return policy**.
2. Select **New** to create a new return policy template. To use an existing template, select the template in the left pane. For new templates, add a name and description that will help you identify the policy when it is being applied to the channel.

The screenshot displays the Dynamics 365 Commerce interface for configuring a channel return policy. The breadcrumb navigation at the top reads: Retail and Commerce > Channel setup > Returns > Channel return policy. The left-hand navigation pane shows the 'Return Refund Policy' section selected, with a sub-item 'West Coast Return Policy'. The main content area is titled 'Channel return policy' and contains the following elements:

- Channel return policy header:** A table with two columns: 'Policy name' (containing 'Return Refund Policy') and 'Description' (containing 'West Coast Return Policy').
- Allowed refund payment methods:** A section with a list of payment methods on the left: Cash, Loyalty Cards, Non-local checks, Check, Cards, Customer Account, and Other. To the right is a table with columns 'Allowed return tender type', 'Payment method name', and 'Function'. The table is currently empty, displaying a 'We didn't find anything to show here.' message.
- Retail channels:** A section with a table with columns 'Name' and 'Operating unit number'. This table is also empty, displaying a 'We didn't find anything to show here.' message.

3. In the **Allowed refund payment methods** section, define **Allowed** return payment tenders that are specific to each payment method.

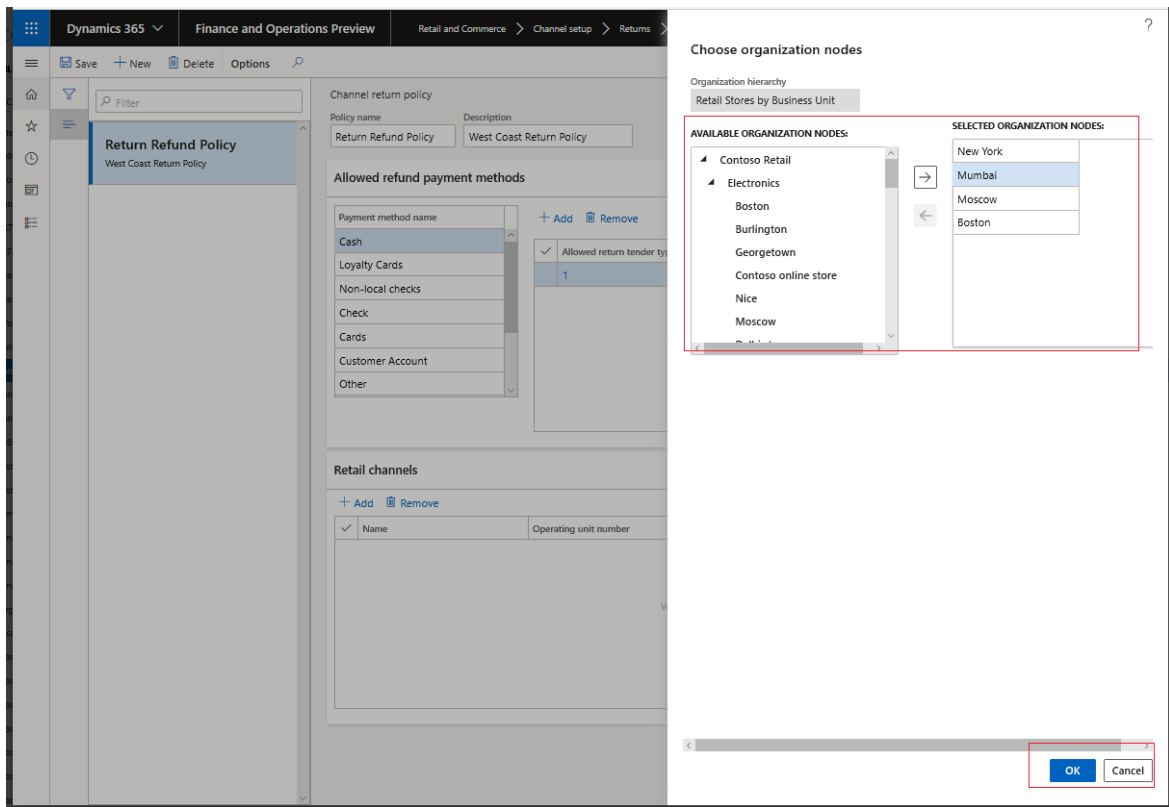


#### IMPORTANT

- The payment methods are derived from the payment methods set for the organization.
- Adding an allowed return tender type for each listed payment method will ensure that returns can be made to the allowed return tender type.

4. Associate the return policy template with the stores where it will be used. Select **Add** in the **Retail Channels** tab and associate the available channels.

- In the **Choose organization nodes** dialog box, select the stores, regions, and organizations that the template should be associated with.
- Only one return policy template can be associated with each store.
- Use the arrow buttons to select stores, regions, or organizations.
- The effective date on the policy will be the date on which the policies are applied to the channels and the channel jobs are run.



5. On the **Distribution schedule** page, run the **1070** job to make the channel return policy available to the POS.

## Preview the channel return policy in the POS

Follow the steps in either of the following examples to view the allowed return tender types in POS.

1. Sign in to the POS as a cashier or manager.
2. Under **Shift and Drawer**, select **Show journal**.
3. Select the transaction that is part of the return.
4. Select the items to refund, and choose the payment method.
  - If the payment tender selected is in the allowed list of return tender types, the cashier can complete the transaction.
  - If the payment tender selected is not allowed, an error message is displayed.
  - Select **Amount Due** to display a list of all the allowed return tender types.

-or-

1. Sign in to the POS as a cashier or manager.
2. Select **Return Transaction** and enter the receipt ID using a barcode scan or by manual entry.
3. Select the transaction that is part of the return.
4. Select the items to refund, and choose the payment method.
  - If the payment tender selected is in the allowed list of return tender types, the cashier can complete the transaction.
  - If the payment tender selected is not allowed, an error message is displayed.
  - Select **Amount Due** to display a list of all the allowed return tender types.

Transaction

Lines Payments

ITEM	QUANTITY	SALES REPRESENTATIVE	TOTAL (WITHOUT TAX)
Boot-Leg Vintage Wash Jeans	-1		(\$270.00)
Casual Fit Button-Down Shirt	-1		(\$65.00)

There are no product recommendations for the items in the cart.

Search or enter quantity

LINE	QUANTITY	SUBTOTAL	TAX	PAYMENTS	CHARGE DUE
2		(\$335.00)	\$0.00	\$0.00	(\$335.00)

Return payment

You can't refund using this tender type, this would violate the refund policy.

OK

Transaction

Lines Payments

ITEM	QUANTITY	SALES REPRESENTATIVE	TOTAL (WITHOUT TAX)
Boot-Leg Vintage Wash Jeans	-1		(\$270.00)
Casual Fit Button-Down Shirt	-1		(\$65.00)

There are no product recommendations for the items in the cart.

Search or enter quantity

LINE	QUANTITY	SUBTOTAL	TAX	PAYMENTS	CHARGE DUE
2		(\$335.00)	\$0.00	\$0.00	(\$335.00)

Return payment

Return amount due (\$335.00)

Refund options

- (\$335.00) Cash
- (\$335.00) Cards

Cancel

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Clienteling overview

2/18/2021 • 11 minutes to read • [Edit Online](#)

Many retailers, especially high-end specialty retailers, want their sales associates to form long-term relationships with their key customers. The associates are expected to know about these customers' likes and dislikes, purchase history, product preferences, and important dates, such as anniversaries and birthdays. Associates need a place where they can capture this information and easily find it when it's required. If this information is available in a single view, the associates can easily target customers who meet specific criteria. For example, they can find all customers who prefer to shop for handbags, or customers who have an important event approaching, such as a birthday or anniversary.

## Client book

In Microsoft Dynamics 365 Commerce, retailers can use the client book functionality to help store associates form long-term relationships with key customers.

The client book includes customer cards that show contact information for each customer, together with three more properties that are defined by the retailer and configured in Headquarters. Retailers can decide the three most important things that sales associates should know about customers. For example, a jewelry retailer might want to include important dates such as anniversaries or birthdays, because these dates are occasions when people might buy more jewelry. Similarly, a fashion retailer might want to include the customer's preferred shopping interests and brands.

The client book also lets sales associates filter the list so that it shows only customers who meet specific criteria. For example, a new collection of shoes has arrived in the store, and an associate wants to inform customers who like to buy shoes. In this case, the associate can filter the client book to find the relevant customers and then take further action.

If any customers are no longer considered key customers for some reason, and therefore should not be closely managed, sales associates can remove them from their client book.

Some retailers don't want to manage customers at the sales associate level. Instead, they want to manage a list of key customers at the store level. These retailers can view customers from store client books. By using this option, retailers can view the customers from the client books of all the sales associates whose address book matches the address book of the current store. In this way, if an associate works in multiple stores of the legal entity, the client book shows the customers from all those stores. This functionality supports additional capabilities. For example, customers can be reassigned from one associate to another associate. This capability is useful when associates are transferred or leave the company.

Each sales associate can have one client book per legal entity, and associates can add one or more customers to their client book. In Commerce, each customer can currently be added to only one client book. However, Microsoft plans to add functionality that lets a single customer be added to multiple client books.

### NOTE

Unlike customer search, the client book doesn't filter customer records based on the store's address books.

## Activities and notes

Online channels give retailers ways to learn about customer preferences without requiring that customers explicitly provide this information. By contrast, when customers interact with sales associates in the store, they

explicitly share information about their preferences. Unfortunately, this information can be lost after the sale is over. However, if this information is recorded, it can help retailers better understand customers, and therefore help them provide better recommendations and a better overall shopping experience.

To capture the critical information that customers share, sales associates can use not only the client book attributes, but also use the activities and notes functionality. Retailers can configure the activity types, such as information about the store visit, email address, phone number, and appointments. Activities that associates create can be viewed in a timeline format at the point of sale (POS). They can also be viewed on the **Activities** tab on the **All customers > General** page in Headquarters.

Sales associates can also use notes to capture generic customer information that can be referenced before every interaction. These notes are saved in Headquarters, and can be viewed in the customer profile or on the customer details page in the call center.

#### NOTE

Currently, all notes and activities can be viewed by any sales associate who works for the legal entity and can view customer details pages. Notes and activities aren't limited to the associate who added a customer to the client book.

## Integration with Dynamics 365 Customer Insights

By using the Dynamics 365 Customer Insights application, retailers can aggregate data from the various systems that customers use to interact with the retailer's brand. They can then use this data to generate a single view of the customer and derive insights. The integration of Customer Insights with Commerce lets retailers select one or more measures that should be shown on the customer card in the client book. For example, retailers can use the data in Customer Insights to calculate the "churn probability" for a customer and define the "next best action." If these values are defined as measures, they can be shown on the customer card and can provide crucial information to sales associates. For more information about Customer Insights, see the [Dynamics 365 Customer Insights](#) documentation. For more information about measures, see [Measures](#).

## Set up clienteling

To turn on the clienteling functionality in your environment, follow these steps.

1. In the **Feature management** workspace, filter the features by the **Retail and commerce** module.

The screenshot shows the 'Feature management' interface. At the top, there are statistics: 'New' (28), 'Not enabled' (104), and 'Scheduled' (0). There are buttons for 'Enable all' and 'Check for updates'. Below the statistics, there are tabs for 'New', 'Not enabled', 'Scheduled', and 'All'. A search bar contains 'Retail and commerce'. A table lists features with columns for 'Enable date', 'Feature added', and 'Module'. The 'Clienteling' feature is highlighted in blue, and its 'Module' is 'Retail and commerce'. To the right of the table, there is a detailed view for the 'Clienteling' feature, including its description and a 'Comments' section. At the bottom right of the feature details, there are 'Schedule' and 'Enable now' buttons.

Enable date	Feature added	Module
10/7/2019	Feature added: "Retail and commerce"	Retail and commerce
10/7/2019	Module: "Retail and commerce"	Retail and commerce
8/31/2019	(India) Search customers in Reta...	Retail and commerce
8/1/2019	Clienteling	Retail and commerce
8/1/2019	Fiscal integration local storage b...	Retail and commerce
7/24/2019	Enable display order for mercha...	Retail and commerce
6/3/2019	(Retail GST for India) Update cre...	Retail and commerce

2. Turn on the **Clienteling** feature by selecting **Enable now**.
3. On the **Commerce Parameters** page, on the **Number sequence** tab, select the **Client book identifier** row. Then, in the **Number sequence code** field, select a number sequence. The system will

use this number sequence to assign an ID to client books.

4. Select **Save**.
5. Create a new attribute group that contains the attributes that you want to capture for customers who are managed in client books. For instructions, see [Attributes and attribute groups](#).
  - Define the required attributes as **Can be refined**. Sales associates can then use these attributes to filter their client book.
  - Set the display order for these attributes. This display order determines which attributes should be shown on the customer card in the client book. A display order of 1 is considered higher than a display order of 2. Therefore, the attribute that has a display order of 1 will be shown before to the attribute that has a display order of 2.

#### NOTE

You can make Customer Insights available from the same page. However, an Azure application ID and secret must be created, for authentication purposes. (For information about the requirements, see the [Turn on the integration of Customer Insights with Commerce](#) section later in this topic.) If Customer Insights is turned on, and you select one or more measures that should be shown on the customer card, those measures will be shown first. Next, client book attribute groups will be shown, based on the display order. For example, if you select two measures from Customer Insights, those two measures and one client book attribute will be shown on the customer card. (The client book attribute that is shown will be the attribute that has the highest display order.)

6. On the **Commerce parameters** page, on the **Clienteling** tab, in the **Client book attribute group** field, select the attribute group that you just created.

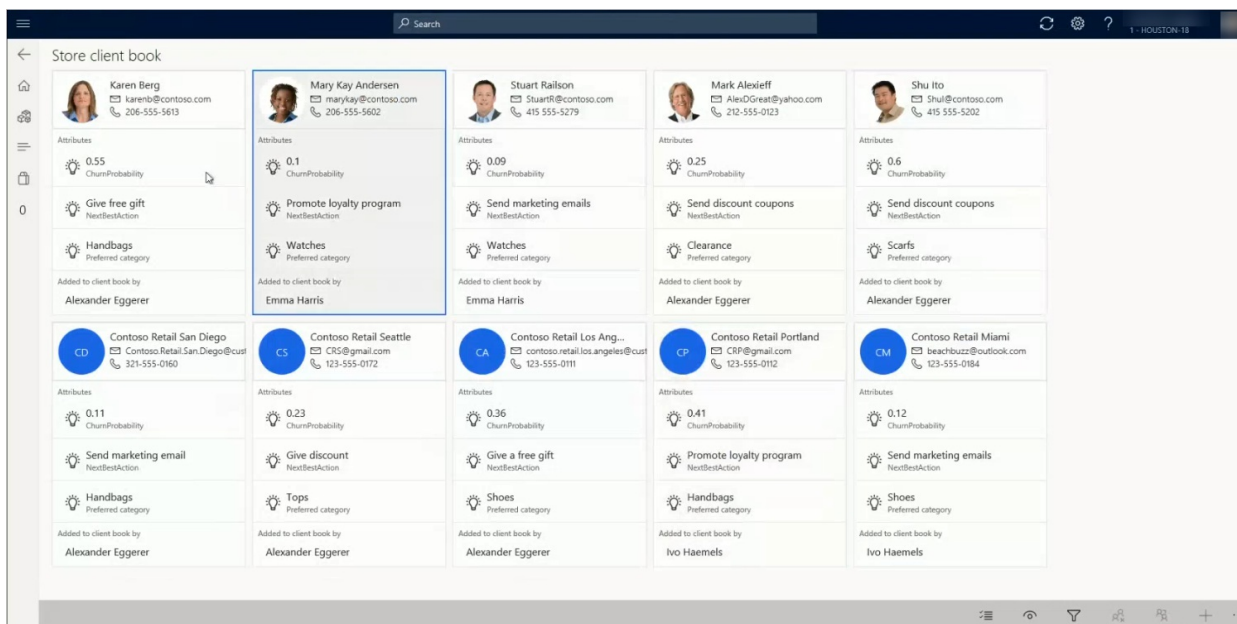
7. To capture activities that occur at the POS, define the activity types on the **Activity types** page (**Retail and Commerce > Customers > Activity types**).

#### NOTE

Activity types are pulled by the Commerce Scale Unit when it makes a real-time call for the first time. After the activities are pulled, they are cached for a few hours. If you change the activity types, wait until the cache is no longer valid. Alternatively, for non-production environments, restart the Commerce Scale Unit service.

8. Add two buttons to the appropriate POS screen layout, so that sales associates can view their own client book and the store client book. (Store client books include clients from all client books of all associates who share an address book with the store.) The corresponding operations are named **View customers in client book** and **View customers from store client books**, respectively. Three additional operations that are related to client books are available. These operations determine which associates can add, remove, and reassign customers from the client book. They are named **Add customer to client book**, **Remove customers from client book**, and **Reassign customers to a client book**, respectively.
9. Run the following distribution schedule jobs: 1040, 1150, 1110, and 1090.

After you've completed this procedure, sales associates can open the customer details page at the POS, and add customers to their client book, view and capture activities and notes for customers, and target customers by using customer and client book attributes to filter the client book. The following illustration shows an example of a client book.



## Turn on the integration of Customer Insights with Commerce

To turn on the integration of Customer Insights with Commerce, you must make sure that you have an active instance of Customer Insights in the tenant where Commerce is provisioned. You must also have an Azure Active Directory (Azure AD) user account that has an Azure subscription.

Follow these steps to set up the integration.

1. In the Azure portal, register a new application and make a note of the application name, application ID, and secret. This information will be used for service-to-service authentication between Commerce and Customer Insights. Note the secret safely, as it will be required to save it in the key vault. For the following example, use `CI_Access_name`, `CI_Access_AppID`, `CI_Access_Secret` for the application name, application ID, and secret respectively. For more information, see [Quickstart: Register an application with the Microsoft identity platform](#).

### IMPORTANT

Take steps so that you will remember to change the secret before it expires. Otherwise, the integration will unexpectedly stop.

2. Go to your Customer Insights instance and search for the name of the application created above (in this example, "CI\_Access\_name").
3. Create an Azure key vault, and take a note of the name and URL (in this example, "KeyVaultName", "KeyVaultURL"). For instructions, see [Quickstart: Set and retrieve a secret from Azure Key Vault using the Azure portal](#).
4. Save the secret (in this example, "CI\_Access\_Secret") in the vault. When this secret is stored in the vault, the secret gets a name. Note the secret name (in this example, 'SecretName').
5. To access the secret from Azure Key Vault, you need to create another application with an application ID and secret (in this example, "KeyVault\_Access\_AppID" and "KeyVault\_Access\_Secret"). Note the secret safely, as it will not be displayed again.
6. Next, you need to give permissions to the application to access the Key Vault from Commerce using APIs. Go to the application page in Azure portal. Under the **Manage** section, select **API permissions**. Add the permission to access **Azure key vault**. For this permission, select **Access policy**. Select the template as **Secret management**, and select the **Get**, **List**, **Decrypt**, and **Encrypt** options.

7. In Commerce headquarters, go to **System administration > Setup > Key Vault parameters**, and enter the required information for the key vault. Then, in the **Key Vault client** field, enter the application ID that you used in step 4, so that Commerce can access the secrets in the key vault.
8. To add the application that you created in step 1 to the list of safe applications (sometimes referred to as a safe list), go to Customer Insights, and select **View** access to the application. For instructions, see [Permissions](#).
9. On the **System administration > Setup > Key Vault parameters** page in Commerce HQ, update the fields as described below:
  - **Key Vault url:** "KeyVaultURL" (from step 3 above).
  - **Key Vault client:** "KeyVault\_Access\_AppID" (from step 5 above).
  - **Key Vault secret key:** "KeyVault\_Access\_Secret" (from step 5 above).
  - Under **Secrets** section:
    - **Name:** Any name, for example "CISecret".
    - **Description:** Any value.
    - **Secret:** `vault:///>` In this example it will be "vault://KeyVaultName/SecretName".

After you update the fields, select **Validate** to ensure the secret can be accessed by the Commerce application.

8. In Commerce, on the **Commerce parameters** page, on the **Clienteling** tab, on the **Dynamics 365 Customer Insights** FastTab, set the **Application ID** to "CI\_Access\_AppID" (from step 1 above). For **Secret name**, select the name of the secret entered in step 7 above ("CISecret"). Set the **Enable Customer Insights** option to **Yes**. If the setup is unsuccessful for any reason, an error message will be displayed, and this option will be set to **No**.

You can have multiple environments in Customer Insights, such as test and production environments. In the **Environment instance ID** field, enter the appropriate environment. In the **Alternate customer ID** field, enter the property in Customer Insights that is mapped to the customer account number. (In Commerce, the customer account number is the customer ID.) The remaining three properties are the measures that will be shown on the customer card in the client book. You can select up to three measures to show on the customer card. However, you are not required to select any measures. As mentioned previously, the system shows these values first, and then it shows the values for the client book attribute group.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Loyalty overview

2/18/2021 • 18 minutes to read • [Edit Online](#)

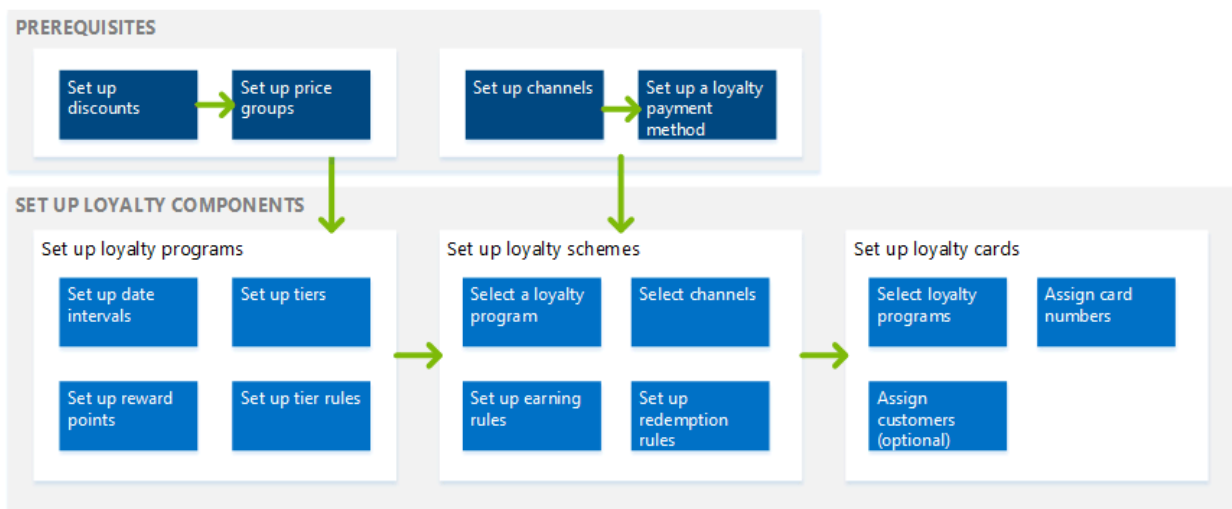
Loyalty programs can help increase customer loyalty by rewarding customers for their interactions with the retailer's brand. In Dynamics 365 Commerce, you can set up simple or complex loyalty programs that apply across your legal entities in any commerce channel. This topic describes the loyalty capabilities within Commerce and the corresponding setup steps to help the retailer easily get started with their loyalty programs.

You can set up your loyalty program so that they include the following options.

- Set up multiple types of rewards that you offer in your loyalty programs, and track participation in your loyalty programs.
- Set up loyalty programs that represent the different reward incentives that you offer. You can include loyalty program tiers to offer greater incentives and rewards to customers who shop more frequently or spend more money in your stores.
- Define earning rules to identify the activities that a customer must complete to earn rewards. You can also define redemption rules to identify when and how a customer can redeem rewards.
- Issue loyalty cards from any channel that participates in your loyalty programs, and link loyalty cards to one or more loyalty programs that the customer can participate in. You can also link a customer record to a loyalty card, so that the customer can pool loyalty points from multiple cards and redeem them.
- Manually adjust loyalty cards, or transfer the loyalty rewards balance from one card to another to accommodate or reward a customer.

## Setting up loyalty programs

You must set up several components to enable the loyalty feature in Commerce. The following diagram illustrates the loyalty components and how they relate to each other.



## Loyalty components

The following table describes each component and where it's used in the loyalty setup.

COMPONENT	DESCRIPTION	WHERE IT'S USED
-----------	-------------	-----------------

COMPONENT	DESCRIPTION	WHERE IT'S USED
Set up discounts (prerequisite)	Set up the discounts that you offer to your loyalty customers. For example, you can offer 5 percent off all apparel products.	Discounts must be added to price groups before they can be included in a loyalty program. Price groups are assigned to loyalty programs and loyalty tiers.
Set up price groups (prerequisite)	Price groups are used to create and manage prices and discounts for products. Set up the price groups that include the discounts that apply to your loyalty programs.	Price groups are assigned to loyalty programs and loyalty program tiers.
Set up channels (prerequisite)	Commerce channels are the stores that participate in your loyalty programs, such as a brick-and-mortar store, an online store, or a call center. You must set up your channels before you can assign loyalty programs to them.	You assign channels to a loyalty program if the channel is participating in the loyalty program.
Set up the loyalty payment method (prerequisite)	To ensure the loyalty points can be redeemed in any channel, such as brick and mortar stores, online stores, or call centers, you must set up the bin range for the loyalty cards on the <b>Card numbers</b> page.	Set up a loyalty type payment method, and then assign the loyalty payment method to the channels that are participating in the loyalty program.
Set up date intervals	Date intervals provide a flexible way to set the time span that applies to loyalty tiers. Use date intervals to specify how long a customer can stay in a tier or how much time a customer has to complete an activity to qualify for a tier.	Date intervals apply only if you use tiers in your loyalty programs. You select the date interval that applies to program tiers, and also the date intervals that apply to program tier rules.
Set up reward points	Reward points are the types of reward that you offer to your customers. Rewards points can be redeemable or non-redeemable. Redeemable reward points can be exchanged for products. Non-redeemable reward points are used for tracking purposes or to advance a customer to the next tier in a loyalty program.	Reward points are referenced in tier rules and are used to qualify a customer for a specific tier. Reward points are also referenced in loyalty schemes in earning and redemption rules. In earning rules, you specify the rewards that a customer can earn for a specific activity. In redemption rules, you specify the reward that the customer can redeem.
Set up loyalty programs	Loyalty programs are the core loyalty entity that you offer. Each loyalty program can also have loyalty tiers assigned to it. Discounts and price groups are assigned to the loyalty programs at either the program level or the tier level.	You create loyalty schemes for your loyalty programs. You assign loyalty cards to your loyalty programs, and loyalty cards can be assigned to a customer. Channels participate in the loyalty programs that are assigned to the loyalty schemes. Any customer who holds a loyalty card can participate in the loyalty programs that are assigned to the card.

COMPONENT	DESCRIPTION	WHERE IT'S USED
Set up loyalty tiers and tier rules	Loyalty tiers are optional levels that you can define for your loyalty programs. You can set up base discounts and rewards for all customers who participate in the loyalty program, and you can set up additional discounts and rewards for customers who achieve the various levels in the program. For each loyalty tier that you define, you can set up the rules that qualify a customer for that tier. You can also define how long customers can remain in that tier after they have reached it.	Loyalty tiers and loyalty tier rules are defined in the loyalty programs. If you don't define any loyalty tiers, all customers who participate in the loyalty program qualify for the discounts that you assign in the loyalty program price group. If you define loyalty tiers, you can set up earning rules and redemption rules for the loyalty tiers in the loyalty scheme.
Set up loyalty schemes	Loyalty schemes specify the earning rules and redemption rules that apply to a selected loyalty program. You assign channels to a loyalty scheme to identify which loyalty program, earning rules, and redemption rules apply to a store.	A loyalty scheme is assigned to a loyalty program and channels. You can assign many loyalty schemes to the same loyalty program, and you can assign many loyalty schemes to many channels.
Set up loyalty cards	A loyalty card entitles the card holder to participate in the loyalty programs that are assigned to the card. Loyalty cards can be issued anonymously, or they can be assigned to a specific customer. You can view the loyalty transactions for a specific card, and you can view a summary of loyalty points that have been accumulated on the card. You can issue loyalty cards from any channel. You can also manually adjust a loyalty card to upgrade the customer to a different tier, add loyalty points, or transfer the loyalty point balance from one card to another.	You assign loyalty programs to a loyalty card.

## Loyalty processes

The following table describes the processes that must be run to send the loyalty configurations and data to your stores, and to retrieve the loyalty transactions from your stores.

PROCESS NAME	DESCRIPTION	PAGE NAME
1050 (loyalty information)	Run this process to send the loyalty data from Commerce to the stores. It's a good idea to schedule this process to run frequently, so that loyalty data is transmitted to all stores.	Distribution schedule



PROCESS NAME	DESCRIPTION	PAGE NAME
Process loyalty schemes	Run this process to associate loyalty schemes with the channels that the loyalty scheme is assigned to. This process can be scheduled to run as a batch process. You must run this process if you change loyalty configuration data, such as loyalty schemes, loyalty programs, or loyalty reward points.	Process loyalty schemes
Post earned loyalty points in batches	Run this process to update loyalty cards so that they include transactions that were processed offline. This process applies only if the <b>Post earned points in batches</b> check box is selected on the <b>Commerce shared parameters</b> page, so rewards can be earned offline.	Post earned loyalty points in batches
Update loyalty card tiers	Run this process to evaluate the customer's earning activity against the tier rules for a loyalty program, and to update the customer's tier status. This process is required only if you change the tier rules in loyalty programs and want the updated rules to be retroactively applied to loyalty cards that have already been issued. This process can be run as a batch process or for individual cards.	Update loyalty card tiers

## Loyalty capabilities

- Using the price groups associated with the loyalty program and loyalty tiers, retailers can easily create special prices and discounts for loyalty members.
- As a part of a loyalty scheme retailers can create different earning and redemption rules by tiers to differentiate the rewards for customers in different tiers. Retailers can also include "affiliations" as a part of the earning and redemption rules so that certain group of customers can be a part of existing tiers, but still be rewarded differently. This prevents the need to create additional tiers.

### NOTE

The earning rules within a loyalty scheme are additional. For example, if you create a rule to reward a gold tier member 10 points for each US dollar, and you also create a rule for a customer with "veteran" affiliation to reward 5 points for each US dollar, then a veteran who is also a gold tier member would earn 15 points for 1 US dollar, as the customer qualifies for both lines. However, if the veteran customer was not a gold tier member, then he would earn 5 points for each dollar. To reflect the changes in the channels, run the **Process loyalty schemes** and **1050** (loyalty information) jobs.

Dynamics 365 Finance and Operations Commerce > Customers > Loyalty > Loyalty schemes USRT

PERSONALIZE Always open for editing Personalize this form Add to workspace

PAGE OPTIONS Security diagnostics Advanced filter or sort Record info Change view

SHARE Get a link Create a custom alert Manage my alerts

LOYALTY SCHEMES **Fabrikam**

General Fabrikam | Fabrikam

Scheme ID: Fabrikam Description: Fabrikam base scheme Loyalty program: Fabrikam rewards

Earning rules

+ Add line Remove

Program tier	Affiliation	Activity type	Category	Product	Product vari...	Activity amount...	Activity currency	Reward point ID	Reward point type	Reward points
	Veteran	Purchase products by amount	Fashion			1.0000	USD	Fabrikam rewards	Amount	5.0000
		Purchase products by amount	Fashion			0.0100	USD	Total spent	Amount	0.0100
		Purchase products by quantity	Mens Shoes			1.0000		Fabrikam rewards	Amount	25.0000
		Sales transaction count				1.0000		Total transactions	Quantity	1.0000
Gold		Purchase products by amount	Fashion			1.0000	USD	Fabrikam rewards	Amount	10.0000

- Retailers often have special prices for a certain group of customers that they don't want loyalty programs applied to. For example, wholesalers or employees who get special pricing and no loyalty points. Commonly, "affiliations" are used to provide the special pricing to such customer groups. To restrict certain customer groups of customers from earning loyalty points, the retailer can specify one or more affiliations under the **Excluded affiliations** section of the loyalty scheme. That way, when customers belonging to excluded affiliations are existing loyalty members, they won't be able to earn loyalty points for their purchases. To reflect the changes in the channels, run the **Process loyalty schemes and 1050** (loyalty information) jobs.

Dynamics 365 Finance and Operations Commerce > Customers > Loyalty > Loyalty schemes

Save + New Delete OPTIONS

PERSONALIZE Always open for editing Personalize this form Add to workspace

PAGE OPTIONS Security diagnostics Advanced filter or sort Record info Change view

EDIT Read mode Revert

SHARE Get a link Create a custom alert Manage my alerts

LOYALTY SCHEMES **Fabrikam**

Earning rules

Redemption rules

Channels

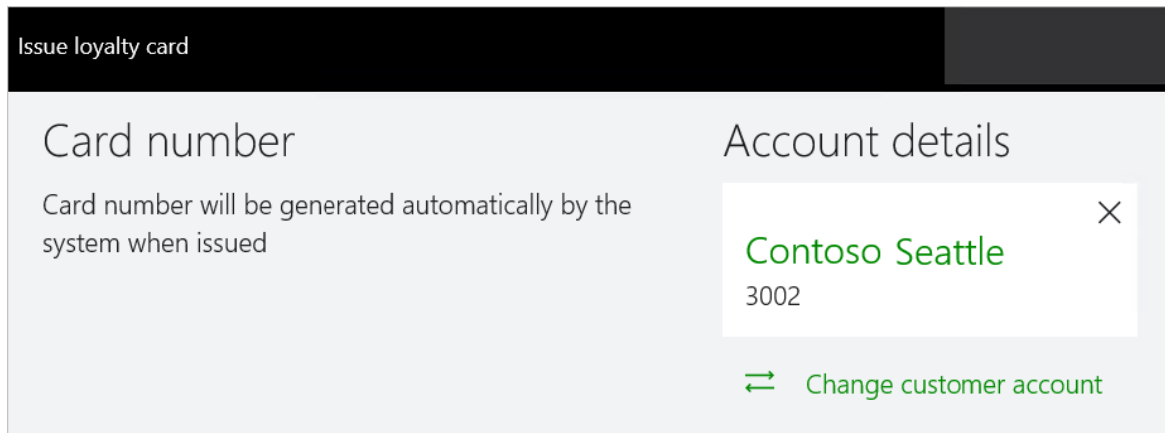
**Excluded affiliations**

+ Add line Remove

Affiliation	Description
Employee	Employee

- The Point of Sale allows the flexibility for retailers to either use the physical loyalty cards or generate a

unique loyalty card number automatically. To enable the automatic generation of loyalty cards in the stores, turn on **Generate loyalty card number** in the functionality profile associated to the store. For online channels, retailers can use the IssueLoyaltyCard API to issue loyalty cards to customers. Retailers can either provide a loyalty card number to this API, which will be used to generate the loyalty card, or the system will use the loyalty card number sequence set in Commerce. However, if the number sequence is not present, and the retailer does not provide a loyalty card number while calling the API, then an error is displayed.



Issue loyalty card

Card number

Card number will be generated automatically by the system when issued

Account details

Contoso Seattle  
3002

↔ Change customer account

- Earned and redeemed loyalty points can now be saved for each transaction and sales orders against the sales line, so that the same amount can be refunded or taken back in the case of full or partial returns. Moreover, having the visibility for points at the sales line level provides the capability for call center users to answer customer questions on how many points were earned or redeemed for each line. Prior to this change, reward points were always recalculated during returns, which resulted in a different amount than the original, if the earning or redemption rules were changed and also the call center users did not have the visibility on the points breakdown. The points can be viewed under the **Card transactions** form for each loyalty card. To enable this feature turn on the configuration **Post loyalty points per sales line** under **Commerce shared parameters > General** tab.

#### NOTE

We strongly recommend turning this feature on to ensure, in case of returns, the correct amount of points can be refunded or taken from the customer.

- Retailers can now define the vesting period for each reward point. A vesting period configuration will define the duration from the earn date, after which the reward points would become available to the customers. Unvested points can be viewed in the **Unvested points** column on the **Loyalty cards** page. When the customers return some items for which the loyalty points were earned, then by default, the system will deduct the unvested points first and then deduct any balance from the available points. However, you can configure to deduct the available points only instead of deducting from unvested points.

Additionally, retailers can define the maximum loyalty reward point limit per loyalty card. This field can be used to reduce the impact of loyalty fraud. When the maximum award points have been reached, the user cannot earn more points. The retailer can decide to block such cards until they have investigated for potential fraud. If the retailer determines fraud, the retailer can block the loyalty card for the customer and mark the customer as blocked. To do this, set the **Block customer for loyalty enrollment** property to **Yes** under **All customers** on the **Commerce** FastTab. The blocked customers will not be able to be issued a loyalty card in any of the channels.

Dynamics 365 Finance and Operations Commerce > Customers > Loyalty > Loyalty reward points USRT

Edit + New Delete Translations OPTIONS

LOYALTY REWARD POINTS

Filter

Fabrikam rewards Fabrikam fashion rewards

Loyalty amount spent Total amount spent for Contoso and AW

Loyalty rewards Contoso and Adventure works loyalty poi...

Total spent Total amount spent

Total transactions

General

Currency USD

REDEMPTION Redeemable Yes

Redeem ranking 0

EXPIRATION Expiration time value 1 Expiration time unit Years

VESTING Vesting time value 0 Vesting time unit Day

Maximum reward points per loyalty card 0

- Affiliations are used to provide special pricing and discounts, but there be some affiliations that retailers do not want their customers to see. For example, an affiliation titled "High spend customer" might not be well received by some customers. Moreover, there are some affiliations that should not be managed in the store, for example, employees, because you do not want the cashiers to decide who is an employee and thus provide employee-based discounts. Retailers can now select the affiliations which should be hidden in the channels. Affiliations marked as **Hide in channels** cannot be viewed, added, or removed in the POS. However, the pricing and discounts associated with the affiliation will still be applied to the products.

Dynamics 365 Finance and Operations Commerce > Customers > Affiliations

Save + New Delete Translations Info codes Price groups OPTIONS

AFFILIATIONS

Filter

Name ↑	Description	Hide in channels
HNI	High spending customer	Yes
Veteran	Members of Military services	No
Student		No
Vendor	Vendor affiliation	No

- Call center users can now more easily search for a customer using their loyalty card information, and navigate to the customer's loyalty card and loyalty card transaction pages from the **Customer service** page.

Dynamics 365 Finance and Operations Commerce > Customers > Customer service

USRT

New customer New sales order Return order CUSTOMER VIEW OPTIONS

DETAILS Other address Customer details Contact details Related orders Backorder lines

OPTIONS Notes Return inquiry Source codes Customer item statistics

CATALOG Send catalog

CASES View customer cases Create case

JOURNALS Invoice journal

EMAIL NOTIFICATION Email notification log

LOYALTY Loyalty cards Loyalty card transactions

CUSTOMER SERVICE

2001 - Karen Berg

Keyword 2001 Search

Customer information USD

Name: Karen Berg Address: 712 1st Ave SW, Kirkland, WA 98007, USA Customer account: 2001 Prospect Telephone: Extension: Email: karen@contoso.com Currency: USD

Notes

Sales orders 012555 Karen Berg

Options Details Sales tax Detailed status Invoice Order history Tracking details Payments

Sales order	Status	Payment status	Payment out of balance	Channel	Created date	Ship date	Order type
012555	Open order	Paid	None	Fabrikam call center	10/12/2018	10/12/2018	Sales order
012554	Open order	Paid	None	Fabrikam call center	10/12/2018	10/12/2018	Sales order

- If a loyalty card is compromised, a replacement card needs to be generated and the existing points transferred to the new card. The replacement card flow has been simplified in this release. Additionally, customers can gift some or all of their loyalty points to friends and family. When points are transferred, points adjustment entries are created for each loyalty card. The replacement card and transfer balance functionality can be accessed from the Loyalty cards page.

Dynamics 365 Finance and Operations Commerce > Customers > Loyalty > Loyalty

Edit + New Delete Card adjustments Replace card Transfer points Card transactions OPTIONS

PERSONALIZE Always open for editing Personalize this form Add to workspace

PAGE OPTIONS Security diagnostics Record info Advanced filter or sort Change view

SHARE Get a link Create a custom alert Manage my alerts

LOYALTY CARDS Filter

Card number	Card type	Customer name
100002	As card tender	Олег Евгеньевич Зубарев
55101	As contact tender	
55102	As contact tender	Stuart Railson
55103	As card tender	Karen Berg
55104	As card tender	Mark Alexieff
55105	As card tender	Karen Berg

Transfer points

Cards

Source loyalty card number: 55103 Destination loyalty card number: 55102

Points available to transfer

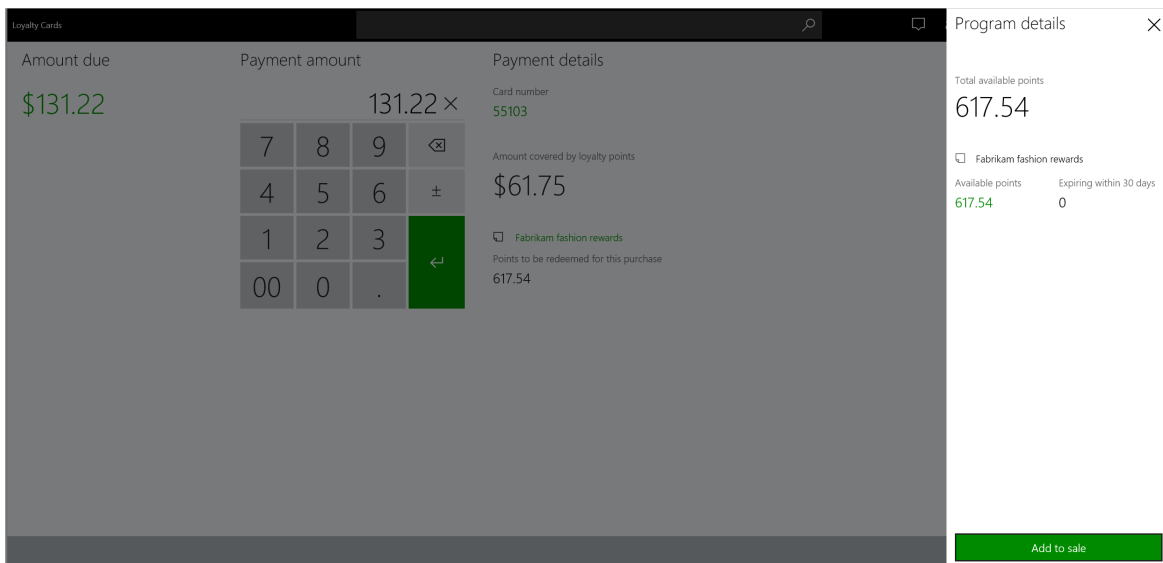
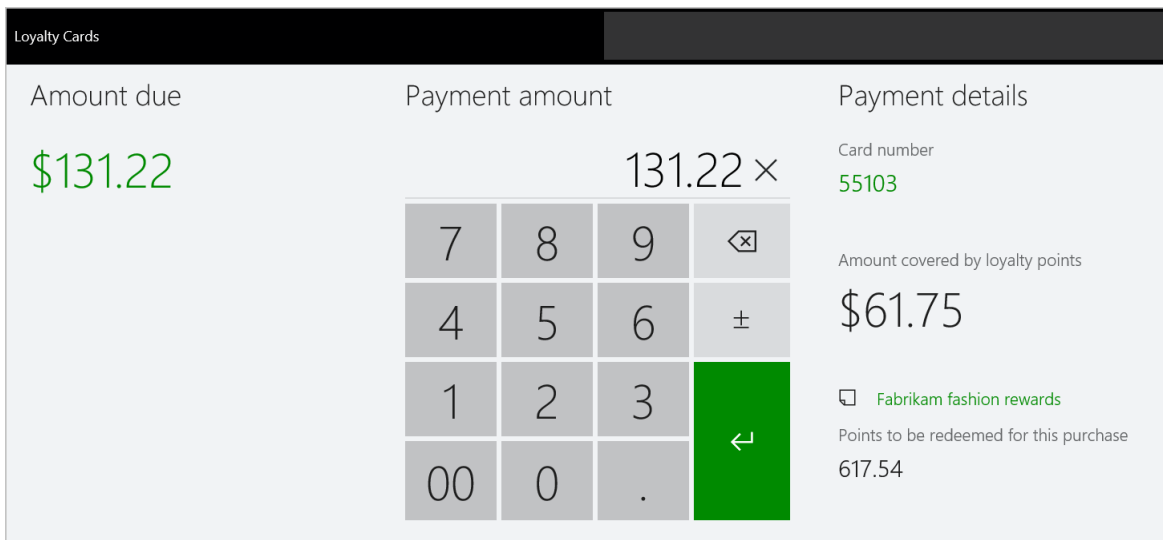
Transfer all points: No

Fabrikam rewards Available points: 324.00 Points to transfer to destination: 0.00

Loyalty rewards Available points: 0.00 Points to transfer to destination: 0.00

OK Cancel

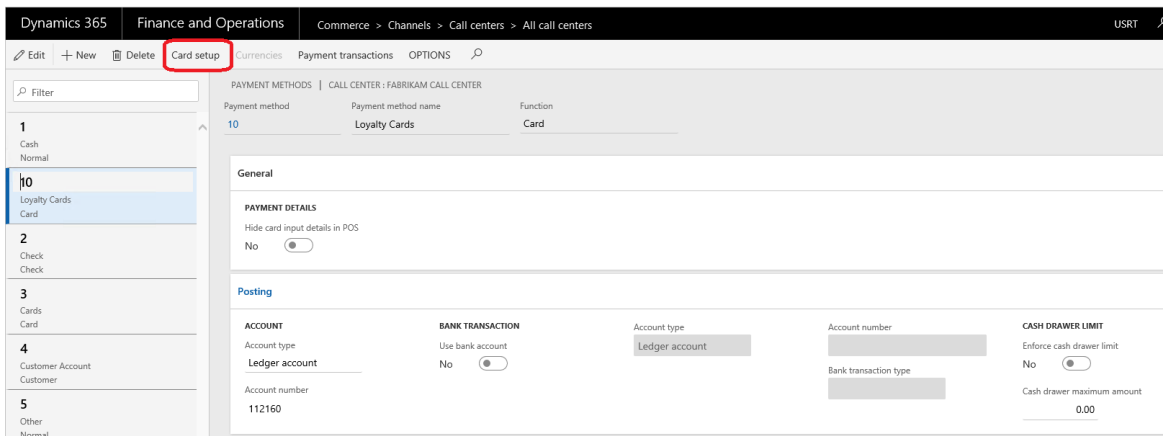
- Retailers may want to capture the effectiveness of a particular channel to enroll customers into a loyalty program. The enrollment source for the loyalty cards is now saved so that retailers can run reports on this data. The enrollments source is automatically captured for all the issued loyalty cards from MPOS/CPOS or e-Commerce channels. For the loyalty cards issued from the back office application, the call center user can select an appropriate channel.
- In earlier releases, retailers could use MPOS/CPOS to redeem loyalty points for customers in a store. However, in those releases, because the loyalty balance is displayed in loyalty points, the cashier could not view the currency value amount that could be applied toward the current transaction. The cashier had to do the points to currency conversion before paying by loyalty points. In the current release, after lines are added to the transaction the cashier can see the amount that the loyalty points can cover for the current transaction, making it easy to apply some or all of the loyalty points to the transaction. Additionally, the cashier can see the points that will be expiring in next 30 days, so they can upsell or cross-sell to motivate the customer to spend the expiring points at that transaction.



- With the 8.1.3 release, we have enabled the "pay by loyalty" option in the call center channel. To enable this option, create a loyalty tender type and associate it with the call center.

**NOTE**

Because the loyalty payments are set up as card payments, you will have to select a card from the **Card setup** page.



After this is set up, customers can redeem their loyalty points in the call center. Additionally, we are enhancing the user experience further to show the "Amount covered by loyalty points", so that the call center users do not have to navigate to a different screen to view the loyalty balance.

- Many retailers award loyalty points only based on the sales transactions, but the more customer-centric retailers want to reward their customers for any of their engagement activity with their brand. For example, they want to provide rewards for completing an online survey, visiting a store, liking the retailers on Facebook, or tweeting about the retailer. To do this, the retailer can define any number of "Other activity type" and define the corresponding earning rules for these activities. There is also an exposed Commerce Scale Unit API "PostNonTransactionalActivityLoyaltyPoints" that can be called when an activity is identified that should reward the customer with loyalty points. This API expects the Loyalty card ID, Channel ID, and the Other Activity Type ID, so that the customer who should be rewarded can be located and the earning rule for the activity can be identified.

Awarding points for non-transaction activities generally has two major steps:

- Realizing an activity has occurred that should be rewarded.
- Rewarding the appropriate points.

The first step is external to Commerce, such as tweeting about the brand or liking the brand on Facebook. After this activity has been recognized, the retailers can call the above-mentioned Commerce Scale Unit API and award loyalty points in real time. In such scenarios, there is no need for a review step because an activity has occurred and corresponding points should be awarded. However, there are scenarios where the retailer would want to review the records prior to awarding the points. For example, the retailer has set up a workshop in the store for which the customers sign up on the ecommerce website or any other event registering application. However, only the attending customers should earn loyalty points. For such scenarios, in the 10.0 release, we introduced a data entity named **Retail loyalty other activity type lines**. This data entity enables the retailers to use either Data Import/Export Framework (DIXF) or OData API to record the activities that should award customers with loyalty points. The data entity stores the activities in a journal named **Loyalty lines for other activities**, which can be used for review and modification purposes. After the data has been reviewed, the IT user can either manually post the activity lines or run a job named **Process other activity type for loyalty lines**, which will post all the unposted activity lines and award the points to the customers based on the earning rules. In the above scenario, the event registration application would call OData API to send the customer information to Commerce. However, the IT user can post the activity lines for only those customers who attended the workshop and delete the activity lines for the other customers.

#### NOTE

Currently, the system forces users to set up a number sequence for "other activity types", but this will not be a required step in future releases. To set up a number sequence, go to **Commerce shared parameters > Number sequences** and select a number sequence for **Loyalty other activity type ID**.

- To provide good customer service and effectively resolve customer queries, it is important for the cashiers to have access to complete customer's profile. With the 10.0 release, cashiers will be able to see loyalty history details along with the associated loyalty program and tier information on POS.
- Free or discounted shipping is one of the highly motivating factors for customers to buy online. To enable the retailers to set up shipping promotions, with the 10.0 release, we have introduced a new type of promotion named "Shipping threshold discount," where the retailer can define the thresholds, which once met, will qualify the customers for discounted or free shipping. For example, spend \$35 for free 'Two day ship' or Free 'Two day ship' for all loyalty customers. This feature leverages the new Advanced auto charges capability. Refer the [documentation on Advanced auto charges](#). These advanced auto charges need to be enabled for shipping promotion to work. These can be enabled from the **Customer orders** tab on the **Commerce parameters** page and turn on the "Use advanced auto-charges" configuration. Additionally, because a retailer can set up multiple types of charges, such as handling or installation, the retailer needs to specify which charge is considered shipping charge. The shipping discounts are only applied to the shipping charges. To specify charge as Shipping charge, navigate to the

**Charge codes** form present under **Retail and Commerce > Retail and Commerce IT > Channel setup > Charges** and turn on the "Shipping charge" checkbox for the desired charges. Now you can navigate to the **Shipping threshold discount** form and set up the discount.

Like product discounts, this discount honors all the existing standard discount capabilities, such as allowing the retailer to restrict these discounts with the coupons so that only the customers with coupons can get these discounts. Also, these discounts leverage the Price groups capability to determine the eligibility of such discounts. For example, the retailer can choose to run these promotions only in the online channels and/or across channels for certain customer groups such as loyalty customers. After the order lines with the specified delivery mode meets the defined threshold, then the shipping discount gets applied and reduces the shipping charge based on the discount set up.

**NOTE**

Unlike other periodic discounts such as quantity, simple, mix and match, and threshold discounts, the shipping discount does not create discount lines, rather edits the shipping charge directly and appends the name of the discount to the charge description.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Define loyalty programs

2/18/2021 • 3 minutes to read • [Edit Online](#)

This procedure shows how to set up a loyalty program with two loyalty tiers. This procedure uses the USRT demo data company.

1. Go to Retail and Commerce > .. > Loyalty programs.
2. Click New.
3. In the Name field, type a value.
4. In the Description field, type a value.
5. Click Add line.
6. In the Level field, enter a number.
7. In the Tier field, enter a name for the loyalty tier.
8. In the Description field, type a value.
9. In the Date interval field, click the drop-down button to open the lookup.
  - This date interval should extend into the future. For example, if the date interval that is selected for gold tier is a one-year period, any customer who qualifies for the gold tier can receive the rewards that are assigned to the gold tier for one year. If the customer re-qualifies for the gold tier while they are in the tier, the date that the tier expires is extended by another year from the date when they re-qualify.
10. In the list, click the link in the selected row.
11. Click Add line.
12. In the Level field, enter a number.
13. In the Tier field, enter a name for the loyalty tier.
14. In the Description field, type a value.
15. In the Date interval field, click the drop-down button to open the lookup.
16. In the list, click the link in the selected row.
17. Click Save.
18. In the list, find and select the desired record.
  - Tier rules define the minimum number of a reward point needed to be earned during a time period to qualify for the tier.
19. Toggle the expansion of the Tier rules section.
20. Click New.
  - You can have more than one tier rule per tier. For example, you could have three different criteria to earn a tier; by spending \$500 in one month, by spending \$1000 over one year, and by having 20 transactions in one year. To do this, you would need to create three tier rules.
21. In the Reward point field, click the drop-down button to open the lookup.
  - This should be a non-redeemable loyalty reward point.
22. In the list, click the link in the selected row.
23. In the Minimum issued points field, enter a number.
  - For the lowest level tier, if all customers qualify simply by participating in the program, enter '0'.
24. In the Evaluation date interval field, click the drop-down button to open the lookup.
  - This date interval should extend into the past. Only points earned during this date interval will be counted towards reaching the minimum issued points value.
25. In the list, click the link in the selected row.

26. Click Save.
27. In the list, find and select the desired record.
28. Click New.
29. In the Reward point field, click the drop-down button to open the lookup.
30. In the list, click the link in the selected row.
31. In the Minimum issued points field, enter a number.
32. In the Evaluation date interval field, click the drop-down button to open the lookup.
  - This date interval should extend into the past.
33. In the list, click the link in the selected row.
34. Click Save.
35. Click Price groups.
  - If you want to give loyalty customers discounts, you'll need to assign one or more price groups to the loyalty program and assign the price groups to discounts. It is a best practice to not mix price groups across different types of discounting entities. For example, don't use the same price group for a loyalty discount and a channel discount.
36. In the Price group field, click the drop-down button to open the lookup.
  - The Price groups link at the top of the page is for the loyalty program. The Price groups link in the Program tiers FastTab is for a specific loyalty tier only.
37. In the list, click the link in the selected row.
38. Click Save.
39. Close the page.
40. Click Save.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Define loyalty reward points

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through defining loyalty reward points. You should set up loyalty reward points before you set up a loyalty program. This procedure uses the USRT demo data company.

1. Go to Retail and Commerce > Customers > Loyalty > Loyalty reward points.
2. Click New.
3. In the Reward point ID field, type a value.
4. In the Description field, type a value.
5. In the Reward point type field, select an option.
  - Select Quantity if you want the reward points to be rounded to the nearest integer. Select Amount if you want the reward points to be rounded according to currency rounding rules. If you select Quantity, skip the next step of this procedure..
6. In the Currency field, type a value.
  - For Amount type reward points, all points issued will have the selected currency. For Quantity type reward points, this field doesn't apply—skip this step.
7. Check or uncheck the Redeemable checkbox.
8. In the Redeem ranking field, enter a number.
  - Redeem ranking is used when two or more redeemable reward points can be used to pay for products. If the two reward points have the same redeem ranking, then the one that needs to lower number of points will be used.
9. In the Expiration time value field, enter a number.
  - The reward points will expire the specified number of days, months, or years after when the points are issued. A value of '0' means the loyalty reward points will never expire.
10. In the Expiration time unit field, select an option.
11. Click Save.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Define loyalty schemes

2/18/2021 • 3 minutes to read • [Edit Online](#)

This procedure walks through how to define a loyalty scheme. Loyalty schemes are reward earning and redeeming rules for a loyalty program. This procedure uses the USRT demo data company.

1. Go to Retail and Commerce > Customers > Loyalty > Loyalty schemes.
2. Click New.
3. In the Scheme ID field, type a value.
4. In the Description field, type a value.
5. In the Name field, click the drop-down button to open the lookup.
  - You can have multiple loyalty schemes for a loyalty program. Loyalty schemes can be for all channels or only a sub-set of channels.
6. In the list, find and select the desired record.
7. In the list, click the link in the selected row.
8. Click Save.
9. Click Add line.
10. In the Activity type field, select an option.
  - Select Purchase products by amount if you want customers to earn rewards based on how much they spend. Select Purchase products by quantity if you want customers to earn rewards based on how many products they buy. Select Sales transaction count if you want customers to earn rewards for each sales transaction, regardless of what or how much is purchased.
  - Select a category. The category will limit which products this earning rule applies to.
  - If you want the earning rule to apply to all products, leave this field blank.
11. In the Activity amount/quantity field, enter a number.
  - For activity type Sales transaction count, you should always use a value of '1.0'. For activity types of Purchase by amount or Purchase by quantity, any transaction that is less than the value entered will not trigger the earning rule. For example, if the activity type is Purchase by amount, and you enter '10.00', then a sales transaction for '9.00' will not earn rewards for this earning rule.
12. In the Activity currency field, type a value.
13. In the Reward point ID field, click the drop-down button to open the lookup.
14. In the list, click the link in the selected row.
15. In the Reward points field, enter a number.
  - Amount type reward points will record earned amounts with decimals. For example, if the earning rule states 1 reward point earned for every 1 Canadian Dollar spent, and the customer spends 1.25 Canadian Dollars, then the customer will earn 1.25 reward points. Quantity type reward points will record earned amounts in integers. Using the example where the earning rule states 1 reward point earned for every 1 Canadian Dollar spent, and the customer spends 1.25 Canadian Dollars, then the customer will earn 1.0 reward points.
16. Click Save.
17. Click Add line.
  - Redemption rules are used when the loyalty payment method is used.
18. In the Reward point ID field, click the drop-down button to open the lookup.
  - Only redeemable reward points are shown.
19. In the list, click the link in the selected row.
20. In the Reward points field, enter a number.

21. In the Redemption type field, select an option.

- Selecting Payment by amount causes the Amount or quantity field to be treated as a currency value. In this case, the amount of reward points used per currency unit of payment is a fixed ratio. Selecting Payment by quantity causes the Amount or quantity field to be treated as a quantity value. In this case, the amount of reward points used per item quantity is a fixed ratio, however, the amount in currency can vary if the price of items paid for with loyalty reward points varies for the same quantity. Redemption type of Loyalty points discount is only valid when the 'Russia' Country/Regional specific features configuration key is enabled, and the POS functionality profiles has an ISO code of 'RU'.
- Select a category. The category will limit which products this redemption rule applies to.
- If you want the redemption rule to apply to all products, leave this field blank.

22. In the Amount or quantity field, enter a number.

23. In the Currency field, type a value.

24. Click Save.

25. Click Add line.

- Select one or more nodes from the Available organization nodes list and move them to the Selected organization nodes list by clicking the arrow between the two lists.

26. Click OK.

27. Click Save.

- Anytime you change the channels for a loyalty scheme, you must run Process loyalty schemes. That way, the channels will get updated loyalty schemes.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Process loyalty reward point adjustments

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates how to look up loyalty card information and adjust loyalty reward points. The demo data company used to create this task is USRT. This task is intended for the Commerce operations manager role or a Customer service manager role.

1. Go to Loyalty cards.
2. In the list, find and select the desired record.
3. In the list, click the link in the selected row.
4. Click Card transactions.
  - On this page you can view all loyalty transactions for the selected loyalty card.
5. Close the page.
6. Click Card adjustments.
7. Click New.
8. In the Reward point field, enter or select a value.
9. In the Amount or quantity field, enter a number.
  - You can add or remove points from the loyalty card by using positive or negative amounts.
10. In the Loyalty program field, enter or select a value.
11. In the Comment field, type a value.
12. Click Post adjustment.
13. Click Yes.
14. Close the page.
  - Normally at this point you'd refresh the page to see the result of the reward points adjustment in the Reward point summary tab. But if you are running this as a task guide, don't refresh now because if you do, the task guide will stop.
15. Click Card transactions.
16. Close the page.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

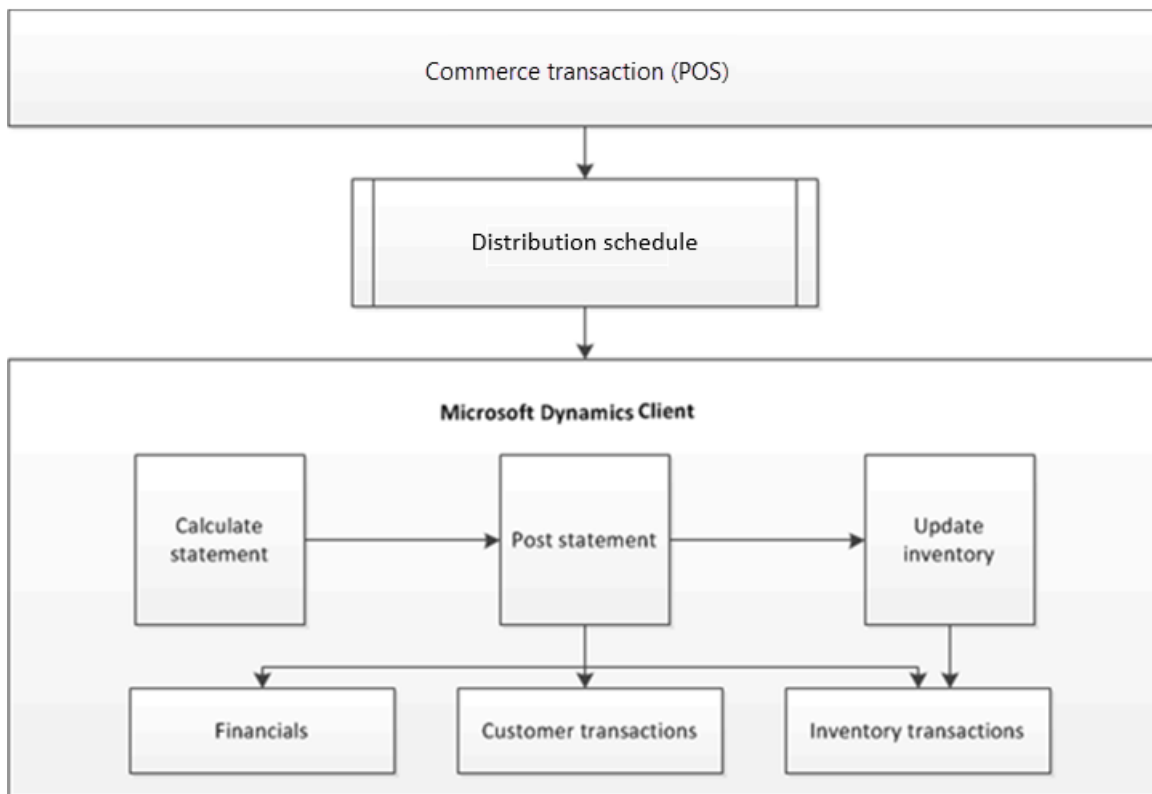
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail statements

2/18/2021 • 4 minutes to read • [Edit Online](#)

In Dynamics 365 Commerce, the statement posting process is used to account for the transactions that occur in Cloud point of sale (POS) or Modern POS (MPOS). The statement posting process uses the distribution schedule to pull a set of POS transactions into the headquarters (HQ) client. The parameters that are defined on the **Commerce parameters** and **Stores** pages are used to select the transactions that are pulled into individual statements.

The following illustration shows the statement posting process. In this process, transactions that are recorded in the POS are transmitted to the client by using the Commerce scheduler. After the client receives the transactions, you can create, calculate, and post the transaction statement for the store.



## Creating and posting statements

You can create a statement manually or by using batch processes that you set up to run periodically throughout the day. In both cases, the following steps are used to create and post statements.

### Create the statement

This step identifies the store that the statement is manually created for. If you configure a batch process, you can automatically create statements for all stores, based on a schedule that you define.

### Calculate the statement

In this step, the transaction lines are selected based on criteria that are defined for each store on the **Commerce parameters** and **Stores** pages. On these pages, you define the criteria and specify how the transactions are calculated. To view a list of the transactions that are included in the statement before you calculate the statement, use the **Transactions** page.

Statement calculation uses tender declarations from the registers as the counted amount. Alternatively, you can

enter the counted amount manually. The statement shows the difference between the sales amount for the transactions and the actual counted amount in all payment methods. The statement is posted only if this difference is less than the maximum posting difference that is defined for the store.

#### NOTE

The statement calculation process uses the global number sequence.

When you calculate a statement, the calculation includes the following tasks:

- For the selected date range, mark transactions that weren't included in a previous statement calculation.
- Calculate the total amounts that were tendered in the selected transactions. The results are shown on the statement lines, depending on the statement method:
  - If the statement method is **Total**, a line is created for each payment method in the selected transactions.
  - If the statement method is **Staff**, a line is created for each payment method in transactions that were performed by the selected staff member.
  - If the statement method is **POS terminal**, a line is created for each payment method in transactions that were performed on the selected register.
  - If the statement method is **Shift**, a line is created for each payment method in transactions that were performed during a shift.

If the **Split by Statement method** check box is selected on the **Stores** page, a separate statement is created based on the value that is selected in the **Statement method** field.

If your store's operating hours extend past midnight, you can configure statement posting so that it's based on the end of the business day instead of the end of the calendar day.

On the **Stores** page, on the **Statement/closing** FastTab, in the **End of business day** field, enter the time that the last transaction must be recorded to be included in the business day's statement. Select the **Post as business day** check box to post the transactions within the same business day. When the statement is posted, transactions that are recorded within the same business day can be included on the same sales order, even if some transactions occur before midnight and other transactions occur after midnight.

#### **Example: Post a statement for a business day that extends over two calendar days**

A store is open between 8:00 AM and 3:00 AM, and the **Post as business day** check box is selected in the store's configuration. On May 31, the store records transactions between 8:00 AM and midnight. The store also records transactions between 12:01 AM and 3:00 AM on June 1.

When the store posts its statement for the close of the business day, the sales order that is generated includes all transactions that were recorded between the business hours of 8:00 AM and 3:00 AM, even though the transactions occurred on two days, May 31 and June 1.

If the **Post as business day** check box is cleared for the same store, separate sales orders are generated when the store posts its statement for the close of the business day. One sales order includes the transactions that were recorded between the business hours of 8:00 AM and midnight on May 31, and the second sales order includes the transactions that were recorded between the business hours of 12:01 AM and 3:00 AM on June 1.

#### NOTE

Before you can create statements, you should close the shifts in the statement period.

## Post the statement



When you post a statement, sales orders and invoices are created for the sales in the statement.

- Cash and carry sales are aggregated onto one sales order, and are invoiced for the default customer who is assigned to the store.
- Sales for which a customer was added to the transaction in POS generate separate sales orders and invoices, one for each unique customer.

Payment journals are automatically created for the payments in the statement, and the inventory is updated for the POS store.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail transaction consistency checker

2/18/2021 • 5 minutes to read • [Edit Online](#)

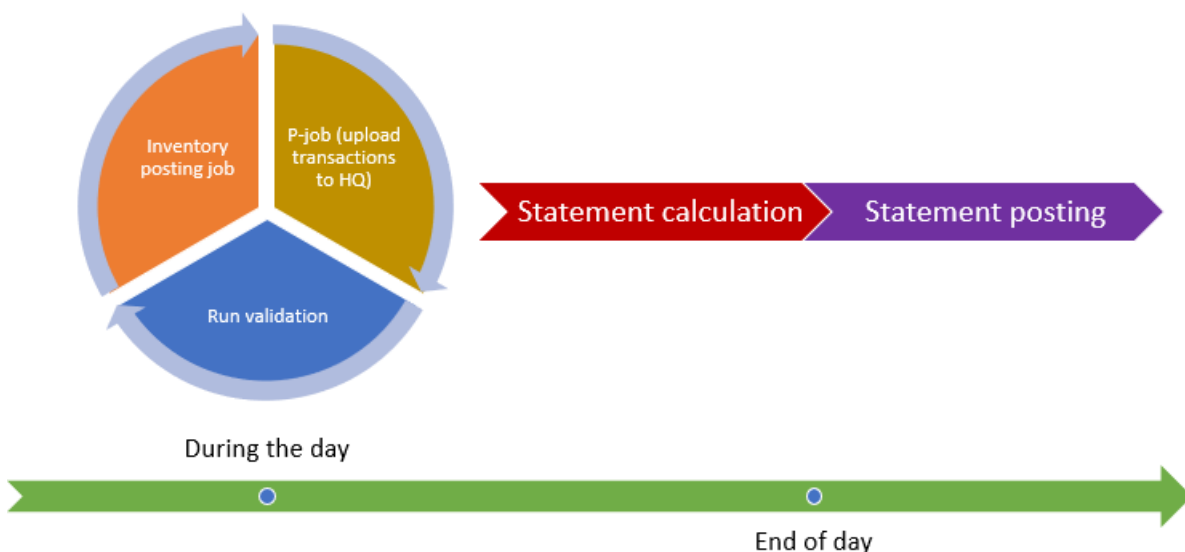
This topic describes the transaction consistency checker functionality in Microsoft Dynamics 365 Commerce. The consistency checker identifies and isolates inconsistent transactions before they are picked up by the statement posting process.

When a statement is posted, posting can fail due to inconsistent data in the commerce transaction tables. The data issue may be caused by unforeseen issues in the point of sale (POS) application, or if transactions were incorrectly imported from third-party POS systems. Examples of how these inconsistencies may appear include:

- The transaction total on the header table does not match the transaction total on the lines.
- The line count on the header table does not match with the number of lines in the transaction table.
- Taxes on the header table do not match the tax amount on the lines.

When inconsistent transactions are picked up by the statement posting process, inconsistent sales invoices and payment journals are created, and the entire statement posting process fails as a result. Recovering the statements from such a state involves complex data fixes across multiple transaction tables. The transaction consistency checker prevents such issues.

The following chart illustrates the posting process with the transaction consistency checker.



The **Validate store transactions** batch process checks the consistency of the commerce transaction tables for the following scenarios.

- **Customer account** – Validates that the customer account in the transaction tables exists in the HQ customer master.
- **Line count** – Validates that the number of lines, as captured on the transaction header table, matches the number of lines in the sales transaction tables.
- **Price includes tax** – Validates that the **Price includes tax** parameter is consistent across transaction lines and the price on the sales line is in accordance with price includes tax and tax exempt configuration.
- **Payment amount** - Validates that the payment records match the payment amount on the header, while also factoring in the configuration for penny rounding in General Ledger.
- **Gross amount** – Validates that the gross amount on the header is the sum of the net amounts on the lines plus the tax amount, while also factoring in the configuration for penny rounding in General Ledger.

- **Net amount** – Validates that the net amount on the header is the sum of the net amounts on the lines, while also factoring in the configuration for penny rounding in General Ledger.
- **Under / Over payment** – Validates that the difference between the gross amount on the header and the payment amount doesn't exceed the maximum underpayment/overpayment configuration, while also factoring in the configuration for penny rounding in General Ledger.
- **Discount amount** – Validates that the discount amount on the discount tables and the discount amount on the transaction line tables are consistent, and that the discount amount on the header is the sum of the discount amounts on the lines, while also factoring in the configuration for penny rounding in General Ledger.
- **Line discount** – Validates that the line discount on the transaction line is the sum of all the lines in the discount table that corresponds to the transaction line.
- **Gift card item** – Commerce doesn't support the return of gift card items. However, the balance on a gift card can be cashed out. Any gift card item that is processed as a return line instead of a cash-out line fails the statement posting process. The validation process for gift card items helps guarantee that the only return gift card line items on the transaction tables are gift card cash-out lines.
- **Negative price** – Validates that there are no negative price transaction lines.
- **Item & Variant** – Validates that items and variants on the transaction lines exist in the item and variant master file.
- **Tax amount** - Validates that tax records match the tax amounts on the lines.
- **Serial number** - Validates that the serial number is present in the transaction lines for items that are controlled by serial number.
- **Sign** - Validates that the sign on the quantity and the net amount are the same in all the transaction lines.
- **Business date** - Validates that the financial periods for all the business dates for the transactions are open.
- **Charges** - Validates that the header and line charge amount is in accordance with price, including tax and tax exempt configuration.

## Set up the consistency checker

Configure the "Validate store transactions" batch process, at **Retail and Commerce > Retail and Commerce IT > POS posting**, for periodic runs. The batch job can be scheduled based on store organization hierarchy, similar to how the "Calculate statement in batch" and "Post statement in batch" processes are set up. We recommend that you configure this batch process to run multiple times in a day and schedule it so that it runs at the end of every P-job execution.

## Results of validation process

The results of the validation check by the batch process are tagged on the appropriate transaction. The **Validation status** field on the transaction record is either set to **Successful** or **Error**, and the date of the last validation run appears on the **Last validation time** field.

To view more descriptive error text relating to a validation failure, select the relevant store transaction record and click the **Validation errors** button.

Transactions that fail the validation check and transactions that have not yet been validated will not be pulled into statements. During the "Calculate statement" process, users will be notified if there are transactions that could have been included in the statement but weren't.

If a validation error is found, the only way to fix the error is to contact Microsoft Support. In a future release, capability will be added so that users can fix the records that failed through the user interface. Logging and auditing capabilities will also be made available to trace the history of the modifications.

**NOTE**

Additional validation rules to support more scenarios will be added in a future release.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Disable rules in the retail transaction consistency checker

2/18/2021 • 2 minutes to read • [Edit Online](#)

Retailers can have business scenarios and processes that are unique to them. Therefore, not all the rules that are included by default in the commerce transaction consistency checker are applicable to all retailers. To accommodate differences, Microsoft Dynamics 365 Commerce provides functionality that can be used to disable the rules that aren't applicable.

To view the list of rules that are available in the transaction consistency checker in your environment, and to see the status of each rule, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**, and select the **Transaction validation** tab.

By default, the status of every rule is set to **Enabled**. Therefore, all the rules are used to validate transactions before they are pulled into the commerce statements. To disable a rule, change its status to **Disabled**. Disabled rules aren't considered when transactions are validated during the statement calculation process.

To bypass the whole validation process, regardless of the rules that are enabled, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**, and then, on the **Transaction validation** tab, set the **Disable consistency checker for Commerce transactions** option to **Yes**. After this option is set to **No**, it can't be set back to **Yes** from the user interface (UI).

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Edit and audit cash and carry and cash management transactions

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to edit and audit cash and carry and cash management transactions in Microsoft Dynamics 365 Commerce.

## Overview

Dynamics 365 Commerce customers use both a first-party point of sale (POS) application and third-party POS applications. In the case of the first-party application, store transactions are pulled into Commerce headquarters from the channels through a batch process. In the case of third-party applications, transactions are pulled into Commerce headquarters through integration. In both cases, after transactions are pulled into Commerce headquarters, a consistency check process must be performed. This process runs multiple validations on the transactions, and only transactions that are successfully validated are pulled into the statement so that they can be posted in Commerce headquarters.

Commerce transactions might fail validation for various reasons. A bug in the integration code or in the POS application might cause inconsistent data. Alternatively, a user error might cause inconsistent data. For example, a user deletes a product after it was synced to the channel, or a user closes a fiscal period without posting transactions for that period. Although these transactions are flagged and are excluded from the statements, they can disrupt a customer's daily process of posting daily sales to the financials. In Commerce, you can edit the transactions that fail validation while you also maintain an audit of all the changes.

## Edit transactions

In Commerce version 10.0.5, the only transactions that can be edited are cash and carry transactions, such as sales and returns. Customer orders and online orders can't be edited. In Commerce version 10.0.6 and later, cash management transactions can also be edited.

To edit transactions in Commerce headquarters, follow these steps.

1. Install the [Microsoft Dynamics Office Add-in](#).
2. In Commerce headquarters, open the **Store financials** workspace. The **Transaction validation failures** tile provides a prefiltered view of the transaction page that failed one or more validation rules.
3. Open the transaction page, select the record that failed validation, select **Office Add in**, and then select **Edit selected transaction**. An Excel file that shows the details of the selected transaction is opened. This file contains the following worksheets:
  - **Lines** – This worksheet has the header and product lines for the transaction, and related data for the transaction.
  - **Payments** – This worksheet has the details of the payment lines for the transaction.
  - **Discounts** – This worksheet has the discount-related details for the transaction.
  - **Taxes** – This worksheet has the tax-related details for the transaction.
  - **Charges** – This worksheet has the charges-related data for the transaction.
4. In the Excel file, modify the appropriate fields, and then upload the data back into Commerce headquarters by using the publishing functionality of the Dynamics Excel Add-in. After the data is published, the changes will be reflected in the system. During publication, no validation is done for

changes that users make.

5. You can view a complete audit trail of the changes by selecting **View audit trail** in the **Retail transaction** header for the header-level changes, and in the relevant section and record on the appropriate transaction page. For example, all changes that are related to sales lines will be shown on the **Sales transactions** page, and all changes that are related to payments will be shown on the **Payment transactions** page. The following audit details are maintained for the changes:

- Modified date and time
- Field
- Old value
- New value
- Modified by

6. After you've made and published your changes, run **Validate store transactions** to validate that those changes are consistent and valid.

#### NOTE

You can edit only transactions that failed validation. If you want to edit a transaction that passed validation, change the validation status of the transaction to **Error** or **None**, and then publish the change. You can then edit the data on the header or in any other child records of the transaction, and publish the header or records.

## Bulk-edit transactions that are linked to a statement

Commerce version 10.0.6 and later support the option to bulk-edit transactions at the statement level.

To bulk-edit transactions that are linked to a statement in Commerce headquarters, follow these steps.

1. Open the **Statements** page, and select the statement that contains the transactions that must be edited.
2. Select the **Open in Microsoft Office** button.
3. Depending on what must be edited, select one of the following options:
  - **Edit cash and carry transactions** – This option lets you edit all the cash and carry transactions that are included in the statement. The following Excel worksheets are available:
    - **Transaction** – This worksheet has all the header-level information for the sales transactions.
    - **Sales transaction** – This worksheet has all the line-level information for the sales transactions.
    - **Payment transactions** – This worksheet has all the payment line information for the sales transactions.
    - **Discount transactions** – This worksheet has all the discount line information for the sales transactions.
    - **Tax transactions** – This worksheet has all the tax line information for the sales transactions.
    - **Charge transactions** – This worksheet has all the charge line information for the sales transactions.
  - **Edit cash management transactions** – This option lets you edit all the cash management transactions that are included in the statement. The following Excel worksheets are available:
    - **Transaction** – This worksheet has all the header-level information for the cash management transactions.
    - **Bank tender transactions** – This worksheet has all the bank drop transaction details.
    - **Safe tender transactions** – This worksheet has all the safe drop transaction details.

- **Tender declaration** – This worksheet has all the tender declaration transaction details.
- **Income-expense transaction** – This worksheet has all the income-expense transaction line details.
- **Payment transactions** – This worksheet has all the payment-related information for the **Pay invoice** operation and the income-expense transaction.

4. Edit the required transactions.

#### NOTE

Validations aren't done when you publish bulk-edited transactions. You must make sure that all your edits are accurate, and that the fidelity of data across the worksheets is maintained. For example, if you want to change the transaction date, so that you can manage scenarios where the fiscal or inventory period for the open transactions is closed, you must change the date on all the Excel worksheets that have the **Business date** column. To validate transactions after they have been edited, you can select **Revalidate transactions** on the **Statements** page. Wait for the validation job to finish running before you post the statement.

5. If the aggregation has already been generated, open the **Aggregated transactions** page, and select **Regenerate sales order xml**.

## Bulk-edit transactions that aren't linked to a statement

Commerce version 10.0.10 and later support the option to bulk-edit transactions that fail validation but aren't linked to a statement.

To bulk-edit transactions that aren't linked to a statement in Commerce headquarters, follow these steps.

1. Open the **All stores** page, and select the store that transactions must be edited for.
2. Select the **Open in Microsoft Office** button, and then select **Edit cash and carry transactions**.
3. Edit the required transactions, and then publish them.

## Additional resources

[Edit and audit online order and asynchronous customer order transactions](#)

[Edit financial dimensions for retail transactions](#)

[Create an Excel workbook to edit retail transactions](#)

[Add fields to an Excel workbook to edit retail transactions](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Edit and audit online order and asynchronous customer order transactions

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to edit and audit online order and asynchronous customer order transactions in Microsoft Dynamics 365 Commerce.

## Overview

Between Commerce versions 10.0.5 and 10.0.6, support was added for editing cash and carry transactions (such as sales and returns) and cash management transactions (such as float entry and tender removal). In Commerce version 10.0.7, support was added for editing online order transactions and asynchronous customer order transactions.

## Edit and audit order transactions

To edit and audit order transactions in Commerce headquarters, follow these steps.

1. Install the [Microsoft Dynamics Office Add-in](#).
2. On the **Retail parameters** page, on the **Customer orders** tab, on the **Order** FastTab, specify a hold code for **Hold code for order synchronization errors**.
3. Open the **Store financials** workspace. The **Online order synchronization errors** and **Customer order synchronization errors** tiles provide a prefiltered view of the retail transaction page. Each shows the transaction records that have failed synchronization for the corresponding order type.
4. Open either the **Online order synchronization errors** page or the **Customer order synchronization errors** page. Select a record to view the synchronization error details. The **Synchronization status** FastTab provides the following error details:
  - Pending order status
  - Order error details
  - Modified date and time
  - Retry count
5. If the error details indicate that the record must be fixed, select **Office Add in**, and then select **Edit selected transaction**. An Excel file that shows the details of the selected transaction is opened.
  - If the transaction that is being edited is an online order transaction, the Excel file contains the following worksheets:
    - **Transaction** – This worksheet has the header details for the transaction.
    - **Sales transaction** – This worksheet has the line details for the transaction.
    - **Payment transactions** – This worksheet has the details of the payment lines for the transaction.
    - **Discount transactions** – This worksheet has the discount-related details for the transaction.
    - **Tax transactions** – This worksheet has the tax-related details for the transaction.
    - **Charges transactions** – This worksheet has the charges-related data for the transaction.
  - If the transaction that is being edited is an asynchronous customer order transaction, the Excel file contains the following worksheets:

- **Lines** – This worksheet has the header and line details for the transaction.
  - **Payments** – This worksheet has the details of the payment lines for the transaction.
  - **Discounts** – This worksheet has the discount-related details for the transaction.
  - **Taxes** – This worksheet has the tax-related details for the transaction.
  - **Charges** – This worksheet has the charges-related data for the transaction.
6. In the Excel file, in the **Pending order status** field, enter **Editing**, and then publish the change. In this way, you prevent the **Synchronize order** job that is running in batch mode from skipping this record during processing.
  7. In the Excel file, modify the appropriate fields, and then upload the data back into Commerce headquarters by using the publishing functionality of the Dynamics Excel Add-in. After the data is published, the changes will be reflected in the system. During publication, no validation is done for changes that users make.
  8. You can view a complete audit trail of the changes by selecting **View audit trail** in the **Retail transaction** header for the header-level changes, and in the relevant section and record on the appropriate transaction page. For example, all changes that are related to sales lines will be shown on the **Sales transactions** page, and all changes that are related to payments will be shown on the **Payment transactions** page. The following audit details are maintained for the changes:
    - Modified date and time
    - Field
    - Old value
    - New value
    - Modified by
  9. After you've made and published your changes, select **Synchronize order** to immediately start the synchronization process. Alternatively, you can wait for the synchronization process that is running in batch mode to process the transaction.

By default, after the orders are successfully synced, they are put in a hold status, based on the hold code that is defined in the Commerce parameters. The hold on the orders must be removed before the orders can be processed further for fulfillment or other operations.

## Additional resources

[Edit and audit cash and carry and cash management transactions](#)

[Edit financial dimensions for retail transactions](#)

[Create an Excel workbook to edit retail transactions](#)

[Add fields to an Excel workbook to edit retail transactions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Edit financial dimensions for retail transactions

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to edit financial dimensions for retail transactions in Microsoft Dynamics 365 Commerce.

## Edit financial dimensions

To edit financial dimensions for retail transactions in Commerce headquarters, follow these steps.

1. Open the **Financial dimensions configuration for integrating applications** page.
2. Select the active **Default dimensions integration** record.
3. On the **Financial dimensions** FastTab, make sure that all the dimensions that you want to edit in the Excel worksheet are present in the **Selected** list. For more information, see [Data entities](#).
4. Download and open the Excel file from the **Statements** page, the **Retail transactions** page, or the **Transaction validation failures** tile in the **Store financials** workspace.
5. To change the transaction financial dimension, select **Design**, and then select the pencil symbol next to the **Transaction (auditable)** row.
6. Find and select the **FinancialDimensionDisplayValue** field, select a cell in the header part of the Excel worksheet, and then select **Add label**.
7. Select a cell below the cell that you selected in the previous step, select **Add Value**, and then select **Refresh**. The financial dimensions are added to the Excel worksheet, and they can then be edited and published.
8. To change the dimensions on the transaction lines, select the **Sales transactions (auditable)** row, select the pencil symbol, and then repeat steps 6 and 7.
9. To change the dimensions on the payment lines, select the **Payment transactions (auditable)** row, select the pencil symbol, and then repeat steps 6 and 7.

## Additional resources

[Edit and audit cash and carry and cash management transactions](#)

[Edit and audit online order and asynchronous customer order transactions](#)

[Create an Excel workbook to edit retail transactions](#)

[Add fields to an Excel workbook to edit retail transactions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create an Excel workbook to edit retail transactions

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create an Excel workbook so that you can edit retail transactions in Microsoft Dynamics 365 Commerce.

## Overview

There is a predefined Excel template that customers can access from different parts of the system and use to edit and audit retail transactions. However, customers can also create a custom Excel workbook for this purpose.

## Create and configure an Excel workbook

To create and configure an Excel workbook so that you can edit retail transactions, follow these steps.

1. Open Excel, and create a blank workbook.
2. On the **Insert** tab, select **My add-ins**.
3. In the right pane, select the **Add server information** link.
4. Enter the server URL, and then select **OK**.
5. If you receive a "No applet registrations found" error message, follow these steps to resolve the issue:
  - a. In Commerce, go to **System administration > Setup > Office app parameters**.
  - b. On the **App parameters** FastTab, select **Initialize app parameters**.
  - c. In the confirmation message box, select **OK**.
  - d. On the **Registered applets** FastTab, select **Initialize applet registration**.
  - e. Repeat the previous three steps as required.
6. Select **Design**, and then select **Add table**.
7. Based on the data that has to be modified, select the entities that must be added to the Excel workbook for editing. Use the following table as a reference.

TRANSACTION TYPE	DATA ENTITIES TO USE
Cash and carry transactions, Online orders, Async customer orders, Async customer quotes	Transaction (auditable), Sales transaction (auditable), Payment transactions (auditable), Tax transactions (auditable), Discount transactions (auditable), Charge transactions (auditable)
Bank drop	Transaction (auditable), Banked tender transactions (auditable)
Safe drop	Transaction (auditable), Safe tender transactions (auditable)
Tender declaration	Transaction (auditable), Tender declaration transactions (auditable)

TRANSACTION TYPE	DATA ENTITIES TO USE
Income, Expense	Transaction (auditable), Income/Expense transactions (auditable), Payment transactions (auditable)
Declare starting amount, Tender removal, Float entry, Change tender, Invoice payment, Customer deposit	Transaction (auditable), Payment transactions (auditable)

**NOTE**

It's important that you add only one data entity to each Excel workbook. Additionally, all fields that are marked by a key symbol must be added to the relevant workbook.

8. After the workbook is configured, apply the required filters. Be sure to apply the same filters to all the worksheets in the file. Avoid loading very large amounts of data into the Excel file. Otherwise, performance might be affected, and Excel and your system might slow down. We recommend that you always use "Company" and either "Statement Number" or "Transaction Number" as filters for your file.
9. After the filters are configured, select **Refresh** to load the data.
10. Edit the required data, and then publish it. If the **Publish** button is unavailable, some key fields probably weren't added to the Excel workbook.

## Additional resources

[Edit and audit cash and carry and cash management transactions](#)

[Edit and audit online order and asynchronous customer order transactions](#)

[Edit financial dimensions for retail transactions](#)

[Add fields to an Excel workbook to edit retail transactions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add fields to an Excel workbook to edit retail transactions

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add fields to a Microsoft Excel workbook so that you can edit retail transactions in Microsoft Dynamics 365 Commerce.

## Overview

When you generate an Excel file so that you can edit retail transactions, the file is filled with some default fields. If a field that must be updated isn't visible by default in the generated Excel file, you can add it.

## Add fields to a worksheet in an Excel workbook

To add fields to an Excel workbook so that you can edit retail transactions, follow these steps.

1. Download and open the Excel file from the **Statements** page, the **Retail transactions** page, or the **Transaction validation failures** tile in the **Store financials** workspace.
2. Select **Design**.
3. Select the pencil symbol for the desired table, and then, in the list of available fields, find and select the field that you want to add.
4. Select **Add**, and then select **Update**. You can reorder fields.
5. After the update is completed, select **Refresh** to fetch the data for the new column.

The new field and data for it should now be available for editing in Excel.

## Additional resources

[Edit and audit cash and carry and cash management transactions](#)

[Edit and audit online order and asynchronous customer order transactions](#)

[Edit financial dimensions for retail transactions](#)

[Create an Excel workbook to edit retail transactions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Trickle feed-based order creation for retail store transactions

2/18/2021 • 2 minutes to read • [Edit Online](#)

In Dynamics 365 Retail versions 10.0.4 and earlier, statement posting is an end-of-day operation and all transactions are posted in the books at the end of the day. Large transactions must then be processed in a limited time window, sometimes resulting in load and locks and statement posting failures. Retailers also can't recognize revenue and payments in their books throughout the day.

With trickle feed-based order creation introduced in Retail version 10.0.5, transactions are processed throughout the day, and only the financial reconciliation of tenders and other cash management transactions are processed at the end of the day. This functionality splits the load of creating sales orders, invoices, and payments throughout the day, providing better perceived performance and the ability to recognize revenue and payments in the books in near real-time.

## How to use trickle feed-based posting

1. To enable trickle feed-based posting of retail transactions, enable the feature named **Retail statements - Trickle feed** using Feature management.

### IMPORTANT

Before you enable the feature, make sure that no pending statements are waiting to be posted.

2. The current statement document will be split into two types: transactional statement and financial statement.
  - The transactional statement will pick up all unposted and validated transactions and create sales orders, sales invoices, payment and discount journals, and income-expense transactions at the cadence that you configure. You should configure this process to run at a high frequency so that documents are created when the transactions are uploaded into Headquarters through the P-job. With the transactional statement that already creates sales orders and sales invoices, there is no real need to configure the **Post inventory** batch job. However, you can still use it to meet specific business requirements that you may have.
  - The financial statement is designed to be created at the end of the day and only supports the closing method of **Shift**. This statement will be limited to financial reconciliation and will only create the journals for the difference amounts between counted amount and transaction amount for the different tenders, along with journals for other cash management transactions.
3. To calculate the transactional statement, go to **Retail and Commerce > Retail and Commerce IT > POS Posting > Calculate transactional statements in batch**. To post the transactional statements in batch, go to **Retail and Commerce > Retail and Commerce IT > POS Posting > Post transactional statements in batch**.
4. To calculate the financial statement, go to **Retail and Commerce > Retail and Commerce IT > POS Posting > Calculate financial statements in batch**. To post the financial statements in batch, go to **Retail and Commerce > Retail and Commerce IT > POS Posting > Post financial statements in batch**.

**NOTE**

The menu items **Retail and Commerce > Retail and Commerce IT > POS Posting > Calculate statements in batch** and **Retail and Commerce > Retail and Commerce IT > POS Posting > Post statements in batch** are removed with this new feature.

Alternately, transactional and financial statement types can be created manually. Go to **Retail and Commerce > Channels > Stores** and click **Statements**. Click **New** and then choose the type of statement that you want to create. Fields on the **Statements** page and actions under the **Statement group** of the page will show relevant data and actions based on the selected statement type.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Create, calculate, and post statements for a retail store

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the manual steps for creating, calculating, and posting a statement for a store. There are also batch jobs that can be configured for the same tasks. The steps for configuring and running the batch jobs can be found in other topics. To complete this procedure, you must have transactions that were completed in POS and then pulled into Dynamics 365 Commerce. This recording uses the USRT company in demo data.

1. Select **Store financials** from the home page.
2. Select **New statement**.
3. In the **Store number** field, select a option from the drop-down.
4. Select **OK**.
5. The **Setup** group has the settings that control what transactions are included in the statement and how they are grouped into statement lines. You can open the **Setup** group and change these settings, or you can use the defaults.
  - The **Statement method** field defines how the statement lines will be grouped.
  - Select a staff member or a register in the **staff/register** field if you want to calculate a statement only for the specific staff member or register.
6. In the **Closing method** field, select an option.
7. Select **Calculate statement** from the Action Pane.
8. Select **Yes**.
  - After calculating the statement, there should be lines created with total amounts for each payment method and statement method that was used.
  - Enter a counted amount in each line if it needs to be entered or updated. The counted field is populated with amounts from tender declarations done in POS.
9. Select **Post statement** from the Action Pane.
10. Select **Close**.
11. Close the pane.
12. At the home page, select **Store financials**.
13. Select the **Posted statements** tab.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Financial reconciliation in retail stores

2/18/2021 • 3 minutes to read • [Edit Online](#)

In Microsoft Dynamics 365 Commerce version 10.0.10 and earlier, the functionality that the point of sale (POS) client provides for end-of-day processes in retail stores lets store clerks and store managers perform end-of-day operations. For example, they can do tender declarations, blind-close shifts, reconcile shift transactions, and close shifts. However, there is no capability in POS to finalize the financial information for shifts, so that it can be used to post the financials in Commerce headquarters. Typically, store managers are responsible for completing this task. Before they can sign off on a shift, they must review the information, make any corrections that are required, and finalize the totals for that shift. The finalized totals should then be posted in financial modules in Commerce headquarters.

Additionally, in Commerce version 10.0.10 and earlier, store managers can review and make some adjustments to statement lines in Commerce headquarters. However, the capability is limited, and store managers rarely have access to the Commerce headquarters client. Moreover, financial retail statement review and adjustment can be done only when the statements are created in Commerce headquarters. However, that process is typically a nightly process. Therefore, store managers must wait for the shift sign-off when financial retail statements are created in Commerce headquarters.

In Commerce version 10.0.11 and later, store managers can review, adjust, and finalize their shifts in the POS client itself. That data is then used to create and post financial retail statements in Commerce headquarters.

## NOTE

The functionality that is related to financial reconciliation in retail stores works only if trickle feed-based order creation is turned on. For more information, see [Trickle feed-based order creation for retail store transactions](#).

## Set up financial reconciliation

Follow these steps to use the financial reconciliation functionality.

1. In the **Feature management** workspace, turn on the **Retail statements - Trickle feed** feature.
2. In the POS functionality profile for the appropriate store, set the **Enable financial reconciliation in store** option to **Yes**.

## More information about financial reconciliation

When the financial reconciliation functionality is turned on, some of the parameters that are defined on the **Statement/closing** FastTab of the **Retail stores** page in Commerce headquarters are synced to the channel database. The same set of calculation criteria and amount thresholds that is used for retail statements is enforced.

When the **Close shift** operation is invoked, the system validates that the system-computed amounts and the declared amounts match. If they differ, and the difference exceeds defined thresholds, the user is prompted and can make the required adjustments.

Adjustments can be made for each tender. When a tender is selected, the user can view the totals for different transaction types and edit the totals for a specific transaction type. During editing, the system shows the original computed amount and the overridden amount for that transaction type. The user can also capture notes as a part of the editing process. Notes can be used for auditing purposes.

Users can ignore the validation prompts and messages, and can proceed to close the shift. However, if a user ignores the validation prompts, the same issues will arise and will have to be fixed when the shift posts financial statements in Commerce headquarters.

The **Close shift** operation in POS also validates that all transactions in the offline database are synced to the channel database. If any transactions aren't synced, the user receives a warning message, because this situation can cause the system amounts to be incorrectly computed. In this situation, the user can end the **Close shift** operation and try to sync the offline transactions to the channel database. Alternatively, the user can manually adjust the system-computed amounts to account for the transactions that aren't synced, so that the correct set of financial numbers is finalized and posted.

When trickle feed statement posting is used, so that the posting of transactions is separated from the posting of financials, you can choose to adjust the system-computed amounts for missing offline transactions. In this way, you ensure that financials are always accounted and posted correctly, regardless of missing transactions. Offline transactions can be synced to the channel database and Commerce headquarters, and then posted later, separately from the financial postings.

Details of the financial reconciliation for a shift are synced to Commerce headquarters by using the P-job.

Financial retail statements in Commerce headquarters don't compute totals to show the details on the statement lines. Instead, the finalized amounts in the POS client are used to create and post financial retail statements.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Improvements to statement posting functionality

2/18/2021 • 14 minutes to read • [Edit Online](#)

This topic describes the first set of improvements that have been made to the statement posting feature. These improvements are available in Microsoft Dynamics 365 for Finance and Operations 7.3.2.

## Activation

By default, during deployment of Finance and Operations 7.3.2, the program is set up to use the legacy feature for statement postings. To enable the improved statement posting feature, you must turn on the configuration key for it.

- Go to **System administration > Setup > License configuration**, and then, under the **Retail and Commerce** node, clear the **Statements (legacy)** check box, and select the **Statements** check box.

When the new **Statements** configuration key is turned on, a new menu item that is named **Statements** is available. This menu item lets you manually create, calculate, and post statements. Any statement that causes an error when the batch posting process is used will also be available through this menu item. (When the **Statements (legacy)** configuration key is turned on, the menu item is named **Open statements**.)

Commerce includes the following validations that are related to these configuration keys:

- Both configuration keys can't be turned on at the same time.
- The same configuration keys must be used for all the operations that are performed on a given statement during its lifecycle (Create, Calculate, Clear, Post, and so on). For example, you can't create and calculate a statement while the **Statement (legacy)** configuration key is turned on, and then try to post the same statement while the **Statement** configuration key is turned on.

### NOTE

We recommend that you use the **Statements** configuration key for the improved statement posting feature, unless you have compelling reasons to use the **Statements (legacy)** configuration key instead. Microsoft will continue to invest in the new and improved statement posting feature, and it's important that you switch to it at the earliest opportunity to benefit from it. The legacy statement posting feature is deprecated starting in 8.0 release.

## Setup

As part of the improvements to the statement posting feature, three new parameters have been introduced on the **Statement** FastTab on the **Posting** tab of the **Commerce parameters** page:

- **Disable clear statement** – This option is applicable only for the legacy statement posting feature. We recommend that you set this option to **No** to prevent users from clearing statements that are in a semi-posted state. If statements that are in a semi-posted state are cleared, data becomes corrupted. You should set this option to **Yes** only in exceptional circumstances.
- **Reserve inventory during calculation** – We recommend that you use the **Post inventory** batch job for inventory reservation, and that you set this option to **No**. When this option is set to **No**, the improved statement posting feature doesn't try to create inventory reservation entries at the time of calculation (if entries weren't already created through the **Post inventory** batch job). Instead, the feature creates inventory reservation entries only at the time of posting. This implementation was a design choice and was based on the fact that the time window between the calculation process and the posting process is

typically small. However, if you want to reserve inventory at the time of calculation, you can set this option to **Yes**.

The legacy statement posting feature always reserves inventory during the statement calculation process (if reservation wasn't already done through the **Post inventory** batch job), regardless of the setting of this option.

- **Disable counting required** – When this option is set to **Yes**, the posting process for a statement continues, even if the difference between the counted amount and the transaction amount on the statement is outside the threshold that is defined on the **Statement** FastTab for stores.

Additionally, the following parameters have been introduced on the **Batch processing** FastTab on the **Posting** tab of the **Commerce parameters** page:

- **Maximum number of parallel statement posting** – This field defines the number of batch tasks that will be used to post multiple statements.
- **Max thread for order processing per statement** – This field represents the maximum number of threads used by the statement posting batch job to create and invoice sales orders for a single statement. The total number of threads that will be used by the statement posting process will be computed based on the value in this parameter multiplied by the value in the **Maximum number of parallel statement posting** parameter. Setting the value of this parameter too high can negatively impact the performance of the statement posting process.
- **Max transaction lines included in aggregation** – This field defines the number of transaction lines that will be included in a single aggregated transaction before a new one is created. Aggregated transactions are created based on different aggregation criteria such as customer, business date, or financial dimensions. It is important to note that the lines from a single transaction will not be split across different aggregated transactions. This means that there is a possibility that the number of lines in a aggregated transaction is slightly higher or lower based on factors such as number of distinct products.
- **Maximum number of threads to validate store transactions** – This field defines the number of threads that will be used to validate transactions. Validating transactions is a required step that needs to occur before the transactions can be pulled into the statements. You also need to define a **Gift card product** on the **Gift card** FastTab on the **Posting** tab of the **Commerce parameters** page. This needs to be defined even if gift cards are not used by the organization.

#### **NOTE**

All settings and parameters that are related to statement postings, and that are defined on stores and on the **Commerce parameters** page, are applicable to the improved statement posting feature.

## Processing

Statements can be calculated and posted in batch using the menu items **Calculate statements in batch** and **Post statements in batch**. Alternatively, statements can be manually calculated and posted by using the **Statements** menu item that the improved statement posting feature provides.

The process and steps for calculating and posting statements in a batch are the same as they were in the legacy statement posting feature. However, significant improvements have been made in the core back-end processing of the statements. These improvements make the process more resilient, and provide for better visibility into the states and error information. Therefore, users can address the root cause of errors and then continue the posting process without causing data corruption and without causing data fixes to be required.

The following sections describe some of the major improvements for the statement posting feature that appear in the user interface for statements and posted statements.

## Status details

A new state model has been introduced in the statement posting routine across the calculation and posting processes.

The following table describes the various states and their order during the calculation process.

STATE ORDER	STATE	DESCRIPTION
1	Started	The statement was created and is ready to be calculated.
2	Marked	The transactions that are in scope for the statement are identified based on the statement parameters, and they are marked with the statement ID.
3	Calculated	The statement lines are computed and shown.

The following table describes the various states and their order during the posting process.

STATE ORDER	STATE	DESCRIPTION
1	Checked	Multiple validations are done that are related to parameters (for example, the disposition charge), and to the statement and statement lines (for example, the difference between the counted amount and the transaction amount).
2	Aggregated	Sales transactions for named and unnamed customers are aggregated based on the configuration. Every aggregated transaction is eventually converted to a sales order.
3	Customer order created	Based on the aggregated transaction, sales orders are created in the system.
4	Customer order invoiced	Sales orders are invoiced.
5	Discounts posted	Periodic discount journals are posted based on the configuration.
6	Income/expense posted	Income/expense transactions are posted as vouchers.
7	Vouchers linked	Payment journals are created and linked to the corresponding invoice.
8	Payments posted	Payment journals are posted.
9	Gift cards posted	Gift card transactions are posted as vouchers.
10	Posted	The statement is marked as posted.

Every state in the preceding tables is independent in nature, and a hierarchical dependency is built between the states. This dependency flows from top to bottom. If the system encounters any errors while it's processing a state, the status of the statement is reverted to the previous state. Any subsequent reattempt of the process resumes from the state that failed and continues to move forward. This approach has the following benefits:

- The user has complete visibility into the state where the error occurred.
- Data corruption is avoided. For example, in the legacy statement posting feature, there were instances where some sales orders were invoiced but others were left open. There were also instances where some payment journals didn't have a corresponding invoice to settle, because the invoice posting had an error.
- Users can see the current state of a statement by using the **Status details** button in the **Execution details** group of the statement. The status details page has three sections:
  - The first section shows the current status of the statement, together with the error code and a detailed error message, if an error occurred.
  - The second section shows the various states of the calculation process. Visual cues indicate states that have been successfully run, states that could not be run because of errors, and states that haven't yet been run.
  - The third section shows the various states of the posting process. Visual cues indicate states that have been successfully run, states that could not be run because of errors, and states that haven't yet been run.

Additionally, the header of the second and third sections shows the overall status of the relevant process.

### Event logs

A statement goes through various operations (for example, Create, Calculate, Clear, and Post), and multiple instances of the same operation might be called during the statement's lifecycle. For example, after a statement is created and calculated, a user can clear the statement and calculate it again. The **Event logs** button in the **Execution details** group of the statement provides a complete audit trail of the various operations that were called on a statement, together with information about when those operations were called.

### Aggregated transactions

During the posting process, the sales transactions are aggregated based on the configuration. These aggregated transactions are stored in the system and used to create sales orders. Every aggregated transaction creates one corresponding sales order in the system. You can view the aggregated transactions by using the **Aggregated transactions** button in the **Execution details** group of the statement.

The **Sales order detail** tab of an aggregated transaction shows the following information:

- **Record ID** – The ID of the aggregated transaction.
- **Statement number** – The statement that the aggregated transaction belongs to.
- **Date** – The date when the aggregated transaction was created.
- **Sales ID** – When a sales order is created from the aggregated transaction, the sales order ID. If this field is blank, the corresponding sales order hasn't been created.
- **Number of aggregated lines** – The total number of lines for the aggregated transaction and sales order.
- **Status** – The last status of the aggregated transaction.
- **Invoice ID** – When the sales order for the aggregated transaction is invoiced, the sales invoice ID. If this field is blank, the invoice for the sales order hasn't been posted.

The **Transaction details** tab of an aggregated transaction shows all the transactions that have been pulled into the aggregated transaction. The aggregated lines on the aggregated transaction show all the aggregated records from the transactions. The aggregated lines also show details such as the item, variant, quantity, price, net amount, unit, and warehouse. Basically, each aggregated line corresponds to one sales order line.

From the **Aggregated transactions** page, you can download the XML for a specific aggregated transaction by using the **Export sales order XML** button. You can use the XML to debug issues that involve sales order creation and posting. Just download the XML, upload it to a test environment, and debug the issue in the test environment. The functionality for downloading the XML for aggregated transactions isn't available for statements that have been posted.

The aggregated transaction view provides the following benefits:

- The user has visibility into the aggregated transactions that failed during sales order creation and the sales orders that failed during invoicing.
- The user has visibility into how transactions are aggregated.
- The user has a complete audit trail, from transactions, to sales orders, to sales invoices. This audit trail wasn't available in the legacy statement posting feature.
- Aggregated XML file make it easier to identify issues during sales order creation and invoicing.

### **Journal vouchers**

The **Journal vouchers** button in the **Execution details** group of the statement shows all the various voucher transactions that are created for a statement, and that are related to discounts, income/expense accounts, gift cards, and so on.

Currently, the program shows this data only for posted statements.

### **Payment journals**

The **Payment journals** button in the **Execution details** group of the statement shows all the various payment journals that are created for a statement.

Currently, the program shows this data only for posted statements.

## Other improvements

Other, back-end improvements that users can see have been made to the statement posting feature. Here are some examples:

- The aggregation doesn't consider the staff, terminal, and shift entities. Because there are fewer aggregation parameters, fewer sales order lines must be processed.
- The occurrence of deadlock on transaction tables is reduced by introducing additional extension tables and by doing insert operations instead of update operations on the transaction tables.
- The number of running batch tasks has been parameterized and limited. Therefore, this number can be fine-tuned specifically to a customer's environment. In the legacy statement posting feature, an unlimited number of batch tasks was created at the same time. The results were unmanageable loads, overhead, and bottlenecks on the batch server.
- Statements are efficiently queued for processing by prioritizing the statements that have the maximum number of transactions.
- Batch processes such as **Calculate statements in batch** and **Post statements in batch** are run only in batch mode. In the legacy statement posting feature, users could choose to run these batch processes in an interactive mode which is a single threaded operation unlike batch processes which are multi-threaded.
- In the legacy statement posting feature, any failure of a batch task put the whole batch job in an error state. In the improved feature, batch task failures don't put the batch job in an error state if other batch tasks are successfully completed. You should assess the posting status for a batch execution run by using the **Statements** page, where you can see any statements that weren't posted because of errors.
- In the legacy statement posting feature, the first occurrence of a statement failure causes the whole batch to fail. The remaining statements aren't processed. In the improved feature, the batch process continues to process all statements, even if some of the statements fail. One benefit is that users gain visibility into the exact number of statements that have errors. Therefore, users don't have to be stuck in a continuous loop of



fixing the errors and running the post statement process till all statements are posted.

## General guidance about the statement posting process

- We recommend that you run the statement posting process in a batch, because batch runs take advantage of the power of the batch framework in terms of multithreading. Multithreading is required in order to handle the huge volumes of transactions that are normally seen in statement postings.
- We recommend that you turn on negative physical inventory on the item model group, so that you have a seamless posting experience. In some scenarios, negative statements might not be able to be posted unless there is negative physical inventory. For example, in theory, if there is only one unit of an item in inventory, and there have been a sales transaction and a return transaction for the item, the transaction should be able to be posted even if negative inventory isn't turned on. However, because the statement posting process pulls both the sales transaction and the return transaction in a single customer order, there is no guarantee that the sales line will be posted first, followed by the return line. Therefore, errors can occur. If negative inventory is turned on in this scenario, the transaction posting isn't negatively affected, and the system will correctly reflect the inventory.
- We recommend that you use aggregation while you calculate and post statements. Therefore, the following settings are recommended for some of the aggregation parameters:
  - Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**. Then, on the **Posting** tab, on the **Inventory update** FastTab, in the **Detail level** field, select **Summary**.
  - Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**. Then, on the **Posting** tab, on the **Aggregation** FastTab, set the **Voucher transactions** option to **Yes**.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Improved handling of batch-tracked items

2/18/2021 • 2 minutes to read • [Edit Online](#)

In Point of Sale (POS), batch numbers can't be captured for batch-tracked items at the time of sale. However, for specific configurations, when sales are posted in the headquarters through customer orders or statement posting, the Microsoft Dynamics system expects that valid batch numbers for batch-tracked items exist, and that they will be used during the invoicing process.

If valid batch numbers are available for products, the customer order invoicing process and the sales order invoicing process from statement posting use them. Otherwise, the customer order invoicing process can't post, and the POS user receives an error message. Statement posting then goes into an error state. This error state occurs even when negative inventory has been turned on for the products.

Improvements that have been made in Retail version 10.0.4 and later help guarantee that, when negative inventory is turned on for batch-tracked items, customer order invoicing and sales order invoicing through statement posting aren't blocked for those items if the inventory is 0 (zero) or a batch number isn't available. The new functionality uses a default batch ID for the sales lines when batch numbers aren't available.

To define the default batch ID that is used for customer orders, on the **Commerce parameters** page, on the **Customer orders** tab, on the **Order** FastTab, set the **Default batch id** field.

To define the default batch ID that is used for sales order invoicing through statement posting, on the **Commerce parameters** page, on the **Posting** tab, on the **Inventory update** FastTab, set the **Default batch id** field.

## NOTE

This functionality is available only when advanced warehousing is turned on for the specific store warehouse and items. In a later release, the functionality will also be supported for scenarios where advanced warehouse management isn't used.

## NOTE

Support for improved handling of batch-tracked items during statement posting for non-advanced warehouse management scenarios was introduced in Retail version 10.0.5.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Payment configurations for Retail statements

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates configurations for Commerce store payment methods, which affect how statements get created and posted.

This recording uses the USRT demo company.

1. Go to Retail and Commerce > Channels > Stores > All stores.
2. In the list, find and select the desired record.
3. In the list, click the link in the selected row.
4. On the Action Pane, click Set up.
5. Click Payment methods.
6. Expand or collapse the Posting section.
7. Click Edit.
  - Select whether the amounts received for this payment method should be posted to a ledger account or bank account.
  - Select the account that amounts received for this payment method should be posted to.
  - Select an account to post possible differences between the total transaction amount received and the amount counted for this payment method.
  - In this field you can enter an amount to control when the difference amount should be posted to another difference account. You can use this to track big differences.
  - Select an account to post possible differences between the total transaction amount received and the amount counted, when it exceeds the value that is defined in the "Maximum difference amount" field.
  - Select "Yes" to post bank drop amounts to a separate account.
  - In this field you can select whether bank drop amounts should be posted to a ledger account or a bank account.
  - Select the account to post bank drop amounts into.
  - Select the bank transaction type to use when posting bank drop amounts to the bank account.
  - Select "Yes" to post safe drop amounts to a separate account.
  - Select whether safe drop amounts should be posted to the ledger account or the bank account.
  - Select the account to post safe drop amounts into.
8. Click Save.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and run job to calculate statements

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through configuring and running recurrent batch jobs to create and calculate statements for a selected store or group of stores. This procedure uses the USRT company in demo data.

1. Go to All workspaces > Store financials.
2. Click Calculate statements.
  - Select either a specific store, or a node if you want to create the batch job for a group of stores.
  - Click the arrow to add your selection.
3. Click the Run in the background tab.
4. Under Batch processing, select 'Yes'.
5. Click Recurrence.
6. In the Start date field, enter a date.
7. In the Start time field, enter a time.
8. Select the No end date option.
9. In the PatternUnit field, enter 'Days'.
10. In the Per field, enter a number.
11. Click OK.
12. Click OK.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and run job to post statements

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through configuring and running a recurrent batch job to post statements for a selected store or group of stores. This procedure uses the USRT company in demo data.

1. Go to All workspaces > .. > Store financials.
2. Click Post statements in batch.
  - Select an organizational hierarchy and then in the organization nodes tree, select either an individual store or a node. Select a node if you want to create the batch job for a group of stores.
  - Click the arrow to add your selection.
3. Click the Run in the background tab.

**Choose organization nodes**

**General** **Run in the background**

Organization hierarchy  
Retail Stores by Region

**AVAILABLE ORGANIZATION NODES:**

- Contoso Retail
  - Contoso Retail USA
    - East
      - Annapolis
      - Atlanta
      - Atlantic City
      - Boston
      - Burlington
      - Cambridge

**SELECTED ORGANIZATION NODES:**

- Contoso Retail USA

4. Check or uncheck the Batch processing checkbox.

## Choose organization nodes

General Run in the background

Recurrence Alerts

Batch processing  
Yes

Task description  
il statement post batch scheduler

Batch group

Private  
No

Critical Job  
No

Monitoring category

Start date: 7/29/2019 (11:34:24 am) (GMT-08:00) Pacific Time (US & Canada)

5. Click Recurrence.
6. In the Start date field, enter a date.
7. In the Start time field, enter a time.
  - Choose whether you want to end the recurrence after a specific number of runs, at a specific date, or never. Then choose the various options to define how frequently you want the job to run.
8. Click OK.
9. Click OK.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Store configurations for Retail statements

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through configurations for the store that affect how Commerce statements get created and posted. Financial dimensions on stores are covered in another procedure. This procedure uses the USRT demo company.

1. In the **Navigation pane**, go to **Modules > Retail and Commerce > Channels > Stores > All stores**.
2. In the list, find and select the desired record.
3. In the list, click the link in the selected row.
4. Click **Edit**.
5. The settings in the **Statement/closing** FastTab affect the statement creation, validation, and posting for the store. Expand the **Statement/closing** FastTab.
6. In the **Statement method** field, select the method you want to use to group the statement lines by.
7. Select "Yes" in **One statement per day** if there should only be one statement created per day when creating statements from the statement creation batch job.
8. The **Tender declaration calculation** field defines whether tender declarations should be added together or if the last one should be used.
9. In the **Rounding** field, select the ledger account to post rounding differences into.
10. In the **Maximum rounding difference** field, enter the maximum rounding difference allowed.
11. In the **Posting** field, enter the maximum total posting difference allowed for a statement.
12. In the **Shift** field, enter the maximum total difference within a shift in a statement.
13. In the **Transaction** field, enter the maximum total difference in a statement line.
14. In the **Closing method** field, define whether transactions that will be included in a statement should be part of a closed shift or if they can be any transactions within the defined date/time range.
15. In the **End of business day** field, enter a time if transactions that happen after midnight should be posted with the previous day.
16. Select "Yes" in **Post as business day** if transactions that happen after midnight should be posted as part of the previous day.
17. Select "Yes" in **Split by Statement method** to get statements created for each statement method defined. This action can be useful if the performance of the posting needs to be improved for stores with high transaction volumes since it will create many smaller statements that can be processed in parallel.
18. In the **General** FastTab, in the **Default customer** field, you can select the customer account to use for sales to walk-in customers.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Posting of online sales and payments

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure walks through configuring and running a recurrent batch job to create sales orders and payments for online store transactions.

Posting online sales and payments is a two-stage process.

- Pulling the online commerce transaction data in HQ.
- Synchronizing orders to create sales orders in HQ.

Pulling the online transaction data can be done either by manually running the P-job or by creating a recurrent batch job.

## **Manually running the P-job**

1. Go to All workspaces > Retail and Commerce IT.
2. Click Distribution schedule.
3. Select P-0001.
4. Adjust channel database groups, if required.
5. Click Run now.
6. Click Yes.

## **Scheduling a recurring P-job**

1. Go to All workspaces > Retail and Commerce IT.
2. Click Distribution schedule.
3. Select P-0001.
4. Click Create batch job.
5. Click Run in the background.
6. Enable Batch processing.
7. Click Recurrence..
8. Select the No end date option.
9. In the Count field, enter interval between the runs in minutes. Typically this would be 5-10.
10. Click OK.
11. Click OK.

Orders can be synchronized either by manually running the "Synchronize orders"-job or by creating a recurring batch job.

## **Manually running order synchronization**

Follow these steps to manually run "Synchronize orders" job once.

1. Go to All workspaces > Store financials.
2. Click Synchronize orders.
3. In the Organization hierarchy field, select 'Stores by Region'.
  - Select either a specific online store, or select a node if you want to create the batch job for a group of stores.
  - Click the arrow to add your selection.
4. Click the Run in the background tab.
5. Disable Batch processing



6. Click Recurrence.
7. Select End After option
8. In the End After field, enter 1.
9. Click OK.
10. Click OK.

### **Scheduling recurring order synchronization**

This procedure walks through configuring and running a recurrent batch job to create sales orders and payments for online store transactions. This procedure uses the USRT company in demo data.

1. Go to All workspaces > Store financials.
2. Click Synchronize orders.
3. In the Organization hierarchy field, select 'Stores by Region'.
  - Select either a specific online store, or select a node if you want to create the batch job for a group of stores.
  - Click the arrow to add your selection.
4. Click the Run in the background tab.
5. Enable Batch processing
6. Click Recurrence.
7. Select the No end date option.
8. In the Count field, enter interval between the runs in minutes. Typically this would be 2-20
9. Click OK.
10. Click OK.

## Data entities involved in the process

- RetailTransactionTable
- RetailTransactionAddressTrans
- RetailTransactionInfocodeTrans
- RetailTransactionTaxTrans
- RetailTransactionSalesTrans
- RetailTransactionTaxMeasure
- RetailTransactionDiscountTrans
- RetailTransactionTaxTransGTE
- RetailTransactionMarkupTrans
- RetailTransactionPaymentTrans
- RetailTransactionAttributeTrans

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Task management overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of task management for managers and workers in Microsoft Dynamics 365 Commerce.

## Overview

In a retail environment, it's always difficult to make sure that tasks are performed by the right person at the right time. Retailers must be able to notify workers about upcoming tasks and provide related business context, so that the tasks can be completed correctly and on time.

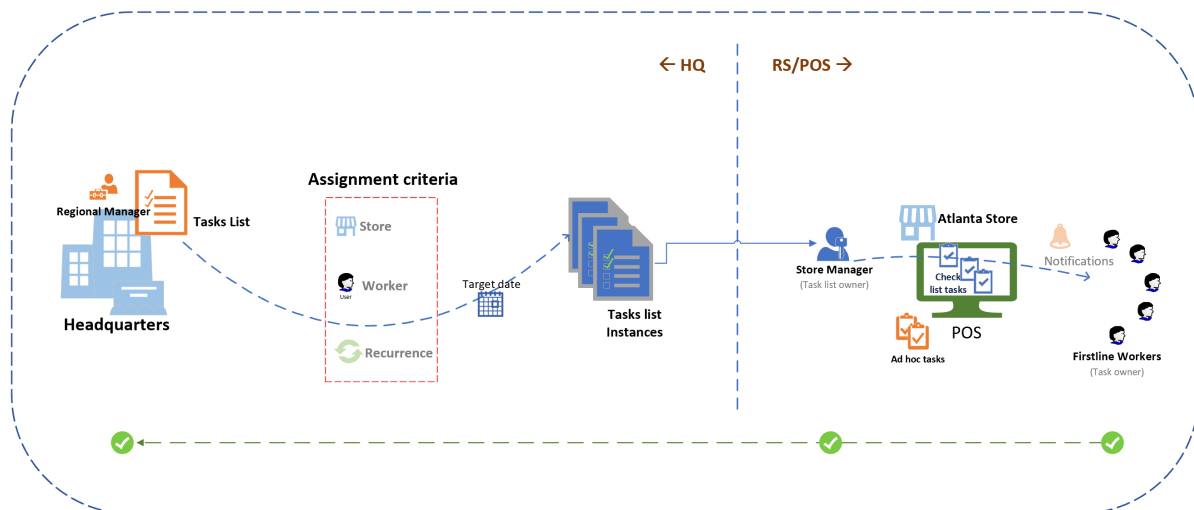
Task management is a productivity feature in Dynamics 365 Commerce that lets managers and workers create task lists, manage assignment criteria, track task status, and integrate these operations between Commerce back office and point of sale (POS) applications.

Headquarters personas can use task management to create task lists for retail stores, and to track status by store or worker. They can also create recurrent tasks (for example, "Thursday night closing checklist").

Store managers can use task management to assign tasks to individual workers, send notifications about upcoming tasks or tasks that are past due, update task status, and create single-purpose tasks in the POS application. Workers can then see notifications, view task details, and update task status at the POS.

The following illustration shows the conceptual architecture of task management in Commerce.

### Dynamics 365 Commerce - Task Management Conceptual Architecture



## Additional resources

[Configure task management](#)

[Create task lists and add tasks](#)

[Assign task lists to stores or employees](#)

[Task management in POS](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure task management

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to configure task management features in Microsoft Dynamics 365 Commerce.

## Overview

Before Dynamics 365 Commerce managers and employees can use the task management features in Commerce, task management must be configured. Configuration steps include granting permissions to managers and employees, distributing permissions to point of sale (POS) clients, setting up POS notifications, and configuring the **Tasks** tile on the home page of a POS application.

## Configure permissions for store managers

Every worker in a given store can view all tasks that are assigned to that store. They can also update the status of the tasks that are assigned to them. However, personas such as store managers must have task management permissions to manage tasks that are assigned to the store and to create single-purpose tasks.

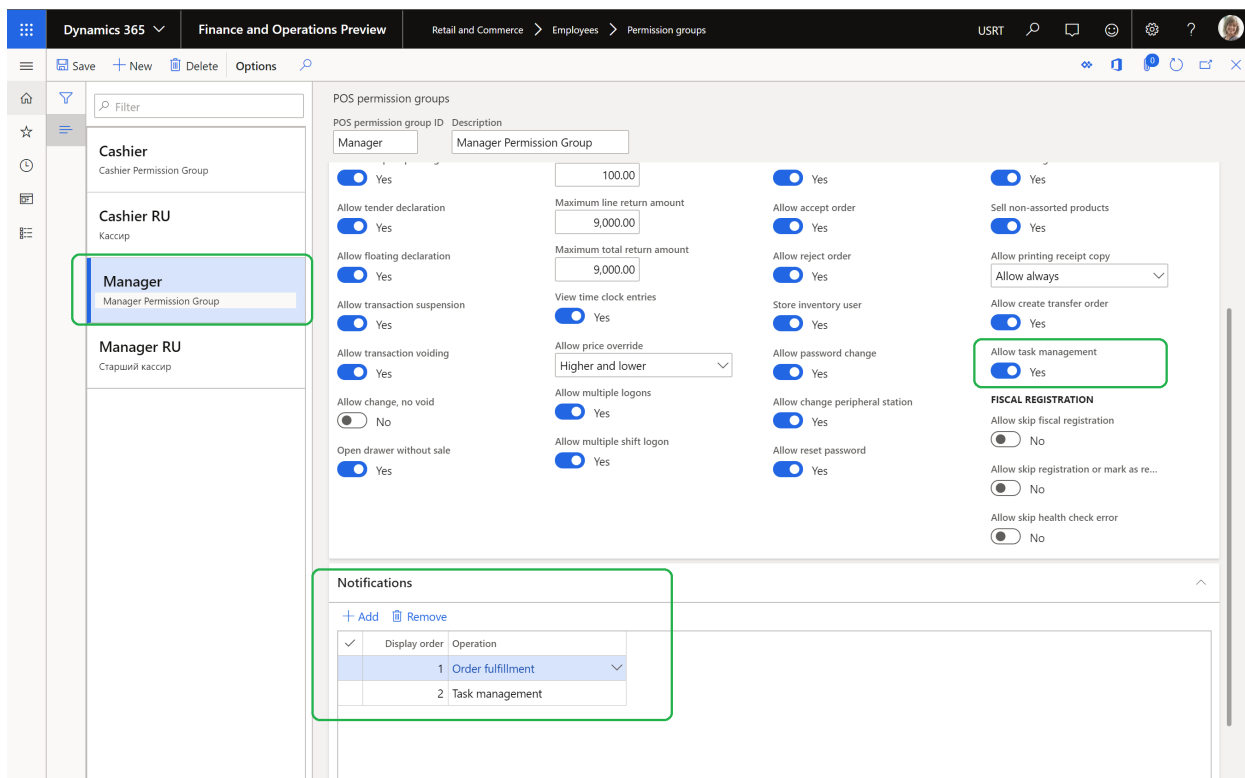
To configure task management permissions for store managers, follow these steps.

1. Go to **Retail and Commerce > Employees > Permission groups**.
2. Select a specific permission group (for example, **Manager**), and then select **Edit**.
3. On the **Permissions** FastTab, set the **Allow task management** option to **Yes**.
4. On the **Notifications** FastTab, add the **Task management** operation, and enter a value in the **Display order** field. For example, enter 2 if the **Order fulfillment** operation already has a **Display order** value of 1.

### NOTE

If a non-manager persona must have task management permissions in the POS, you can grant permission to the individual. Alternatively, you can create a new permission group for non-managers and set the **Allow task management** option to **Yes**.

The following illustration shows how to configure task management permissions for store managers.



## Configure permissions for employees

Employees must have permissions to create task lists, manage assignment criteria, and configure the recurrence of any task list. To configure these permissions, you assign employees to the **Retail task manager** role.

To configure permissions for an employee, follow these steps.

1. Go to **Retail and Commerce > Employees > Users**.
2. Select an employee.
3. On the **User's roles** FastTab, select **Assign roles**.
4. In the **Assign roles to user** dialog box, select the **Retail task manager** role, and then select **OK**.

## Distribute permissions to POS clients

Before employees can use POS clients, permissions must be distributed and synced to those clients.

To distribute permissions to POS clients, follow these steps.

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Select the **1060 (Staff)** distribution schedule, and then select **Run now**.
3. Select the **1070 (Channel configuration)** distribution schedule, and then select **Run now**.

## Configure POS notifications for tasks

Task management must be configured so that notifications are available in the POS application.

To configure POS notifications for tasks, follow these steps.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS > POS operations**.
2. Find operation **1400 (Task management)**, and select the **Enable notifications** check box for it.

The following illustration shows the **Task management** operation on the **POS operations** page.

Operation ID	Operation name	Permission ID	Permission ID2	Check user acc...	User operati...	Enable notificat...	Allow task link
1218	Force unlock peripheral	1002	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1219	Open URL	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1220	Manage safes	1023	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1221	Void suspended transactions	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1300	Skip fiscal registration	1033	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1301	Mark fiscal event registered	1033	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1302	Complete fiscal registration pro...	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1303	Skip health check error	1034	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1400	Task management	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2000	Schedule management	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2001	Schedule requests	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3500	Add customer information	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6000	Allow skip fiscal registration	1031	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1217	Disassemble kits	1024	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

For more information about how to configure POS notifications, see [Show order notifications in the point of sale \(POS\)](#).

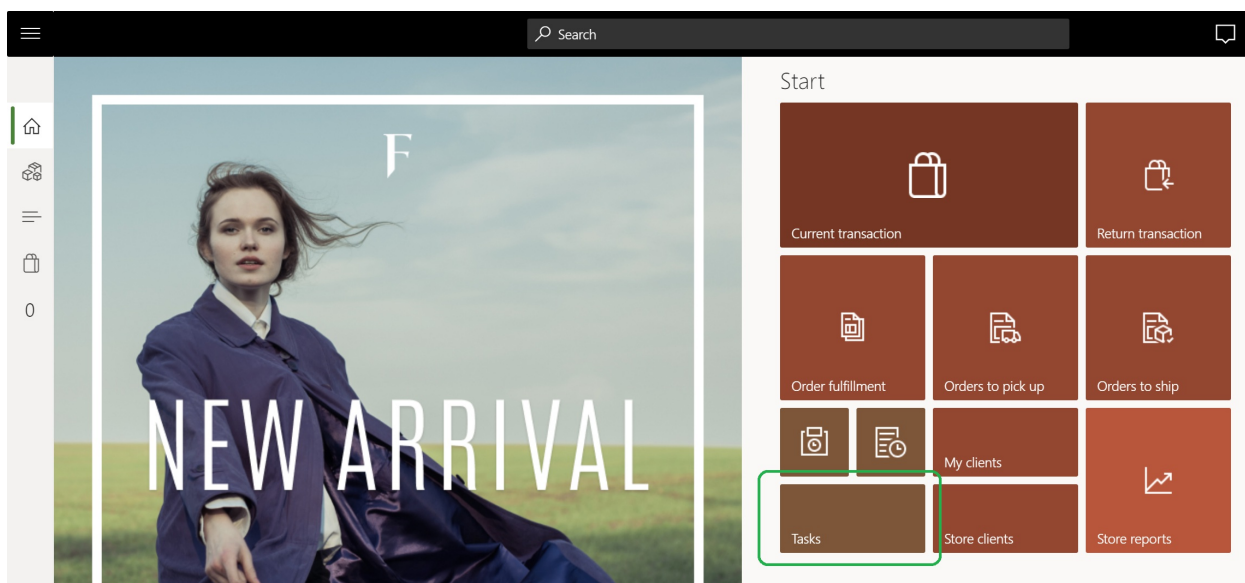
## Configure the Tasks tile on a POS application home page

Before you configure the **Tasks** tile on the home page of a POS application, see [Screen layouts for the point of sale \(POS\)](#) for information about how to configure and add new buttons to a POS screen layout.

To configure the **Tasks** tile on a POS application home page, follow these steps.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS > Screen layouts**.
2. Select a screen layout, select a layout size, and select a button grid.
3. On the **Button grids** FastTab, select **Designer** to edit the selected button grid.
4. Add a **Tasks** tile to the appropriate section of the home page.

The following illustration shows an example of a **Tasks** tile on a POS home page.



## Additional resources

[Task management overview](#)

[Create task lists and add tasks](#)

[Assign task lists to stores or employees](#)

[Task management in POS](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create task lists and add tasks

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to create task lists and add tasks to them in Microsoft Dynamics 365 Commerce.

## Overview

A *task* defines a specific piece of work or an action that someone must complete on or before a specified due date. In Dynamics 365 Commerce, a task can include detailed instructions and information about a contact person. It can also include links to back-office operations, point of sale (POS) operations, or site pages, to help improve productivity and provide the context that the task owner requires to complete the task efficiently.

A *task list* is a collection of tasks that must be completed as part of a business process. For example, there might be a task list that a new worker must complete during onboarding, a task list for cashiers who work evening shifts, or a task list that must be completed to prepare the store for an upcoming holiday season. In Commerce, every task list that has a target date can be assigned to any number of stores or employees, and it can be configured to recur.

Both managers and workers can create task lists in Commerce back office, and then assign them to a set of stores.

## Create a task list

To create a task list, follow these steps.

1. Go to **Retail and Commerce > Task management > Task management administration**.
2. Select **New**, and then enter values in the **Name**, **Description**, and **Owner** fields.
3. Select **Save**.

## Add tasks to a task list

To add tasks to a task list, follow these steps.

1. On the **Tasks** FastTab of an existing task list, select **New** to add a task.
2. In the **Create a new task** dialog box, in the **Name** field, enter a name for the task.
3. In the **Due data offset from target date** field, enter a positive or negative integer value. For example, enter -2 if the task should be completed two days before the task list's due date.
4. In the **Notes** field, enter detailed instructions.
5. In the **Contact person** field, enter the name of a subject matter expert that the task owner can contact if he or she needs help.
6. In the **Task link** field, enter a link, based on the nature of the task.

### TIP

Although you can use the **Assigned to** field to assign tasks to someone while you're creating a task list, we recommend that you avoid assigning tasks during task list creation. Instead, assign the tasks after the list is instantiated for individual stores.

## Use task links to help improve worker productivity

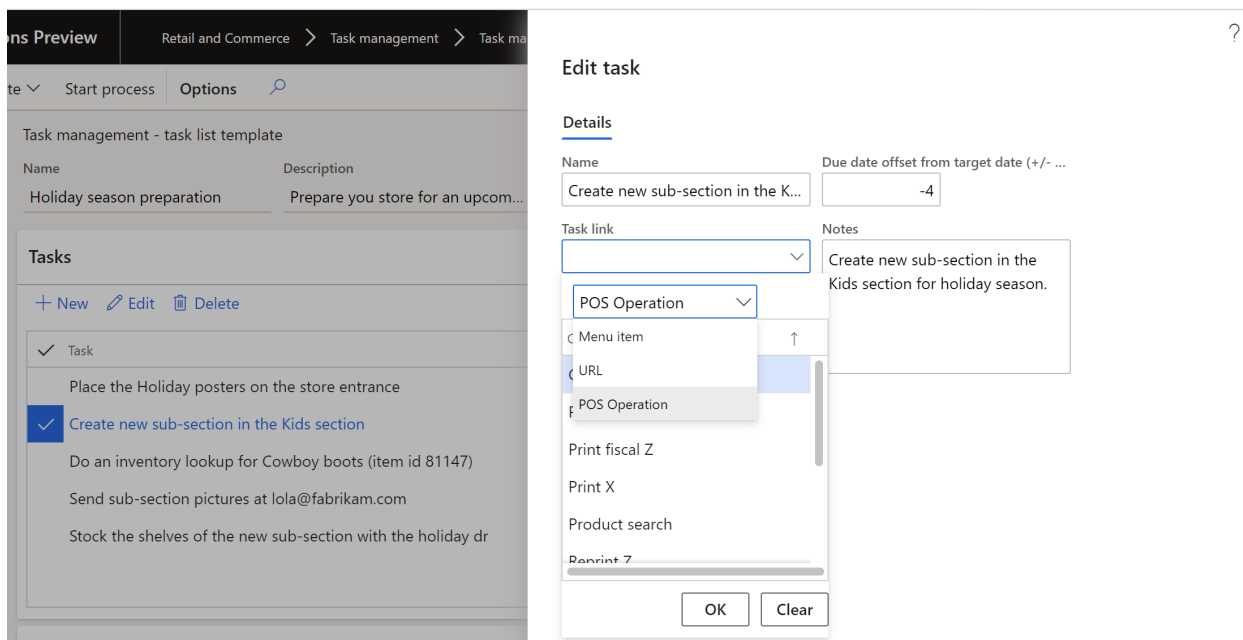


Commerce lets you link tasks to specific POS operations, such as running a sales report, viewing an online training video for new employee orientation, or performing a back-office operation. This feature helps task owners get the information that they need to complete a task efficiently.

To add task links while you create a task, follow these steps.

1. On the **Tasks** FastTab of an existing task list, select **Edit**.
2. In the **Edit task** dialog box, in the **Task link** field, select one or more of the following options:
  - Select **Menu item** to configure a back-office operation, such as "Product kits."
  - Select **POS Operation** to configure a POS operation, such as "Sales reports."
  - Select **URL** to configure an absolute URL.

The following illustration shows the selection of task links in the **Edit task** dialog box.



### Configure a POS operation so that it can be linked to a task

To configure a POS operation so that it can be linked to a task, follow these steps.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS > POS operations**.
2. Select **Edit**, find the POS operation, and then select the **Enable Task Management** check box for it.

## Additional resources

[Task management overview](#)

[Configure task management](#)

[Assign task lists to stores or employees](#)

[Task management in POS](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Assign task lists to stores or employees

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to assign task lists to stores or employees in Microsoft Dynamics 365 Commerce.

## Overview

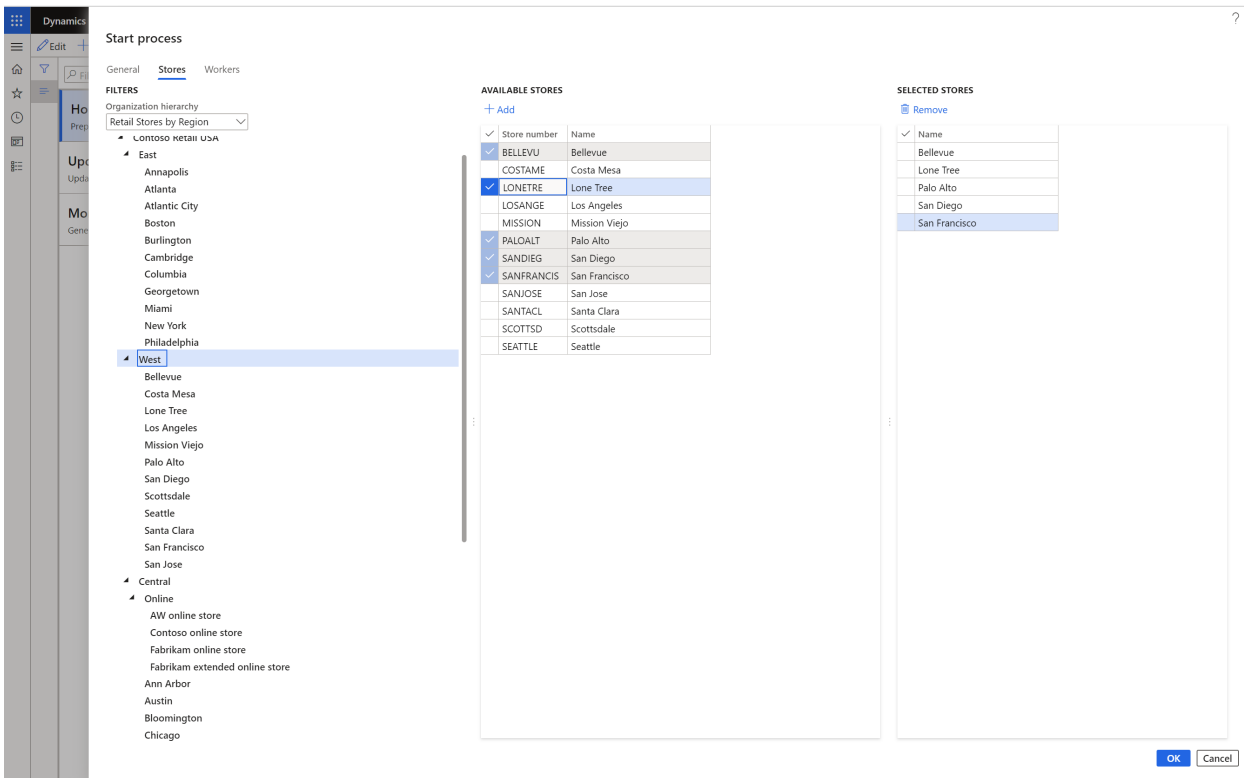
Task management in Dynamics 365 Commerce lets you assign a task list to multiple stores or employees, or to a combination of stores and employees. For example, a regional manager for 20 stores might want to assign the **Holiday season preparation** task list to all 20 stores.

## Start the task list assignment process

To start the process of assigning a task list, follow these steps.

1. Go to **Retail and Commerce > Task management > Task management administration**.
2. Select the task list to assign.
3. Select **Start process**.
4. In the **Start process** dialog box, on the **General** tab, in the **Process name** field, enter a name (for example, **East region stores**).
5. In the **Target date** field, specify a date.
6. To assign the task list to stores, on the **Stores** tab, use the **Organization hierarchy** filter to find and select the stores.  
  
To assign the task list to employees, on the **Workers** tab, find and select the employees.
7. Select **OK** to start the process. The task list is assigned to the selected stores or employees.

The following illustration shows an example of how to find and select stores in the **Start process** dialog box.

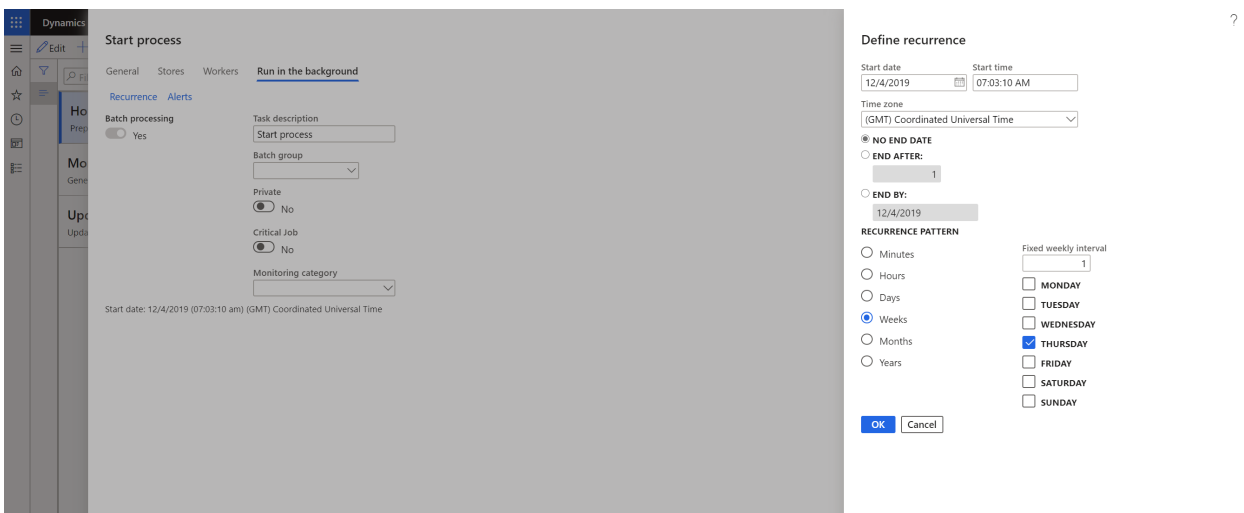


## Assign task lists on a recurring basis

Retailer sometimes have recurrent tasks, such as "Thursday closure checklist" or "First day of the month checklist." Therefore, they might want to assign the task list on a recurring basis.

1. Go to **Retail and Commerce > Task management > Task management administration.**
2. Select the task list to assign.
3. Select **Start process.**
4. In the **Start process** dialog box, on the **General** tab, in the **Process name** field, enter a name.
5. Set the **Recurrence** option to **Yes.**
6. In the **Recurrence target date offset in days** field, enter a number of days. For example, if you enter **4**, the target date is the recurrence date plus four days.
7. On the **Run in the background** tab, select **Recurrence.**
8. In the **Define recurrence** dialog box, enter the frequency criteria, and then select **OK.**

The following illustration shows an example of how to enter frequency criteria in the **Define recurrence** dialog box.



## Track task list status

If you're a regional manager or store manager, you might want to track the status of task lists that have been assigned to multiple stores or employees. You can then follow up with stores or workers that didn't complete their assigned tasks on time. Commerce back office lets you view the status of task lists, reassign tasks, or change the status of a task.

To track the task list status for all tasks, follow these steps.

1. Go to **Retail and Commerce > Task management > Task management processes**.
2. Select the **All task lists** tab to view the status of all task lists that are assigned to various stores.

To track the task list status for all tasks that are assigned to you, follow these steps.

1. Go to **Retail and Commerce > Task management > Task management processes**.
2. Select the **My tasks** or **All tasks** tab to view or update the status of tasks that are assigned to you.

## Additional resources

[Task management overview](#)

[Configure task management](#)

[Create task lists and add tasks](#)

[Task management in POS](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Task management in POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes task management in the Microsoft Dynamics 365 Commerce point of sale (POS) application.

## Overview

The Dynamics 365 Commerce POS application has task management features that let store managers and workers manage tasks and update task status. Store workers can access tasks either by selecting the **Tasks** tile on the POS home page or by selecting task notifications. By default, store workers are taken to the **My tasks** tab, where they can view the tasks that are assigned to them. However, they can easily switch to the **Overdue tasks**, **Open tasks**, and **Task lists** tabs.

## Task operations for store managers

Store managers can perform the following task operations in the POS application by using the buttons on the command bar:

- **Assign** – Assign selected tasks to a store worker.
- **Task status** – Change the status of selected tasks.
- **Filter** – By default, only active tasks are shown. However, by applying filters, managers can view all tasks, even tasks that have been completed or canceled.
- **New task** – Create a task under an existing task list, or create an single-purpose task.

Store workers can perform the following task operations in the POS application by using the buttons on the command bar:

- **Task status** – Change the status of selected tasks.
- **Filter** – By default, only active tasks are shown. However, by applying filters, workers can view all tasks, even tasks that have been completed or canceled.

The following illustration shows the **My tasks** tab in the Commerce POS application.

PRIORITY	TASK NAME	DUE DATE	STATUS	TASK LIST
1	Task name	10/04/2019 5:00 PM	In progress	Task list 1
1	Stock the shelves of the new sub section...	10/04/2019 5:00 PM	Completed	Task list 2
1	Task name	10/04/2019 5:00 PM	In progress	Ad hoc
2	Task name	10/04/2019 5:00 PM	In progress	Task list 1
2	Task name	10/04/2019 5:00 PM	Completed	Task list 2
3	Task name	10/04/2019 5:00 PM	Completed	Task list 1
4	Task name	10/04/2019 5:00 PM	Not started	Task list 1
4	Task name	10/04/2019 5:00 PM	In progress	Task list 1
5	Task name	10/04/2019 5:00 PM	Overdue	Task list 1
5	Task name	10/04/2019 5:00 PM	Not started	Task list 1

**Details**

**PRIORITY 1**  
Stock the shelves of the new sub section and put up decorations

In progress

Assigned to  
**FL Alex Eggerer**

Task list  
Mohan - Houston 2

Instructions  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Lorem ipsum dolor sit amet, consectetur Quis nostrud exercitation ullamco laboris nisi ut

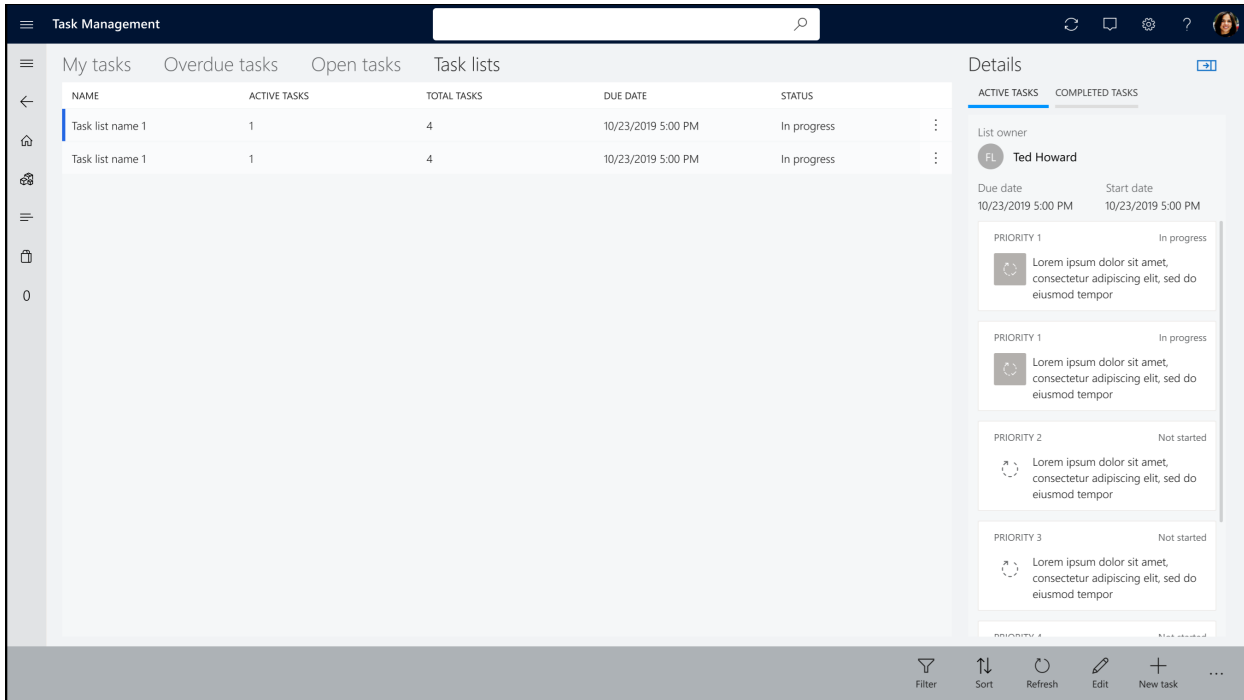
Location  
**Houston**

Contact person  
**FL Cameron Levigne**

Task link - URL  
[View X-Report](#)

Command bar: Select, Assign, Task status, Filter, Sort, Refresh, New task

The following illustration shows the **Task lists** tab.



## Additional resources

[Task management overview](#)

[Configure task management](#)

[Create task lists and add tasks](#)

[Assign task lists to stores or employees](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# E-commerce site overview

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic provides an overview of the support for e-commerce sites in Microsoft Dynamics 365 Commerce. It includes information about how e-commerce online stores are initialized and managed in Dynamics 365 Commerce. It also provides links to more information about online stores, and about how to set up and configure an e-commerce site. Although this topic covers many of the basics, it doesn't cover everything that is required to set up a production e-commerce site. More advanced topics can be found in the Dynamics 365 Commerce documentation.

## Online store channel

Before you can build your site in Dynamics 365 Commerce, at least one online store channel must be set up. For more information, see [Set up an online channel](#).

In Dynamics 365 Commerce, you use an online store channel to establish the products, pricing, languages, payment methods, delivery modes, fulfillment centers, and other aspects of the online experience that should be available to your customers.

Only one online store channel has to be set up before you can get started with Dynamics 365 Commerce. However, a single e-commerce site can provide the online experience for multiple online stores. For example, if multiple online stores are set up to support different geographical regions, a single set of e-commerce pages can be used to provide the unique experiences that are defined by each store. For more information about how to configure a site to support multiple online stores, see [Associate an online site with a channel](#).

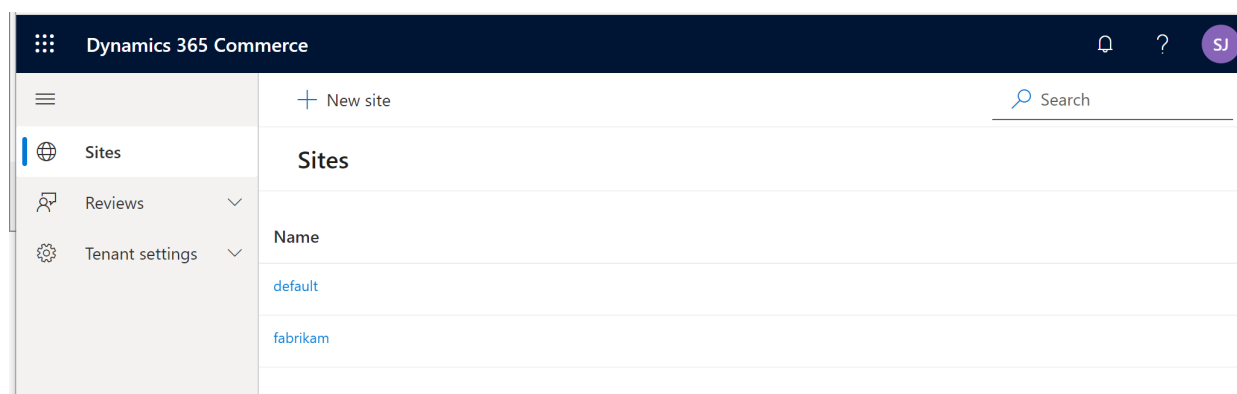
After an online store is set up, it can be associated with the Dynamics 365 Commerce site that will serve as your online storefront. For more information about online stores and how to set them up, see [Set up online stores](#).

## Deploy a new e-commerce tenant

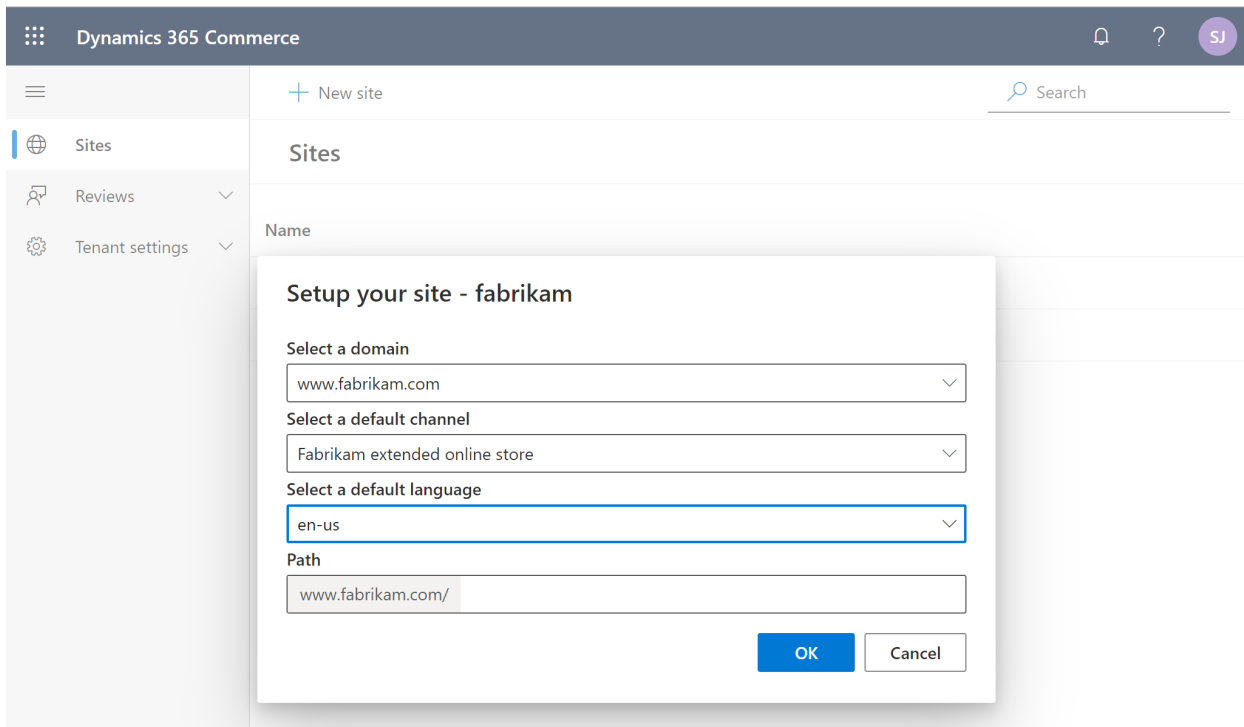
During initialization of an e-commerce site, you're prompted for a domain name. For more information about domains in Commerce, see [Configure your domain name](#) and [Domains in Dynamics 365 Commerce](#). To deploy a new e-commerce tenant by using [Microsoft Dynamics Lifecycle Services \(LCS\)](#), follow the steps in [Deploy a new e-commerce tenant](#). After your e-commerce tenant is set up in LCS, a link to Commerce site builder will be provided. You can then use Commerce site builder to initialize and configure your e-commerce sites.

## Initialize your e-commerce site

When you start Commerce site builder from LCS, the **Sites** page appears. This page includes two preconfigured sites, **default** and **fabrikam**, as shown in the example in the following illustration.



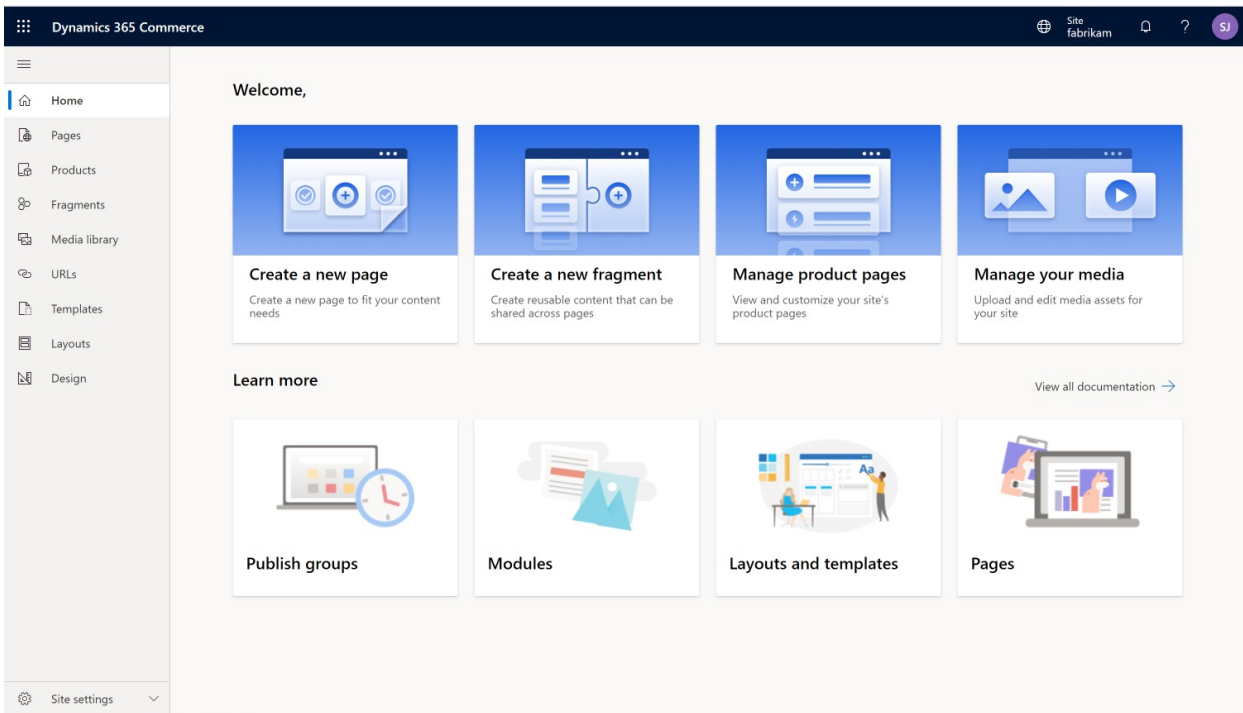
When you select one of these sites, you're prompted to select a domain name, a default online store channel, a supported language for the selected channel, and a path. If only one channel is used, you can leave the path blank. More online store channels or languages can be configured later in Commerce site builder. Each additional channel or language will require a unique path. For example, you have two online channels that are associated with a single site, and the domain name for the site is `www.fabrikam.com`. In this case, the path for one channel can be the default value that has no path (`https://www.fabrikam.com`), and the second channel can be set to a new path, such as `site2`, that will have the URL `https://www.fabrikam.com/site2`. The following illustration shows an example of a site initialization dialog box in Commerce site builder.



The **Sites** page also includes a **New site** button. The dialog box that appears when you select this button resembles the site initialization dialog box, but it's used to create a new site. New sites are blank. They don't include the same default templates, fragments, pages, and images that are provided with the **default** and **fabrikam** sites. However, as you require, you can open a support ticket to request that a copy of the default content be added to a new blank site. For more information, see [Create an e-commerce site](#).

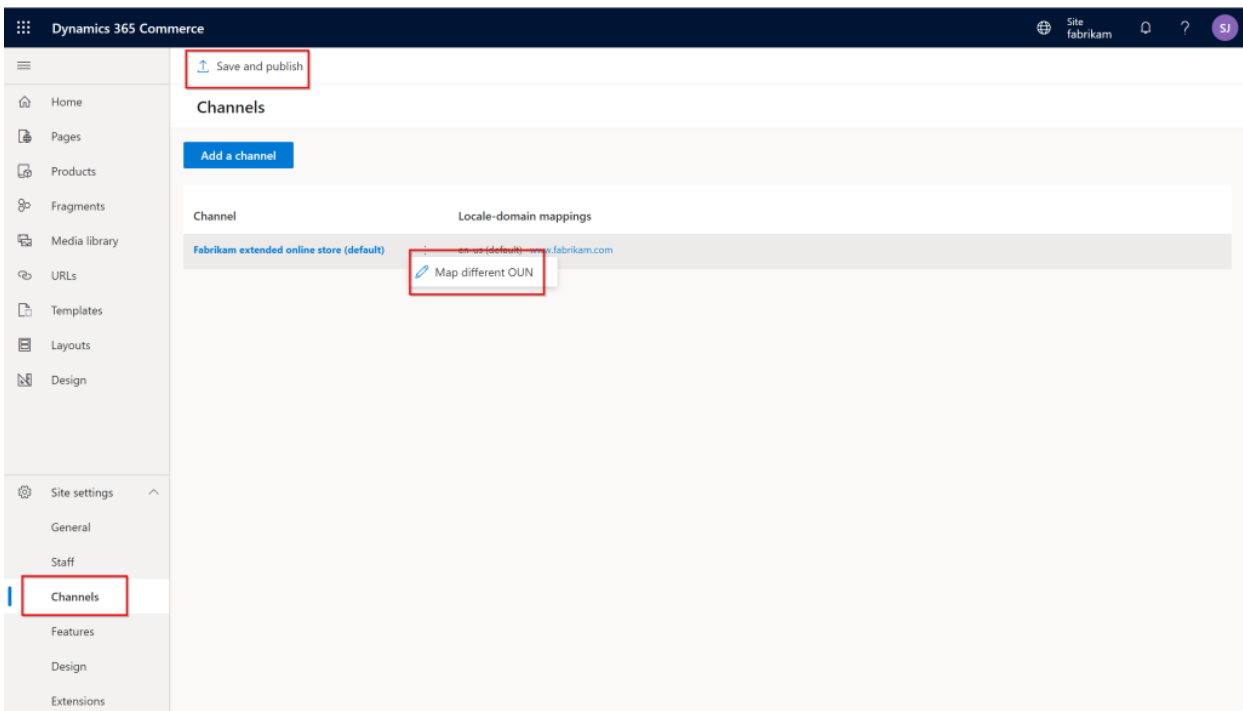
After a new site is initialized, the Commerce site builder **Home** page appears. This page includes links to common actions and guidance content, as shown in the example in the following illustration.



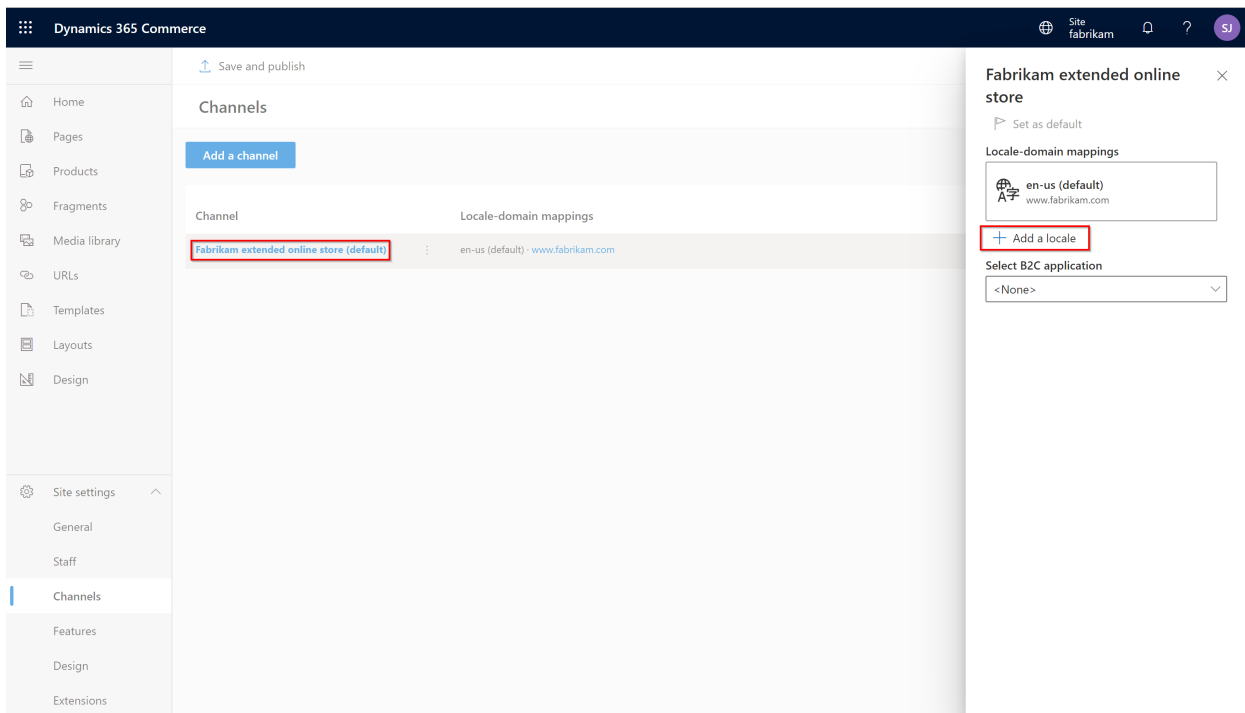


## Modify online store channels or add online store channels to an e-commerce site

After an e-commerce site is created, you can change the channel that it's associated with by following the steps in [Associate an e-commerce site with an online channel](#). The example in the following illustration shows how a channel operating unit number (OUN) can be changed on the **Channels** page (**Site settings** > **Channels**). After you've finished making a change, be sure to select **Save and publish**. In this way, you ensure that the change is published.



You can add new channels by selecting **Add a channel**. To add new languages to a channel, select the channel, and then select **Add a locale** in the channel dialog box that appears. Before locales can appear in the dialog box, they must be preconfigured for the online store channel in Commerce headquarters.



## Set up an Azure B2C tenant

Dynamics 365 Commerce uses Azure Active Directory (Azure AD) business-to-consumer (B2C) to support user credential and authentication flows. For information about how to set up your Azure B2C tenant, see [Set up a B2C tenant in Commerce](#), [Set up custom pages for user sign-ins](#), and [Configure multiple B2C tenants in a Commerce environment](#).

## Overview of the default site pages

The default and fabrikam sites include preconfigured templates, fragments, and pages to help you get started. For more information, see the following topics:

- [Home page overview](#)
- [Product details page overview](#)
- [Cart and checkout pages overview](#)
- [Account management pages overview](#)

## Manage site settings

For information about how to manage your site settings, see the following topics:

- [Manage e-commerce users and roles](#)
- [Search engine optimization \(SEO\) considerations for your site](#)
- [Manage Content Security Policy \(CSP\)](#)
- [Select a site theme](#)

## Manage site content

For information about how to manage site content, see the following topics:

- [Page model glossary](#)
- [Document states and lifecycle](#)
- [Templates and layout](#)
- [Work with fragments](#)

- [Work with modules](#)
- [Digital asset management overview](#)
- [Module library overview](#)

## Additional resources

[Create an e-commerce site](#)

[Deploy a new e-commerce site](#)

[Associate an online site with a channel](#)

[Configure your domain name](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

[Set up custom pages for user logins](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure your domain name

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to configure a domain name for a Microsoft Dynamics 365 e-commerce site.

## Add domains during e-commerce initialization

To associate domains with your Dynamics 365 Commerce e-commerce environment, initialize e-commerce as described in [Deploy a new e-commerce tenant](#). During initialization, you're asked to provide information that will be used to provision your e-commerce environment. In the **Supported host names** field, add all the domains that you plan to use with this environment. Multiple domains should be separated with semi-colon. In this way, the domains are configured in all the required Commerce components, and they are ready to be used when you switch traffic from your content delivery network (CDN) or web server and point it to the e-commerce front ends.

## Add domains after e-commerce initialization

To associate new domains with your e-commerce environment after e-commerce initialization, you must submit a service request.

## Additional resources

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Deploy a new e-commerce tenant

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to deploy a new Dynamics 365 Commerce e-commerce site by using Microsoft Dynamics Lifecycle Services (LCS).

## Overview

Microsoft Dynamics Lifecycle Services (LCS) is a cloud-based collaborative workspace that partners and customers can use to manage their projects and environments, view the latest information about Microsoft Dynamics products and features, and create, track, and browse support incidents. E-commerce management features are integrated into LCS.

To learn more about LCS, see the [Lifecycle Services User Guide](#).

## Get started

Before you can initialize e-commerce, you must initialize a project, an environment, and a Retail Cloud Scale Unit (RCSU). To do the initialization in LCS, you must have permissions for either the Project Owner or Environment manager role. The production and sandbox environment topologies are supported.

For more information about environments, see [Environment planning](#). For more information about RCSU, see [Initialize Retail Cloud Scale Unit](#).

## Initialize e-commerce

Use this procedure to initialize the e-commerce feature in an existing environment.

Before you begin, make sure that you have the following required information:

- The RCSU that will be used.
- The Microsoft Azure Active Directory security group that will be used for e-commerce system admins.
- The Microsoft Azure Active Directory security group that will be used for ratings and reviews moderators.
- The domains that will be associated with the environment.

In addition, you can collect the following optional information:

- Azure AD business-to-consumer (B2C) information:
  - Tenant Name.
  - Client ID.
  - Login Custom Domain.
  - Reply URL.
  - SignUp SignIn Policy ID.
  - Reset password Policy ID.
  - Edit Profile Policy ID.

### NOTE

This information can be added later, through a service request.

After you've collected the required information, follow these steps to initialize e-commerce.

1. Sign in to [LCS](#).
2. Open the project that contains the environment where you want to initialize e-commerce.
3. In the **Environments** section, select the environment.
4. Under **Environment features**, select the **Retail manage** link.
5. On the **e-Commerce** tab, select **Setup**. A dialog box appears, where you must enter the information that is required for provisioning.
6. Fill in the required information, and then go to the next page.
7. On the next page, fill in the required information, and then submit the form. You're returned to the **e-Commerce** tab, where you should see that initialization has been started.
8. To view the initialization status, either **Refresh** or return to the **e-Commerce** tab later.

When e-commerce is initialized from LCS, the system provisions several components that are required for e-commerce and associates them with the environment. After provisioning is complete, the **e-Commerce** tab on the **Retail management** page is updated to reflect the provisioning. The page shows the latest customization deployments and the status of any other ongoing deployments. It also includes links to the e-commerce site and the Commerce site builder where sites are authored.

## Access Commerce site builder

To access Commerce site builder, go to the **e-Commerce** tab on the **Retail management** page in LCS and select the **e-Commerce site management tool** link. The site builder landing page displays a tenant-level view. From this page, you can:

- Modify tenant-level settings.
- Navigate to any site you have created, and have permission to view.
- Access Reviews features such as moderation and reporting.
- Create a new site. For more information about how to create a new site, see [Create an e-commerce site](#) .

## Additional resources

[Configure your domain name](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create an e-commerce site

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the steps and information required to create a new e-commerce site in Dynamics 365 Commerce site builder.

When you license the Dynamics 365 Commerce capabilities, site builder will be provisioned with a starter site that you can use as a basis for your own site. However, if you want to start from scratch or if you want to establish a second site, you will need to establish a new site in the site authoring environment.

## Set up your site

To set up your site, do the following.

1. Open the site builder environment. You can find a link to site builder in Microsoft Lifecycle Services (LCS) in the environment features page for Commerce.
2. On the home page for the site authoring environment, select **New site**.
3. In the **New site** dialog box, provide the following information.

FIELD	DESCRIPTION
Site name	Enter the display name that should be used for your site in the site authoring environment. This name is visible only in the authoring environment and will not be shown to customers.
Site administrator's security group	Specify the Microsoft Azure Active Directory (Azure AD) security group that manages users who have the site administrator role in this site.
Default channel (operating unit number)	Select the online store that this site will serve as the web storefront for. If you want your e-commerce site to support multiple online stores, you must associate the stores with your site in <b>Site settings</b> after the site is set up.
Default language	Specify the default language for this online store and market. An online store can support multiple languages. If you want to support multiple languages for this online store or another online store, you can configure that support in <b>Site settings</b> after the site is set up.
Domain	Select a domain name that will serve as the domain for this online store. If you haven't configured any domains in LCS, you can leave this field blank. After your domain is configured in LCS, you must add it to your online store in <b>Site settings</b> .



FIELD	DESCRIPTION
Path	When your site supports more than one language for a given domain name, use the path field to create a unique site URL for that domain and language combination. If the language you specified in the <b>Default language</b> field is the only language you will support for this domain, or will continue to be the default language after you have localized your site into additional languages, we recommend that you leave this field empty.

After your site is created, you can verify that it is associated with your online store by selecting the **Products** tab. You should see the assortment of products that has been allocated to the online store. You can also use the drop-down menu in the upper left of the page to access the allocated products by category.

## Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Associate a Dynamics 365 Commerce site with an online channel

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to bind your Microsoft Dynamics 365 Commerce site to one or more online stores.

After you've provisioned your Dynamics 365 Commerce e-commerce environment by using the Microsoft Dynamics Lifecycle Services (LCS) portal, you're ready to establish your first e-commerce website. As part of the initial site creation, you associate the site with an online store that was previously created. This step binds the site to an online channel and lets the site show the navigation hierarchy, products, categories, prices, shipping options, and everything else that you defined in the online store.

To establish a new site and associate an online store with it, in LCS, select the link for the site authoring environment. Then, on the page for the site authoring environment, select **New site**. In the **New site** dialog box, you must provide some basic information about your site. For a complete explanation of the information that you must provide, see [Create a new e-commerce site](#).

After your site is created, you can verify that it's associated with your online store by selecting the **Products** tab. You should see the assortment of products that has been allocated to the online store. You can also use the drop-down field in the upper left of the page to access the products by category.

## Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage robots.txt files

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to manage robots.txt files in Microsoft Dynamics 365 Commerce.

## Overview

The robots exclusion standard, or robots.txt, is a standard that websites use to communicate with web robots. It instructs web robots about any areas of a website that should not be visited. Robots are often used by search engines to index websites.

To exclude robots from a server, you create a file on the server. In this file, you specify an access policy for robots. The file must be accessible via HTTP at the local URL `/robots.txt`. The robots.txt file helps search engines index the content on your site.

Dynamics 365 Commerce lets you upload a robots.txt file for your domain. For each domain in your Commerce environment, you can upload one robots.txt file and associate it with that domain.

For more information about the robots.txt file, visit [The Web Robots Pages](#).

## Upload a robots.txt file

After you've created and edited your robots.txt file according to the [robots exclusion standard](#), make sure that the file is accessible on the computer where you will use the Commerce authoring tools. The file must be named **robots.txt**. For best results, it must be in the format that is noted in the standard. Each Commerce customer is responsible for validating and maintaining the contents of its robots.txt file. To upload a robots.txt file, you must be signed in to Commerce as a system admin.

To upload a robots.txt file in Commerce, follow these steps.

1. Sign in to Commerce as a system admin.
2. In the left navigation pane, select **Tenant Settings** (next to the gear symbol) to expand it.
3. Under **Tenant Settings**, select **Robots.txt**. A list of all the domains that are associated with your environment appears in the main part of the window.
4. Select **Manage** to upload a robots.txt file for a domain in your environment.
5. On the menu on the right, select the **Upload** button (the upward-pointing arrow) next to the domain that is associated with the robots.txt file. A file browser dialog box appears.
6. In the dialog box, browse to and select the robots.txt file that you want to upload for the associated domain, and then select **Open** to complete the upload.

### NOTE

During upload, Commerce verifies that the file is a text file, but it doesn't validate the file's contents.

## Download a robots.txt file

To download a robots.txt file in Commerce, follow these steps.

1. Sign in to Commerce as a system admin.
2. In the left navigation pane, select **Tenant Settings** (next to the gear symbol) to expand it.

3. Under **Tenant Settings**, select **Robots.txt**. A list of all the domains that are associated with your environment appears in the main part of the window.
4. Select **Manage** to download a robots.txt file for a domain in your environment.
5. On the menu on the right, select the **Download** button (the downward-pointing arrow) next to the domain that is associated with the robots.txt file. A file browser dialog box appears.
6. In the dialog box, go to the desired location on your local drive, confirm or enter a file name, and then select **Save** to complete the download.

#### NOTE

This procedure can be used to download only robots.txt files that were previously uploaded via the Commerce authoring tools.

## Delete a robots.txt file

To delete a robots.txt file in Commerce, follow these steps.

1. Sign in to Commerce as a system admin.
2. In the left navigation pane, select **Tenant Settings** (next to the gear symbol) to expand it.
3. Under **Tenant Settings**, select **Robots.txt**. A list of all the domains that are associated with your environment appears in the main part of the window.
4. Select **Manage** to delete a robots.txt file for a domain in your environment.
5. On the menu on the right, select the **Delete** button (the trash can symbol) next to the domain that is associated with the robots.txt file. A file browser window appears.
6. In the file browser window, browse to and select the robots.txt file that you want to delete for the domain, and then select **Open**. A warning message box appears.
7. In the message box, select **Delete** to confirm deletion of the robots.txt file.

#### NOTE

This procedure can be used to delete only robots.txt files that were previously uploaded via the Commerce authoring tools.

## Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upload URL redirects in bulk

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to implement URL redirects in bulk by uploading a redirect comma-separated values (CSV) file in Microsoft Dynamics 365 Commerce.

## Overview

After you've finished creating a new e-commerce site in Dynamics 365 Commerce, and it's ready to go live, you must make the Domain Name System (DNS) switch from your old site to your new site. After you make this switch, the URLs for the new site will likely differ from the URLs for the old site. For specific URLs, traffic that uses the old URL can be redirected to the new URL. In this way, you ensure that site visitors reach the desired location. Redirects help prevent broken links for your customers and help you maintain your established search engine optimization (SEO) results.

Although redirects for individual links can be manually configured in DNS, manual configuration becomes tedious if multiple links must be redirected. To streamline the redirect process, Commerce provides the capability to upload a data file that contains bulk redirect mappings for your site. In the site settings in Commerce site builder, your system admin can upload and publish a CSV file that handles bulk URL redirects.

## Redirect CSV file

To handle URL redirects, Commerce supports a simple but specific CSV file. This file uses the following schema:

```
source URL, target URL, redirect type, case sensitivity
```

Here is an explanation of the elements of this schema:

- **Source URL:** The original URL that must be redirected.
- **Target URL:** The URL that users will be redirected to when they try to go to the source URL.
- **Redirect type:** Set this value to **301** or **302**, depending on the type of redirect that you want to use.
  - The **301** redirect type represents permanent redirects and is the most frequently used redirect type. It's the best option when you must maintain search engine optimization (SEO) rankings.
  - The **302** redirect type represents temporary redirects and is less frequently used. It's the best option when you must maintain link targeting during temporary maintenance or other non-permanent scenarios.
- **Case sensitivity:** Set this value to **true** if your source URL paths are case-sensitive. Case sensitivity will then be honored for the source URL paths. Set this value to **false** if case sensitivity isn't required. If you leave this value blank, the default value of **false** is used.

The following example shows a set of redirect rows in a redirect CSV file.

```
https://www.contoso.com/shop, https://www.fabrikam.com/allstores, 301, true
https://www.contoso.com/news, https://www.fabrikam.com/updates, 301, false
https://www.contoso.com/news, https://www.fabrikam.com/updates, 301
```

### IMPORTANT

The following rules must be followed. Otherwise, the bulk URL redirects won't work correctly.

- **No header is allowed in the CSV file:** The first or topmost row must start with the first redirect row.
- **There must be no circular entries:** The source URL in a row must not be used as the target URL in the same row. You must also avoid implementations where a target URL is linked back as a source URL, either in a different row of the CSV file or in a DNS redirect.
- **Source and target URLs must be in valid URL format:** No spaces or invalid characters can be used in URLs.
- **No query string URLs are supported:** Commerce won't run query strings that are provided as source or target URLs.
- **The CSV file must be in valid CSV format:** The CSV file must have comma-separated values, a separate line for each redirect, and valid file formatting, and it must have no header.

## Upload a redirect CSV file

Redirect CSV files can be uploaded by using Commerce site builder. The person who uploads a redirect CSV file to a site must be an admin for that site.

To upload a redirect CSV file, follow these steps.

1. In Commerce site builder, go to the site that will receive the bulk URL redirects.
2. Go to **Site settings > General**.
3. Under **URL Redirect Mapping**, select **Upload**.
4. In File Explorer, browse to your CSV file, select it, and then select **Open**.
5. Under **URL Redirect Mapping**, set the option to **On** to activate the redirects.
6. On the command bar, select **Save and Publish** to commit the changes. Allow up to 15 minutes for the redirects to take effect.

### IMPORTANT

Only one bulk redirect CSV file can be loaded and active per site at any time.

## Update an uploaded redirect CSV file

A previously uploaded redirect CSV file can be downloaded for reference. Alternatively, after the file is downloaded, it can be edited and then uploaded again.

To update an uploaded redirect CSV file, follow these steps.

1. In Commerce site builder, go to the site that will receive the bulk URL redirects.
2. Go to **Site settings > General**.
3. Under **URL Redirect Mapping**, select **Download**.
4. Save the file to your local computer.
5. Edit the CSV file as appropriate, and then save it.
6. Under **URL Redirect Mapping**, select **Replace**.
7. In File Explorer, browse to the replacement CSV file, select it, and then select **Open**.
8. Under **URL Redirect Mapping**, set the option to **On** to activate the redirects.
9. On the command bar, select **Save and Publish** to commit the changes. Allow up to 15 minutes for the redirects to take effect.

# Turn off bulk redirects

To turn off the bulk redirects in an uploaded redirect CSV file, follow these steps.

1. Create and save a new CSV file that has valid but nonexistent source and target URLs (for example, `https://www.com,https://www.com,301`).
2. In Commerce site builder, go to the site that will receive the bulk URL redirects.
3. Go to **Site settings > General**.
4. Under **URL Redirect Mapping**, select **Replace**.
5. In File Explorer, browse to your new CSV file, select it, and then select **Open**.
6. Under **URL Redirect Mapping**, set the option to **On** to activate the redirects.
7. On the command bar, select **Save and Publish** to commit the changes. Allow up to 15 minutes for the redirects to stop working.

## Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Set up a B2C tenant in Commerce

2/18/2021 • 14 minutes to read • [Edit Online](#)

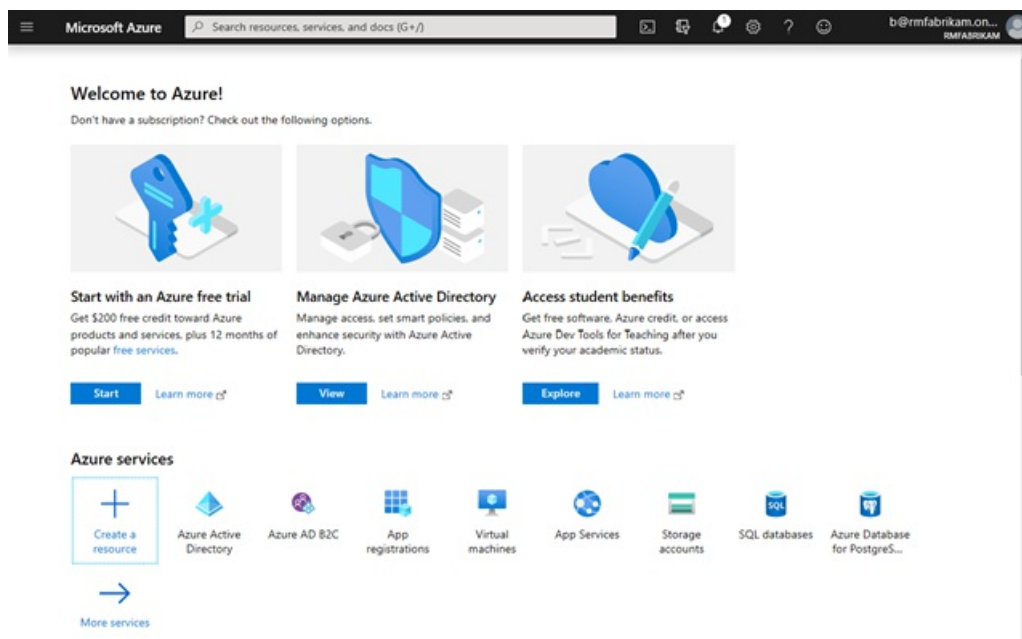
This topic describes how to set up your Azure Active Directory (Azure AD) business-to-consumer (B2C) tenants for user site authentication in Dynamics 365 Commerce.

## Overview

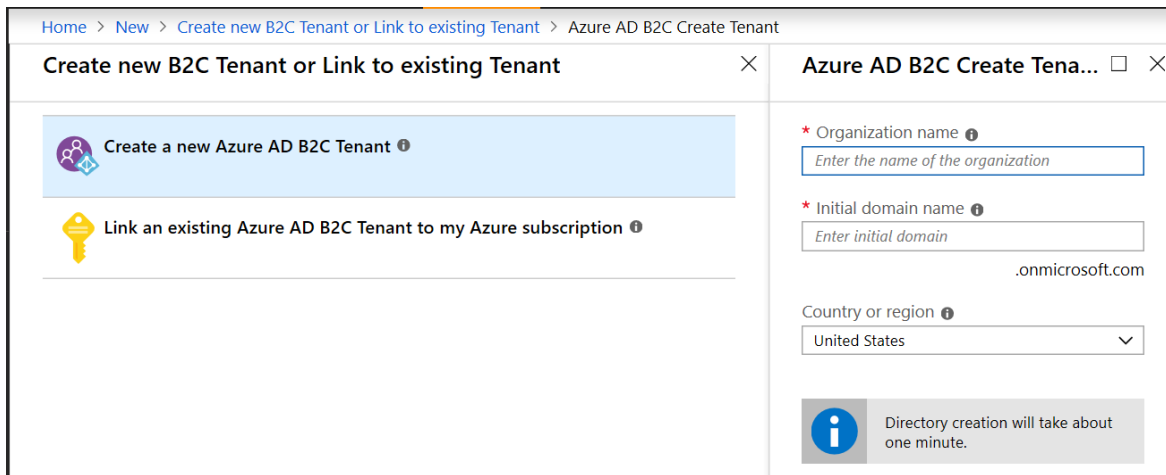
Dynamics 365 Commerce uses Azure AD B2C to support user credential and authentication flows. A user can sign up, sign in, and reset their password through these flows. Azure AD B2C stores sensitive user authentication information, such as username and password. The user record in the B2C tenant will store either a B2C local account record or a B2C social identity provider record. These B2C records will link back to the customer record in the Commerce environment.

## Create or link to an existing AAD B2C tenant in the Azure portal

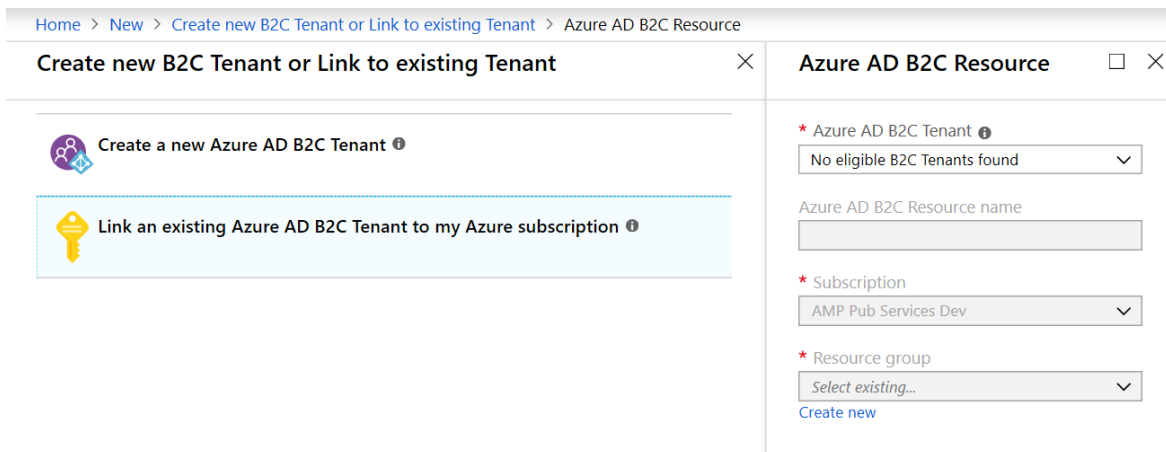
1. Sign in to the [Azure portal](#).
2. From the Azure portal menu, select **Create a resource**. Be sure to use the subscription and directory that will be connected with your Commerce environment.



3. Go to **Identity > Azure Active Directory B2C**.
4. Once on the **Create New B2C Tenant** or **Link to existing Tenant** page, use one of the options below that best suits your company's needs:
  - **Create a new Azure AD B2C Tenant**: Use this option to create a new AAD B2C tenant.
    - a. Select **Create a new Azure AD B2C Tenant**.
    - b. Under **Organization name**, enter the organization name.
    - c. Under **Initial domain name**, enter the initial domain name.
    - d. For **Country or region**, select the country or region.
    - e. Select **Create** to create the tenant.





- **Link an existing Azure AD B2C Tenant to my Azure subscription:** Use this option if you already have an Azure AD B2C tenant you want to link to.
  - a. Select **Link an existing Azure AD B2C Tenant to my Azure subscription**.
  - b. For **Azure AD B2C Tenant**, select the appropriate B2C tenant. If the message "No eligible B2C Tenants found" appears in the selection box, you do not have an existing eligible B2C tenant and will need to create a new one.
  - c. For **Resource group**, select **Create new**. Enter a **Name** for the resource group that will contain the tenant, select the **Resource group location**, and then select **Create**.



5. Once the new Azure AD B2C directory is created (this may take a few moments), a link to the new directory will appear on the dashboard. This link will direct you to the "Welcome to Azure Active Directory B2C" page.

Country or region ⓘ

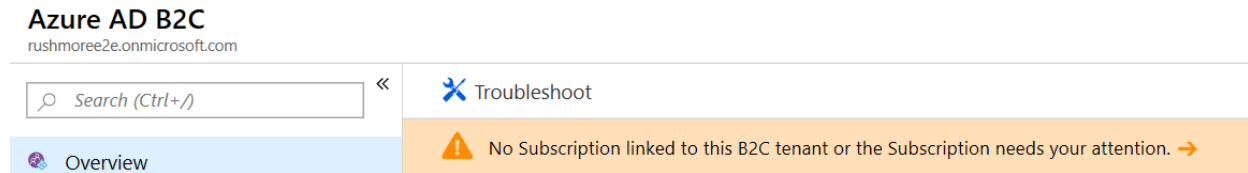
United States ▼

 Click [here](#) to manage your new directory 

## NOTE

If you have multiple subscriptions within your Azure account or have set up the B2C tenant without linking to an active subscription, a **Troubleshoot** banner will direct you to link the tenant to a subscription. Select the troubleshooting message and follow the instructions to resolve the subscription issue.

The following image shows an example of an Azure AD B2C **Troubleshoot** banner.



## Create the B2C application

Once the B2C tenant has been created, you will create a B2C application within the tenant to interact with the Commerce actions.

To create the B2C application, follow these steps.

1. In the Azure portal, select **Applications(Legacy)** and then select **Add**.
2. Under **Name**, enter the name of the desired AAD B2C application.
3. Under **Web App/Web API**, for **Include web app / web API**, select **Yes**.
4. For **Allow implicit flow**, select **Yes** (the default value).
5. Under **Reply URL**, enter your dedicated reply URLs. See [Reply URLs](#) below for information on reply URLs and how to format them here.
6. For **Include native client**, select **No** (the default value).
7. Select **Create**.

### Reply URLs

Reply URLs are important as they provide an allow list of the return domains when your site calls Azure AD B2C to authenticate a user. This permits the return of the authenticated user back to the domain from which they are signing into (your site domain).

In the **Reply URL** box on the **Azure AD B2c - Applications > New application** screen, you need to add separate lines for both your site domain and (once your environment is provisioned) the Commerce-generated URL. These URLs must always use a valid URL format and must be base URLs only (no trailing forward slashes or paths). The string `/_msdyn365/authresp` then needs to be appended to the base URLs, as in the following examples.

- `https://www.fabrikam.com/_msdyn365/authresp` (The domain should match the e-commerce domain completely. If you have multiple domains, you need to add this URL for each domain.)
- `https://fabrikam-prod.commerce.dynamics.com/_msdyn365/authresp`

## Create user flow policies

User flows are the policies Azure AD B2C uses to provide secure sign in, sign up, edit profile, and forget password user experiences. Dynamics 365 Commerce uses these flows to perform the policy actions to interact with the Azure AD B2C tenant. When a user interacts with these policies, they are redirected to the Azure AD B2C tenant to perform the actions.

Azure AD B2C provides three basic user flow types:

- Sign up and sign in

- Profile editing
- Password reset

You can choose to use the default user flows provided by Azure AD, which will display a page hosted by AAD B2C. Alternately, you can create an HTML page to control the look and feel of these user flow experiences.

To customize the user policy pages for Dynamics 365 Commerce, see [Set up custom pages for user logins](#). For additional information, see [Customize the interface of user experiences in Azure Active Directory B2C](#).

### Create a sign up and sign in user flow policy

To create a sign up and sign in user flow policy, follow these steps.

1. In the Azure portal, select **User flows (policies)** in the left navigation pane.
2. On the **Azure AD B2C – User flows (policies)** page, select **New User Flow**.
3. On the **Recommended** tab, select **Sign up and sign in**.
4. Under **Name**, enter a policy name. This name will display afterwards with a prefix the portal assigns (for example, "B2C\_1\_").
5. Under **Identity providers**, select the appropriate check box.
6. Under **Multifactor Authentication**, select the appropriate choice for your company.
7. Under **User attributes and claims**, select options to collect attributes or return claims as appropriate. Commerce requires the following default options:

COLLECT ATTRIBUTE	RETURN CLAIM
Email Address	Email Addresses
Given Name	Given Name
	Identity Provider
Surname	Surname
	User's Object ID

8. Select **Create**.

The following image is an example of the Azure AD B2C sign up and sign in user flow.

Home > Azure AD B2C - User flows (policies) > Create a user flow > Create

### Create

Sign up and sign in

← Select a different type of user flow

Get started with your user flow with a few basic selections. Don't worry about getting everything right here, you can modify your user flow after you've created it.

**1. Name \***

The unique string used to identify this user flow in requests to Azure AD B2C. This cannot be changed after a user flow has been created.

\* B2C\_1\_ signinsignupe2e

**2. Identity providers \***

Identity providers are the different types of accounts your users can use to log into your application. You need to select at least one for a valid user flow and you can select multiple.

Please select at least one identity provider

Email sign up

**3. Multifactor authentication**

Enabling multifactor authentication (MFA) requires your users to verify their identity with a second factor before allowing them into your application. [Learn more about MFA](#)

Multifactor authentication

**4. User attributes and claims**

User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your directory.

	Collect attribute	Return claim
City	<input type="checkbox"/>	<input type="checkbox"/>
Country/Region	<input type="checkbox"/>	<input type="checkbox"/>
Display Name	<input type="checkbox"/>	<input type="checkbox"/>
Email Address	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Email Addresses	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Given Name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Identity Provider	<input type="checkbox"/>	<input type="checkbox"/>
Job Title	<input type="checkbox"/>	<input type="checkbox"/>
Postal Code	<input type="checkbox"/>	<input type="checkbox"/>
State/Province	<input type="checkbox"/>	<input type="checkbox"/>
Street Address	<input type="checkbox"/>	<input type="checkbox"/>
Surname	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
User is new	<input type="checkbox"/>	<input type="checkbox"/>
User's Object ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

The following image shows the Run user flow option in the Azure AD B2C sign up and sign in user flow.

Home > Azure AD B2C - User flows (policies) > B2C\_1\_signinsignupe2e

**B2C\_1\_signinsignupe2e**  
Sign up and sign in

Search (Ctrl-/)

Overview

Settings

- Properties
- Identity providers
- User attributes
- Application claims

Customize

- Page layouts
- Languages

Run user flow | Delete | Download

Got a second? We would love your feedback on the user flows management experience →

**Settings**

- Properties: Multifactor authentication (Disable), Password complexity (Strong)
- Identity providers: Email sign up, Facebook
- User attributes: Email Address, Given Name, Surname
- Application claims: Email Addresses, Given Name, 3 more

**Customize**

- Page layouts: Classic (Default)
- Languages: English

**Run user flow**

https://rushmore2e.b2clogin.com/rushmore2e.onmicrosoft.com/v2.0/well-known/openid-configuration?p=B2C\_1\_signinsignupe2e

For your requests and redirections to the Azure AD B2C service, we recommend that you use B2CLogin.com. To transition your existing experiences make sure that the reply URL of your identity providers matches the domain used by your request (this includes your application and for "run now"). login.microsoftonline.com will continue working for the near future, but we are planning on deprecating this. For details [click here](#)

Application: rushmore2eb2c

Reply URL: https://www.rushmore.com

Select domain: rushmore2e.b2clogin.com

Access Tokens

Run user flow endpoint: https://rushmore2e.b2clogin.com/rushmore2e.onmicrosoft.com

## Create a profile editing user flow policy

To create a profile editing user flow policy, follow these steps.

1. In the Azure portal, select **User flows (policies)** in the left navigation pane.
2. On the **Azure AD B2C – User flows (policies)** page, select **New User Flow**.
3. On the **Recommended** tab, select **Profile editing**.
4. Under **Name**, enter the profile editing user flow. This name will display afterwards with a prefix the portal assigns (for example, "B2C\_1\_").
5. Under **Identity providers**, select **Local Account SignIn**.
6. Under **User attributes**, select the following check boxes:
  - **Email Addresses (Return claim only)**
  - **Given Name (Collect attribute and Return claim)**
  - **Identity Provider (Return claim only)**
  - **Surname (Collect attribute and Return claim)**
  - **User's Object ID (Return claim only)**

## 7. Select Create.

The following image shows an example of the Azure AD B2C profile editing user flow.

Home > Azure AD B2C - User flows (policies) > Create a user flow > Create

### Create

Profile editing

[← Select a different type of user flow](#)

Get started with your user flow with a few basic selections. Don't worry about getting everything right here, you can modify your user flow after you've created it.

#### 1. Name \*

The unique string used to identify this user flow in requests to Azure AD B2C. This cannot be changed after a user flow has been created.

\* B2C\_1\_ editProfilee2e

#### 2. Identity providers \*

Identity providers are the different types of accounts your users can use to log into your application. You need to select at least one for a valid user flow and you can select multiple.

Please select at least one identity provider

Local Account Signin

#### 3. User attributes

User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your application.

	Collect attribute	Return claim
City	<input type="checkbox"/>	<input type="checkbox"/>
Country/Region	<input type="checkbox"/>	<input type="checkbox"/>
Display Name	<input type="checkbox"/>	<input type="checkbox"/>
Email Addresses	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Given Name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Identity Provider	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Job Title	<input type="checkbox"/>	<input type="checkbox"/>
Postal Code	<input type="checkbox"/>	<input type="checkbox"/>
State/Province	<input type="checkbox"/>	<input type="checkbox"/>
Street Address	<input type="checkbox"/>	<input type="checkbox"/>
Surname	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
User's Object ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>

## Create a password reset user flow policy

To create a password reset user flow policy, follow these steps.

1. In the Azure portal, select **User flows (policies)** in the left navigation pane.
2. On the **Azure AD B2C – User flows (policies)** page, select **New User Flow**.
3. On the **Recommended** tab, select **Password Reset**.
4. Under **Name**, enter a name for the password reset user flow.
5. Under **Identity providers**, select **Reset password using email address**.
6. Select **Create**.
7. Under **Application claims**, select the following check boxes:
  - **Email addresses**
  - **Given Name**
  - **Surname**
  - **User's Object ID**
8. Select **Create**.

The following image shows where to set **Reset Password using mail address** in the Azure AD B2C password reset user flow.

Home > Azure AD B2C - User flows (policies) > Create a user flow > Create

### Create

Password reset

[← Select a different type of user flow](#)

Get started with your user flow with a few basic selections. Don't worry about getting everything right here, you can modify your user flow after you've created it.

#### 1. Name \*

The unique string used to identify this user flow in requests to Azure AD B2C. This cannot be changed after a user flow has been created.

\* B2C\_1\_ PasswordReset

#### 2. Identity providers \*

Identity providers are the different types of accounts your users can use to log into your application. You need to select at least one for a valid user flow and you can add more providers.

Please select at least one identity provider

Reset password using email address

#### 3. Multifactor authentication

Enabling multifactor authentication (MFA) requires your users to verify their identity with a second factor before allowing them into your application. [Learn more about MFA](#)

Multifactor authentication Enabled Disabled

#### 4. Application claims

User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your directory.

Return claim

Given Name

Surname

City

Country/Region

Display Name

Show more...

#### 4. Application claims

User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your directory.

Return claim

City

Country/Region

Display Name

Email Addresses

Given Name

Job Title

Postal Code

State/Province

Street Address

Surname

User's Object ID

## Add social identity providers (Optional)

Social identity providers allow users to use their social accounts for authentication. Adding social identity provider authentication is optional in Dynamics 365 Commerce.

If social identity provider authentication is not added, the default Azure AD B2C profiles will be the main profiles for your user base. Users will select their own username (their preferred email address) and set a password. Azure AD B2C will authenticate users directly.

If social identity provider authentication is added and a user chooses one of the social identity providers offered, an entity is still created in the Azure AD B2C tenant. Azure AD B2C will then authenticate the user's credentials with the social identity provider.

### NOTE

The identity provider sign in creates a record in the B2C tenant, but in a different format than local accounts since it will call the external social identity provider reference for authentication. The user can use the same email address across social identity providers, meaning that the email username used for authentication may not be unique to the tenant. Azure AD B2C will only enforce that users have a unique email address on local B2C accounts.

Before you can add a social identity provider for authentication, you must go to the identity provider's portal and set up an identity provider application as instructed in the Azure AD B2C documentation. A list of links to the documentation is provided below.

- [Amazon](#)
- [Azure AD \(Single Tenant\)](#)
- [Microsoft Account](#)
- [Facebook](#)
- [GitHub](#)
- [Google](#)
- [LinkedIn](#)
- [OpenID Connect](#)
- [Twitter](#)

## Add and set up a social identity provider

To add and set up a social identity provider, follow these steps.

1. In the Azure portal, navigate to **Identity Providers**.
2. Select **Add**. The **Add identity provider** screen appears.
3. Under **Name**, enter the name to be displayed to users on your sign in screen.
4. Under **Identity provider type**, select an identity provider from the list.
5. Select **OK**.
6. Select **Set up this identity provider** to access the **Set up the social identity provider** screen.
7. Under **Client ID**, enter the client ID as obtained from the identity provider application setup.
8. Under **Client secret**, enter the client secret as obtained from the identity provider application setup.
9. Attach user flow for sign in sign up policies:
10. Go to **Azure AD B2C – User flows (policies) > {your sign-in sign-up policy} > Identity providers**.
11. To attach the sign in/sign up user flow policy, select each identity provider you have set up for your account. To test these, select **Run user flow** for each identity provider. A new tab will display the sign-in page displaying the new identity provider selection box.

The following image shows examples of the **Add identity provider** and **Set up the social identity provider** screens in Azure AD B2C.

The image shows two side-by-side screenshots from the Azure AD B2C portal. The left screenshot is titled 'Add identity provider' and shows the following fields: 'Name' with the value 'Facebook', 'Identity provider type' with the value 'Facebook', and 'Set up this identity provider' with the value 'Required'. The right screenshot is titled 'Set up the social identity provider' and shows the following fields: 'Client ID' with the value '00' and 'Client secret' with the value '00'. Both screenshots have a breadcrumb trail: 'Home > Azure AD B2C - Identity providers > Add identity provider > Set up the social identity provider'.

The following image shows an example of how to select identity providers on the Azure AD B2C **Identity Providers** page.

The image shows the 'Identity providers' page in the Azure AD B2C portal. The breadcrumb trail is 'Home > Azure AD B2C - User flows (policies) > B2C\_1\_signinsigne2e - Identity providers'. The page title is 'B2C\_1\_signinsigne2e - Identity providers'. Below the title, there is a search bar and a navigation menu with options: 'Run user flow', 'Save', 'Discard', and 'Manage identity providers'. The main content area contains a table of identity providers. The table has two columns: 'IDENTITY PROVIDER' and 'NAME'. The table contains two rows: 'Facebook' and 'Local Account'. Both rows have a checkmark in the 'IDENTITY PROVIDER' column. The 'NAME' column contains 'Facebook' and 'Email signup' respectively.

IDENTITY PROVIDER	NAME
<input checked="" type="checkbox"/> Facebook	Facebook
<input checked="" type="checkbox"/> Local Account	Email signup

The following image shows an example of a default sign-in screen with a social identity provider sign-in button displayed.





Sign in with your social account



OR

Sign in with your existing account

Email Address

Password [Forgot your password?](#)

Sign in

Don't have an account? [Sign up now](#)

## Update Commerce headquarters with the new Azure AD B2C information

Once the Azure AD B2C provisioning steps above are completed, the Azure AD B2C application must be registered in your Dynamics 365 Commerce environment.

To update headquarters with the new Azure AD B2C information, follow these steps.

1. In Commerce, go to **Commerce Shared Parameters** and select **Identity Providers** in the left menu.
2. Under **Identity Providers**, do the following:
  - a. In the **Issuer** box, enter the identity provider issuer URL. To find your issuer URL, see [Obtain issuer URL](#) below.
  - b. In the **Name** box, enter a name for your issuer record.
  - c. In the **Type** box, enter **Azure AD B2C (id\_token)**.
3. Under **Relying Parties**, with the above B2C identity provider item selected, do the following:
  - a. In the **ClientID** box, enter your B2C application ID. You can find this in the **Application ID** box of your B2C application's properties page.
  - b. In the **Type** box, enter **Public**.
  - c. In the **User Type** box, enter **Customer**.
4. On the action pane, select **Save**.
5. In the Commerce search box, search for **Distribution schedule**
6. In the left navigation menu of the **Distribution schedules** page, select job **1110 Global configuration**.
7. On the action pane, select **Run Now**.

### Obtain issuer URL

To obtain your identity provider issuer URL, follow these steps.

1. Create a metadata address URL in the following format using your B2C tenant and policy:

```
https://<B2CTENANTNAME>.b2clogin.com/<B2CTENANTNAME>.onmicrosoft.com/v2.0/.well-known/openid-configuration?p=<B2CSIGN-INPOLICY>
```

- Example:

```
https://d365plc.b2clogin.com/d365plc.onmicrosoft.com/v2.0/.well-known/openid-configuration?p=B2C_1_signinup
```

2. Enter the metadata address URL into a browser address bar.
3. In the metadata, copy the identity provider issuer URL (the value for "issuer").

- Example: `https://login.fabrikam.com/073405c3-0113-4f43-b5e2-df01266e24ae/v2.0/`.

## Configure your B2C tenant in Commerce site builder

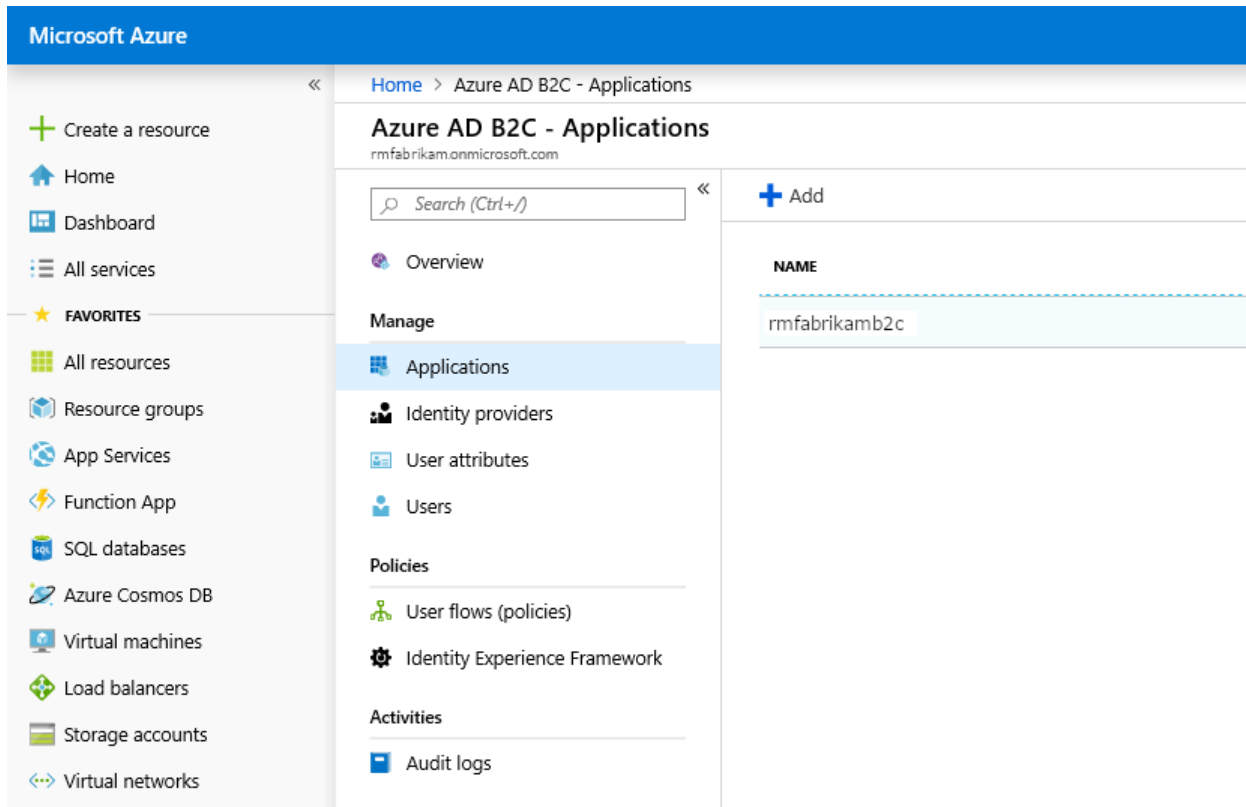
Once setup of your Azure AD B2C tenant is completed, you must configure the B2C tenant in Commerce site builder. Configuration steps include collecting B2C application information from the Azure portal, entering that B2C application information into site builder, and then associating the B2C application with your site and channel.

### Collect the required application information

To collect the required application information, follow these steps.

1. In the Azure portal, go to **Home > Azure AD B2C - Applications**.
2. Select your application, and then in the left navigation pane select **Properties** to obtain the application details.
3. From the **Application ID** box, collect the application ID of the B2C application created in your B2C tenant. This will later be entered as the **Client GUID** in site builder.
4. Under **Reply URL**, collect the reply URL.
5. Go to **Home > Azure AD B2C – User flows (policies)**, and then collect the names of each user flow policy.

The following image shows an example of the Azure AD B2C - Applications page.



The following image shows an example of an application **Properties** page in Azure AD B2C.

rmfabrikamb2c - Properties

Search (Ctrl+F) Save Discard Delete

**General**

Properties

Keys

API Access

API access

Published scopes

Name  
rmfabrikamb2c

Application ID  
76931820-60f8-4569-8060-0813d08d89f7

Web App / Web API

Include web app / web API  
 Yes  No

Allow implicit flow  
 Yes  No

Redirect URIs must all belong to the same domain

Reply URL  
https://www.rmfabrikam.com

App ID URI (optional)  
https://rmfabrikam.onmicrosoft.com/

Native client

Include native client  
 Yes  No

The following image shows an example of user flow policies on the **Azure AD B2C – User flows (policies)** page.

## + New user flow

User flow name

*Search using user flow name*

**NAME**

**B2C\_1\_editProfilee2e**

**B2C\_1\_PasswordReset**

**B2C\_1\_signinsignupe2e**

### Enter your AAD B2C tenant application information into Commerce

You must enter details of the Azure AD B2C tenant into Commerce site builder before associating the B2C tenant with your site(s).

To add your AAD B2C tenant application information to Commerce, follow these steps.

1. Sign in as an administrator to Commerce site builder for your environment.
2. In the left navigation pane, select **Tenant Settings** to expand it.
3. Under **Tenant Settings**, select **B2C Settings**.
4. In the main window next **B2C Applications**, select **Manage**. (If your tenant appears in the B2C Applications list, then it was already added by an administrator. Verify that the items in step 6 below match your B2C Application.)
5. Select **Add B2C Application**.
6. Enter the following required items in the form displayed, using values from your B2C tenant and application. Fields that are not required (those without an asterisk) may be left blank.
  - **Application Name:** The name for your B2C Application, for example "Fabrikam B2C".
  - **Tenant Name:** The name of your B2C tenant (for example, use "fabrikam" if the domain appears as

"fabrikam.onmicrosoft.com" for the B2C tenant).

- **Forget Password Policy ID:** The forget password user flow policy ID, for example "B2C\_1\_PasswordReset".
- **Signup Signin Policy ID:** The sign up and sign in user flow policy ID, for example "B2C\_1\_signup\_signin".
- **Client GUID:** The B2C application ID, for example "22290eb2-c52e-42e9-8b35-a2b0a3bcb9e6".
- **Edit Profile Policy ID:** The profile editing user flow policy ID, for example "B2C\_1A\_ProfileEdit".

7. Select **OK**. You should now see the name of your B2C application appear in the list.

8. Select **Save** to save your changes.

### Associate the B2C application to your site and channel

#### WARNING

If your site is already associated with a B2C application, changing to a different B2C application will remove the current references established for users already signed up in this environment. If changed, any credentials associated with the currently-assigned B2C application will not be available to users.

Only update the B2C application if you are setting up the channel's B2C application for the first time or if you intend to have users re-sign up with new credentials to this channel with the new B2C application. Take caution when associating channels to B2C applications, and name applications clearly. If a channel is not associated to a B2C application in the steps below, users signing into that channel for your site will be entered into the B2C application showing as **default** in the **Tenant Settings > B2C Settings** list of B2C applications.

To associate the B2C application to your site and channel, follow these steps.

1. Navigate to your site in Commerce site builder.
2. In the left navigation pane, select **Site Settings** to expand it.
3. Below **Site Settings**, select **Channels**.
4. In the main window under **Channels**, select your channel.
5. In the channel properties pane on the right, select your B2C application name from the **Select B2C Application** drop down menu.
6. Select **Close**, and then select **Save and Publish**.

## Additional B2C information

### Customer migration

If you are considering migrating customer records from a previous identity provider platform, please work with the Dynamics 365 Commerce team to review your customer migration needs.

For additional Azure AD B2C documentation on customer migration, see [Migrate users to Azure Active Directory B2C](#).

### Custom policies

For additional information regarding customizing Azure AD B2C interactions and policy flows beyond what is offered by B2C standard policies, see [Custom policies in Azure Active Directory B2C](#).

### Secondary admin

An optional, secondary administrator account can be added in the **Users** section of your B2C tenant. This can be a direct account or a general account. If you need to share an account across team resources, a common account can also be created. Due to the sensitivity of the data stored in Azure AD B2C, a common account should be monitored closely per your company's security practices.

# Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)[Associate a Dynamics 365 Commerce site with an online channel](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up custom pages for user sign-ins

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic describes how to build custom pages in Microsoft Dynamics 365 Commerce that handle customized sign-ins for users of Azure Active Directory (Azure AD) business-to-consumer (B2C) tenants.

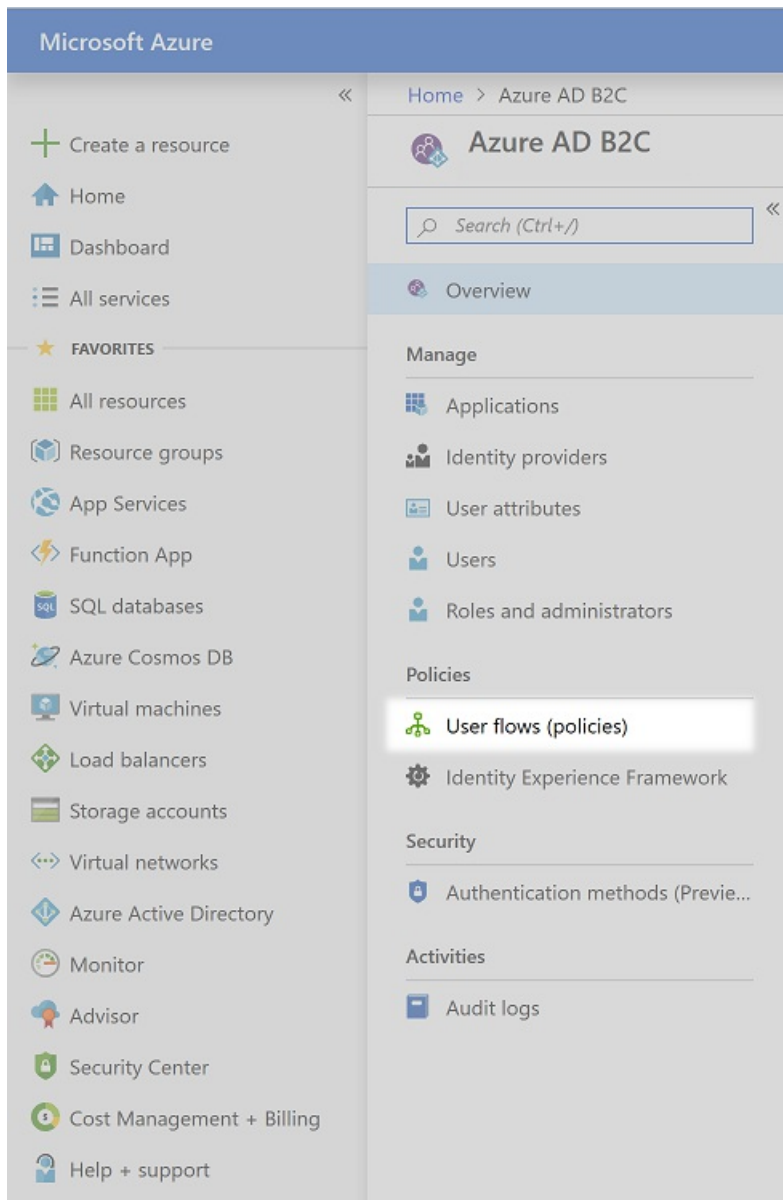
## Overview

To use custom pages that are authored in Dynamics 365 Commerce to handle user sign-in flows, you must set up the Azure AD policies that will be referenced in the Commerce environment. You can configure the "Sign up and sign in," "Profile editing," and "Password reset" Azure AD B2C policies by using the Azure AD B2C application. The Azure AD B2C tenant and policy names can then be referenced during the provisioning process that is done for the Commerce environment by using Microsoft Dynamics Lifecycle Services (LCS).

The custom Commerce pages can be built by using the sign in, sign up, account profile edit, or password reset module. The page URLs that are published for these custom pages should then be referenced in Azure AD B2C policy configurations in the Azure portal.

## Set up B2C policies

After you set up your Azure AD B2C tenant and associate it with your Commerce environment, go to the **Azure AD B2C** page in the Azure portal, and then, on the menu, under **Policies**, select **User flows (policies)**.



You can now configure the "Sign up and sign in," "Profile editing," and "Password reset" user sign-in flows.

### Configure the "Sign up and sign in" policy

To configure the "Sign up and sign in" policy, follow these steps.

1. Select **New user flow**, and then, on the **Recommended** tab, select the **Sign up and sign in** policy.
2. Enter a name for the policy (for example, **B2C\_1\_SignInSignUp**).
3. In the **Identity Providers** section, select the identity providers to use for the policy. At a minimum, **Email signup** must be selected.
4. In the **Collect attribute** column, select the check boxes for **Email Address**, **Given Name**, and **Surname**.
5. In the **Return claim** column, select the check boxes for **Email Addresses**, **Given Name**, **Identity Provider**, **Surname**, and **User's Object ID**.

## Create

### 4. User attributes and claims

User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your directory.

	Collect attribute	Return claim
City ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
Country/Region ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
Display Name ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
Email Address ⓘ	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Email Addresses ⓘ	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Given Name ⓘ	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Identity Provider ⓘ	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Identity Provider Access Token ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
Job Title ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
Postal Code ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
State/Province ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
Street Address ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
Surname ⓘ	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
User is new ⓘ	<input type="checkbox"/>	<input type="checkbox"/>
User's Object ID ⓘ	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ok

6. Select OK to create the policy.
7. Double-click the new policy name, and then, in the navigation pane, select **Properties**.
8. Set the **Enable JavaScript enforcing page layout (preview)** option to **On**.

**B2C\_1\_TestSignUpPolicy - Properties**  
Sign up and sign in

Search (Ctrl+) | Run user flow | Save | Discard

Overview

Settings

- Properties
- Identity providers
- User attributes
- Application claims

Customize

- Page layouts
- Languages

Multifactor authentication

Enabling multifactor authentication (MFA) requires your users to verify their identity with a second factor before allowing them into your application. [Learn more about multifactor authentication.](#)

Multifactor authentication  Enabled  Disabled

Enable JavaScript enforcing page layout (preview)  On  Off [Learn more about using JavaScript with Page Layout versions.](#)

Token lifetime

Access & ID token lifetimes (minutes) ⓘ

Refresh token lifetime (days) ⓘ

Refresh token sliding window lifetime ⓘ  Bounded  No expiry

Lifetime length (days)  ✓



## NOTE

The policy name will be fully referenced in the Commerce environment. (The **B2C\_1\_** prefix will be included in the reference.) Policies can't be renamed after they are created. If you're replacing an existing policy for your Commerce environment, you can delete the original policy and build a new policy that has the same name. Alternatively, if the environment has already been provisioned, you can submit the new policy name through a service request.

You will return to this policy to finish the setup after you've built the custom pages. For now, close the policy to return to the **User flows (policies)** page in the Azure portal.

### Configure the "Profile editing" policy

To configure the "Profile editing" policy, follow these steps.

1. Select **New user flow**, and then, on the **Recommended** tab, select the **Profile editing** policy.
2. Enter a name for the policy (for example, **B2C\_1\_EditProfile**).
3. In the **Identity Providers** section, select the identity providers to use for the policy. At a minimum, **Local Account SignIn** must be selected.
4. In the **Collect attribute** column, select the check boxes for **Email Addresses** and **Surname**.
5. In the **Return claim** column, select the check boxes for **Email Addresses**, **Given Name**, **Identity Provider**, **Surname**, and **User's Object ID**.
6. Select **OK** to create the policy.
7. Double-click the new policy name, and then, in the navigation pane, select **Properties**.
8. Set the **Enable JavaScript enforcing page layout (preview)** option to **On**.

You will return to this policy to finish the setup after you've built the custom pages. For now, close the policy to return to the **User flows (policies)** page in the Azure portal.

### Configure the "Password reset" policy

To configure the "Password reset" policy, follow these steps.

1. Select **New user flow**, and then, on the **Preview** tab, select the **Password reset v1.1** policy.

Create a user flow

Select a user flow type

A user flow is a series of pages for your users to interact with to sign up or sign into their account. Choose one to start with and you can create multiple user flows to define your entire authentication experience for your app. [Learn more about user flow types.](#)

Recommended Preview All

V2 user flows allow you to use new features that are not available for older versions. [For more information click here.](#)

<b>Sign up and sign in v2</b> Lets a user register for or log into their account	<b>Profile editing v2</b> Lets the user configure their user attributes	<b>Password reset v1.1</b> Allows a user to choose a new password after verifying their email (new page layouts available)
<b>Password reset v2</b> Allows a user to choose a new password after verifying their email	<b>Sign up v2</b> Lets the user register for an account	<b>Sign in v2</b> Lets the user log into their account
<b>Sign in using ROPC</b> Lets users with local accounts sign in directly in native apps (no browser required)		

2. Enter a name for the policy (for example, **B2C\_1\_ForgetPassword**).
3. In the **Identity Providers** section, select **Reset password using email address**.
4. In the **Return claim** column, select the check boxes for **Email Addresses**, **Given Name**, **Surname**, and **User's Object ID**.

## Create

---

### 4. Application claims

User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your directory.

	Return claim
City ⓘ	<input type="checkbox"/>
Country/Region ⓘ	<input type="checkbox"/>
Display Name ⓘ	<input type="checkbox"/>
Email Addresses ⓘ	<input checked="" type="checkbox"/>
Given Name ⓘ	<input checked="" type="checkbox"/>
Job Title ⓘ	<input type="checkbox"/>
Postal Code ⓘ	<input type="checkbox"/>
State/Province ⓘ	<input type="checkbox"/>
Street Address ⓘ	<input type="checkbox"/>
Surname ⓘ	<input checked="" type="checkbox"/>
User's Object ID ⓘ	<input checked="" type="checkbox"/>

Ok

5. Select **OK** to create the policy.
6. Double-click the new policy name, and then, in the navigation pane, select **Properties**.
7. Set the **Enable JavaScript enforcing page layout (preview)** option to **On**.

You will return to this policy to finish the setup after you've built the custom pages. For now, close the policy to return to the **User flows (policies)** page in the Azure portal.

## Build the custom pages

To build the custom pages to handle user sign-ins, follow these steps.

1. In the Commerce authoring tools, go to your site.
2. Build the following five templates and five pages:
  - A **Sign In** template and page that use the sign in module.
  - A **Sign Up** template and page that use the sign up module.
  - A **Password Reset** template and page that use the password reset module.
  - A **Password Reset verification** template and page that use the password reset verification module.

- A **Profile Edit** template and page that use the account profile edit module

When you build the pages, follow these guidelines:

- For each page or module, use the layout and style that best suit your business requirements.
- Publish all pages and URLs that must be used in the Azure AD B2C setup.
- After the pages and URLs are published, collect the URLs that must be used for the Azure AD B2C policy configurations. A `?preloadscripts=true` suffix will be added to every URL when it's used.

#### **IMPORTANT**

Don't reuse universal headers and footers that have relative links. Because these pages will be hosted in the Azure AD B2C domain when they are used, only absolute URLs should be used for all links.

## Configure Azure AD B2C policies with custom page information

In the Azure portal, return to the **Azure AD B2C** page, and then, on the menu, under **Policies**, select **User flows (policies)**.

### **Update the "Sign up and sign in" policy with custom page information**

To update the "Sign up and sign in" policy with custom page information, follow these steps.

1. In the **Sign in and sign up** policy that you configured earlier, in the navigation pane, select **Page layouts**.
2. Select the **Unified sign up or sign in page** layout.
3. Set the **Use custom page content** option to **Yes**.
4. In the **Custom page URI** field, enter the full sign-in URL. Include the `?preloadscripts=true` suffix. For example, enter `www.<my domain>.com/sign-in?preloadscripts=true`.
5. In the **Page Layout Version (Preview)** field, select **1.2.0**.
6. Select the **Local account sign up page** layout.
7. Set the **Use custom page content** option to **Yes**.
8. In the **Custom page URI** field, enter the full sign-up URL. Include the `?preloadscripts=true` suffix. For example, enter `www.<my domain>.com/sign-up?preloadscripts=true`.
9. In the **Page Layout Version (Preview)** field, select **1.2.0**.
10. In the **User attributes** section, follow these steps:
  - a. For the **Email Address**, **Given Name**, and **Surname** attributes, select **No** in the **Requires Verification** field.
  - b. For the **Given Name** and **Surname** attributes, select **No** in the **Optional** field.


Select a page to customize its appearance. You can provide your own html and css to add your own branding and layout. [Learn more about customizing your page.](#)

LAYOUT NAME	CUSTOM PAGE
Unified sign up or sign in page	No
Local account sign up page	No
Error page	No

#### Local account sign up page

Use custom page content




 Yes  No


\* Custom page URI 

Page Layout Version (Preview) 

[Learn more about Page Layout versions.](#)

User attributes

 Move up  Move down  Move to top  Move to bottom

NAME	OPTIONAL	REQUIRES VERIFICATION	USER INPUT TYPE	LABEL
Email Address	<input type="text" value="No"/>	<input type="text" value="Yes"/>	 EmailBox	Email Address
Given Name	No	No	TextBox	Given Name
Surname	No	No	TextBox	Surname

## Update the "Profile editing" policy with custom page information

To update the "Profile editing" policy with custom page information, follow these steps.

1. In the **Profile Editing** policy that you configured earlier, in the navigation pane, select **Page layouts**.
2. Select the **Profile edit page layout**.
3. Set the **Use custom page content** option to **Yes**.
4. In the **Custom page URI** field, enter the full profile edit URL. Include the **?preloadscripts=true** suffix. For example, enter  .
5. In the **Page Layout Version (Preview)** field, select **1.2.0**.
6. In the **User attributes** section, follow these steps:
  - a. For the **Email Address**, **Given Name** attributes, select **No** in the **Requires Verification** field.
  - b. For the **Given Name** and **Surname** attributes, select **No** in the **Optional** field.

## Update the "Password reset" policy with custom page information

To update the "Password reset" policy with custom page information, follow these steps.

1. In the **Password Reset** policy that you configured earlier, in the navigation pane, select **Page layouts**.
2. Select the **New password page layout**.
3. Set the **Use custom page content** option to **Yes**.
4. In the **Custom page URI** field, enter the full password reset URL. Include the **?preloadscripts=true** suffix. For example, enter  .
5. In the **Page Layout Version (Preview)** field, select **1.2.0**.
6. Select the **Account verification page layout**.
7. Set the **Use custom page content** option to **Yes**.
8. In the **Custom page URI** field, enter the full password reset verification URL. Include the **?preloadscripts=true** suffix. For example, enter  .

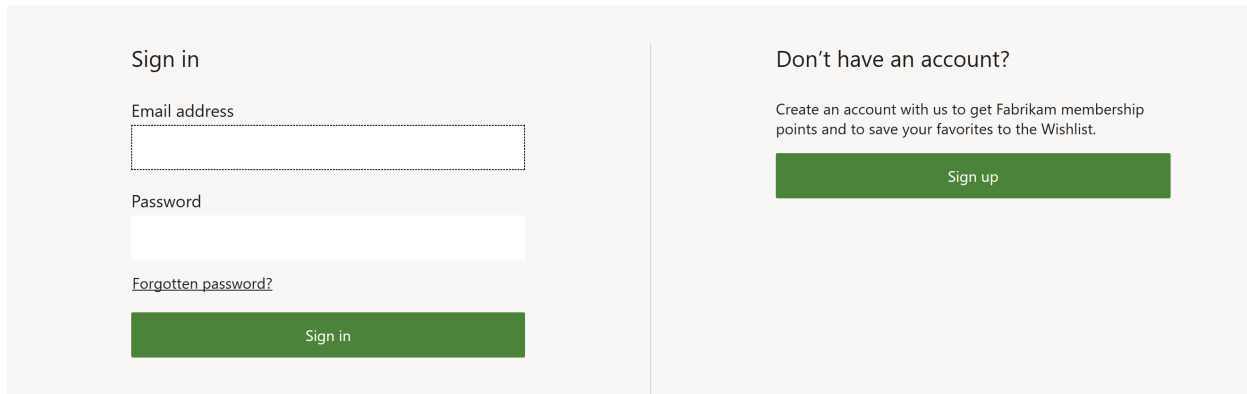
9. In the **Page Layout Version (Preview)** field, select **1.2.0**.

## Customize default text strings for labels and descriptions

In the module library, sign-in modules are prefilled with default text strings for the labels and descriptions. You can customize these strings in the software development kit (SDK) by updating the values in the `global.json` file for the sign in module.

For example, the default text for the forgotten password link is **Forgotten password?**. The following shows this default text on the sign-in page.

### Fabrikam



However, in the `global.json` file for the module library sign-in module, you can edit the text to **Forgot Password?**, as shown in the following illustration.

```
src > resources > modules > {} global.json > {} @msdyn365-commerce-modules.forgotPasswordButtonAriaLabel
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
"@msdyn365-commerce-modules.passwordDescription": {
  "value": "Must have at least 8 characters and contain at least three of the following: uppercase letters, lowercase letters, numbers, and symbols.",
  "_value.comment": "Must have more than 8 characters with at least 1 number."
},
"@msdyn365-commerce-modules.signUpButtonAriaLabel": {
  "value": "Create an account",
  "_value.comment": "Sign-up button aria label"
},
"@msdyn365-commerce-modules.signUpButtonText": {
  "value": "Create an account",
  "_value.comment": "Sign-up button text"
},
"@msdyn365-commerce-modules.forgotPasswordButtonAriaLabel": {
  "value": "Forgot password?",
  "_value.comment": "Forgot password button aria label."
},
```

After you update the `global.json` file and publish your changes, the new link text appears in the sign-in module in both Commerce and on the live sign-in page.

## Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Configure multiple B2C tenants in a Commerce environment](#)

Add support for a content delivery network (CDN)

Enable location-based store detection

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure multiple B2C tenants in a Commerce environment

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes when and how to set up multiple Microsoft Azure Active Directory (Azure AD) business-to-consumer (B2C) tenants per channel for user authentication in a dedicated Dynamics 365 Commerce environment.

## Overview

Dynamics 365 Commerce uses the Azure AD B2C cloud identity service to support user credentials and authentication flows. Users can use the authentication flows to sign up, sign in, and reset their password. Azure AD B2C stores a user's sensitive authentication information, such as his or her user name and password. The user record is unique to each B2C tenant, and it uses either user name (email address) credentials or social identity provider credentials.

In most cases, a single Azure AD B2C tenant is used in a Commerce environment. Commerce customers can then create and publish multiple sites in the same Commerce environment, and the same customer credentials will be used across these sites. However, if the sites in the environment should be treated as different brands and appear to users as separate businesses, a B2C tenant can be configured for the channel that is used for the site/brand separation.

## Considerations when multiple B2C tenants are set up per channel

Often, when each channel or site is being treated as a separate business, the best option with respect to user authentication flows in Commerce is to use separate legal entities. However, if you want to keep each channel/site in the same environment and legal entity, but want to have separate user authentication for each site, it's important that you consider the following points before you proceed:

- Users will have their own distinct credentials for each channel/site.

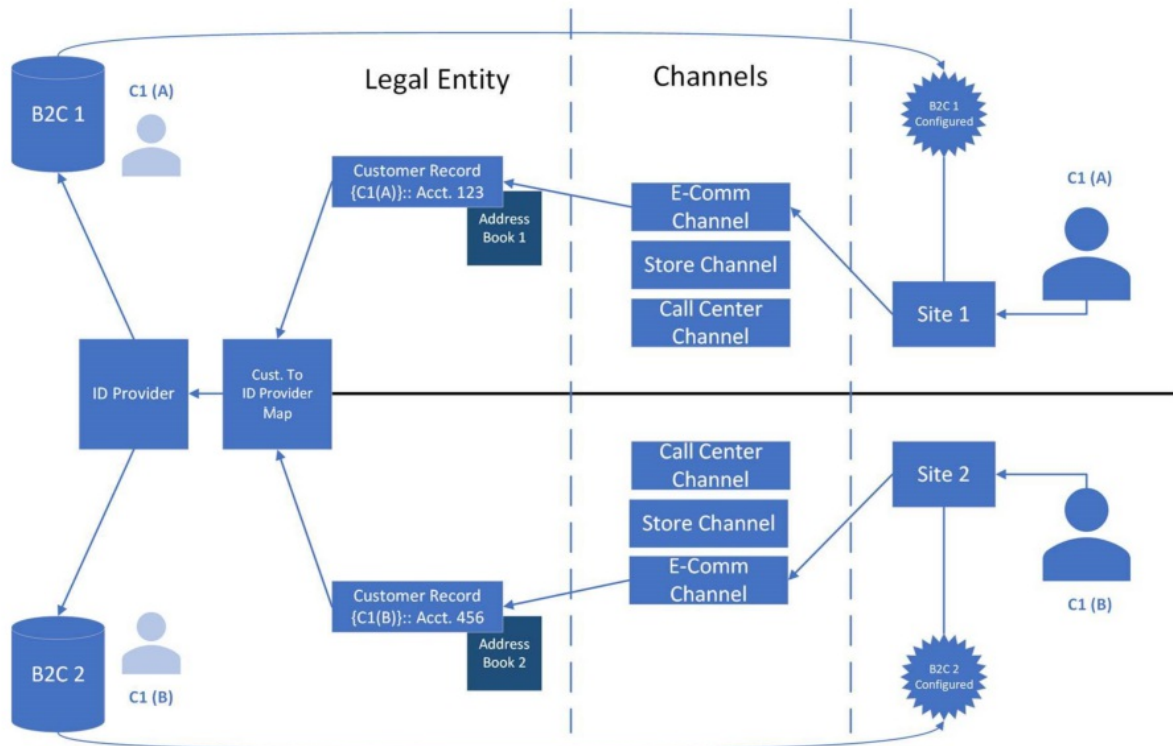
The same person can have two separate accounts per channel/site, because each account will be a unique entry into a separate B2C tenant.

- In the Microsoft Dynamics environment, separate customer records will be returned for global record searches.

If a user uses the same email address across channels/sites, global customer searches will return results for each channel/site. (A channel indicator will be shown.)

- The address book can be used to help group users, so that they can be tracked per channel.
- The number of customer records per channel might increase, and this increase might affect the performance of global customer searches.
- B2C tenants must be carefully mapped to a channel, to help prevent situations where customers sign up for an incorrect tenant. Otherwise, confusion or tracking issues can occur.

The following illustration shows multiple B2C tenants in a Commerce environment.



If you decide that your business requires distinct B2C tenants per channel in the same Commerce environment, complete the procedures in the following sections to request this feature.

## Request that B2C per channel be enabled in your environment

Currently, if you want distinct B2C tenants per channel to be available in the same Commerce environment, you must submit a request to Dynamics 365 Commerce. For more information, see [Get support for Lifecycle Services \(LCS\)](#), or discuss this issue with your Commerce solutions contact.

## Configure B2C tenants in your environment

To configure B2C tenants in your environment, complete the relevant procedures in this section.

### Add an Azure AD B2C tenant

To add an Azure AD B2C tenant to your environment, follow these steps.

1. Sign in to Commerce site builder for your environment as a system admin. To configure Azure AD B2C tenants, you must be a system admin for the Commerce environment.
2. In the left navigation pane, select **Tenant Settings** to expand it.
3. Select **B2C Settings**, and then select **Manage**.
4. Select **Add B2C Application**, and then enter the following information:
  - **Application Name:** Enter the name that should be used for the application in the context of managing it in Commerce. We recommend that you use the application name that you chose when you set up the Azure AD B2C application in the Azure portal. In this way, you can help reduce confusion when you manage B2C tenants in Commerce.
  - **Tenant Name:** Enter the B2C tenant name as it appears in the Azure portal.
  - **Forget Password Policy ID:** Enter the policy ID (the name of the policy in the Azure portal).
  - **Signup Signin Policy ID:** Enter the policy ID (the name of the policy in the Azure portal).
  - **Client GUID:** Enter the Azure AD B2C tenant ID as it appears in the Azure portal (not the application



ID for the B2C tenant).

- **Edit Profile Policy ID:** Enter the policy ID (the name of the policy in the Azure portal).

5. When you've finished entering this information, select **OK** to save your changes.

#### NOTE

You should leave fields such as **Scope**, **Non Interactive Policy ID**, **Non Interactive Client ID**, **Login Custom Domain**, and **Sign Up Policy ID** blank unless the Dynamics 365 Commerce team instructs you to set them. Your new Azure AD B2C tenant should now appear in the list under **Manage B2C Applications**.

### Manage or delete an Azure AD B2C tenant

1. Sign in to Commerce site builder for your environment as a system admin. To configure Azure AD B2C tenants, you must be a system admin for the Commerce environment.
2. In the left navigation pane, select **Tenant Settings** to expand it.
3. Select **B2C Settings**, and then select **Manage**.
4. To edit a B2C tenant, select the pencil symbol next to it. To delete a B2C tenant, select the trash can symbol next to it.
5. Select **Save**, and then select **Publish** to activate your changes.

#### WARNING

When a B2C tenant is configured for a live/published site, users might have signed up by using accounts that are present on the tenant. If you delete a configured tenant on the **Tenant Settings > B2C Tenant** menu, you remove the association of that B2C tenant from sites that are associated with any channels of the tenant. In this case, your users might no longer be able to sign in to their accounts. Therefore, use extreme caution when you delete a configured tenant.

When a configured tenant is deleted, the B2C tenant and records will continue to be maintained, but the Commerce system configuration of that tenant will be changed or removed. Users who try to sign up or sign in to the site will create a new account record in the default or newly associated B2C tenant that is configured for the channel of the site.

## Configure your channel with a B2C tenant

1. Sign in to Commerce site builder for your environment as a system admin. To configure Azure AD B2C tenants, you must be a system admin for the Commerce environment.
2. In the left navigation pane, select **Site Settings** to expand it.
3. Select **Channels**, and then select the channel to configure.
4. In the properties pane on the right, in the **Select B2C Application** field, select the configured Azure AD B2C tenant to use for this channel.
5. On the command bar, select **Save and Publish** to commit the new or updated configuration.

#### WARNING

If you change the B2C application that is assigned to the channel, you remove the current references that have been established for any users who have already signed up in the environment. In this case, any credentials that are associated with the currently assigned B2C application won't be available to users. Therefore, change a channel Azure AD B2C configuration only if you're setting up the channel for the first time, and no users have been able to sign up. Otherwise, users might have to sign up again to establish a record in the new Azure AD B2C tenant.

## Additional resources

[Configure your domain name](#)

Deploy a new e-commerce tenant

Create an e-commerce site

Associate a Dynamics 365 Commerce site with an online channel

Manage robots.txt files

Upload URL redirects in bulk

Set up a B2C tenant in Commerce

Set up custom pages for user logins

Add support for a content delivery network (CDN)

Enable location-based store detection

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Domains in Dynamics 365 Commerce

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic describes how domains are handled in Microsoft Dynamics 365 Commerce.

Domains are web addresses used to navigate to Dynamics 365 Commerce sites in a web browser. You control management of your domain with a chosen Domain Name Server (DNS) provider. Domains are referenced throughout Dynamics 365 Commerce site builder to coordinate how a site will be accessed when published. This topic reviews how domains are handled and referenced throughout the lifecycle of the Commerce site development and launch.

## Provisioning and supported host names

When provisioning an e-commerce environment in [Microsoft Dynamics Lifecycle Services \(LCS\)](#), the **Supported host names** box on the e-commerce provisioning screen is used to enter domains that will be associated with the deployed Commerce environment. These domains will be the customer-facing Domain Name Server (DNS) names where e-commerce websites will be hosted. Entering a domain at this stage does not start diverting traffic for the domain to Dynamics 365 Commerce. Traffic for a domain will only be routed to the Commerce endpoint when the DNS CNAME record is updated to use the Commerce endpoint with the domain.

### NOTE

Multiple domains can be entered into the **Supported host names** box by separating them with semi-colons.

The following illustration shows the LCS e-commerce provisioning screen with the **Supported host names** box highlighted.

## Other Settings

Azure active directory  
Microsoft

Supported host names (separated by ';')

Ex - www.contoso.com;www.contosousa.com \*

AAD security group for system admin

(Search for the group using first few letters of its name then press enter/icon to search.)

Filter

Enable ratings and review service

Yes

AAD security group for ratings and review moderator

(Search for the group using first few letters of its name then press enter/icon to search.)

Filter

Initialize

Cancel

You can create a service request to add additional domains to an environment if provisioning has already occurred. To create a service request in LCS, within your environment go to **Support > Support issues** and select **Submit an incident**.

## Commerce-generated URLs

When provisioning a Dynamics 365 Commerce e-commerce environment, Commerce will generate a URL that will be the working address for the environment. This URL is referenced in the e-commerce site link shown in LCS after the environment is provisioned. A Commerce-generated URL is in the format

`https://<e-commerce tenant name>.commerce.dynamics.com`, where the e-commerce tenant name is the name entered in LCS for the Commerce environment.

You can use production site host names in a sandbox environment as well. This option is ideal when you will be copying a site from a sandbox environment to production.

# Site setup

After your e-commerce environment is provisioned, you must set up your site in Commerce site builder to associate your site to the working URL.

When you first set up a site in site builder, the **Setup your Site** dialog box will appear.

The following illustration shows the **Setup your Site** dialog box for a site named "default" when you access the site for the first time in site builder.

**Setup your Site - default** ✕

Select a domain

Select a default channel

Select a default language

**Path**

/

OK Cancel

The **Select a domain** box allows you to associate one of the supported host names provided for your site in LCS to your site in site builder.

The **Path** box can be left blank, or an additional path string can be added that will be reflected in your working URL. Leaving the **Path** box blank associates the base Commerce-generated URL with the site being set up in site builder. Paths must be unique for each site/domain pair. Within the site and domain selected, only one site in the environment can use the blank path or be associated with a unique path string. Any string added to the **Path** field during site setup will become a subpath of the base Commerce-generated URL used to access the site in a web browser.

**NOTE**

The path is also known as the **Match path** when adding a channel in the **Site Settings > Channels** configuration section of site builder.

For example, if you have a site in site builder called "fabrikam" in an e-commerce tenant named "xyz," and if you set up the site with a blank path, then you would access the published site content in a web browser by going

directly to the base Commerce-generated URL:

```
https://xyz.commerce.dynamics.com
```

Alternately, if you had added a path of "fabrikam" during this same site's setup, you would access the published site content in a web browser using the following URL:

```
https://xyz.commerce.dynamics.com/fabrikam
```

## Pages and URLs

After your site is set up with a path, all URLs associated with pages in site builder will build on the working URL (the Commerce-generated URL, or the Commerce-generated URL plus the path) for the site. Creating a new URL in site builder (**URLS /> +New**) by selecting a page from the list in the **New URL** dialog box and entering the URL path for that page will associate that URL with the selected page. The URL path value then appends to the site's working URL to access the page, and is labeled as `./<URL path>` in the URL list of the **URLs** page in site builder.

The following illustration shows the **New URL** dialog box in site builder with an example URL path highlighted.

### New URL

Select a page

<input checked="" type="checkbox"/> Name ↓	Description
<input checked="" type="checkbox"/> example	

URL path

The following illustration shows the **URLs** page in site builder with an example URL highlighted in the list.

# URLs

 35 results

Name	Type
<code>./example</code>	URL
<code>./password-reset</code>	URL
<code>./orderconfirmation</code>	URL

## Domains in site builder

The supported host names values are available to be associated as a domain when setting up a site. When selecting a supported host name value as the domain, you will see the chosen domain referenced throughout site builder. This domain is only a reference within the Commerce environment, live traffic for that domain will not yet be forwarded to Dynamics 365 Commerce.

When working with sites in site builder, if you have two sites set up with two different domains, you can append the `?domain=` attribute to your working URL to access the published site content in a browser.

For example, environment "xyz" has been provisioned, and two sites have been created and associated in site builder: one with the domain `www.fabrikam.com` and the other with the domain `www.constoso.com`. Each site was set up using a blank path. These two sites could then be accessed in a web browser as follows using the `?domain=` attribute:

- `https://xyz.commerce.dynamics.com?domain=www.fabrikam.com`
- `https://xyz.commerce.dynamics.com?domain=www.constoso.com`

When a domain query string is not given in an environment with multiple domains provided, Commerce uses the first domain you provided. For example, if the path "fabrikam" was provided first during site setup, the URL `https://xyz.commerce.dynamics.com` could be used to access the published site content site for `www.fabrikam.com`.

## Traffic forwarding in production

You can simulate multiple domains using domain query string parameters on the `commerce.dynamics.com` endpoint itself. But when you need to go live in production, you must forward the traffic for your custom domain to the `<e-commerce tenant name>.commerce.dynamics.com` endpoint.

The `<e-commerce tenant name>.commerce.dynamics.com` endpoint does not support custom domain Secure Sockets Layers (SSLs), so you must set up custom domains using a front door service or content delivery network (CDN).

To set up custom domains using a front door service or CDN, you have two options:

- Set up a front door service like Azure Front Door to handle front-end traffic and connect to your Commerce environment. This provides greater control over domain and certificate management and more granular security policies.
- Use the Commerce-supplied Azure Front Door instance. This requires coordinating action with the Dynamics

365 Commerce team for domain verification and obtaining SSL certificates for your production domain.

For information about how to set up a CDN service directly, see [Add support for a content delivery network \(CDN\)](#).

To use the Commerce-supplied Azure Front Door instance, you must create a service request for CDN setup assistance from the Commerce onboarding team.

- You will need to provide your company name, the production domain, environment ID, and production e-commerce tenant name.
- You will need to confirm if this is an existing domain (used for a currently active site) or a new domain.
- For a new domain, the domain verification and SSL certificate can be achieved in a single step.
- For a domain serving an existing website, there is a multistep process required to establish the domain verification and SSL certificate. This process has a 7-working-day service level agreement (SLA) for a domain to go live, because it includes multiple sequential steps.

To create a service request in LCS, within your environment go to **Support > Support issues** and select **Submit an incident**.

#### NOTE

Custom domains with SSL are only supported on production environments. For non-production environments such as sandbox and user acceptance testing (UAT), use the Commerce-generated URL to access published content in a web browser.

## SSL certificate process

When a service request is filed, the Commerce team will coordinate the following steps with you.

For new domains:

- The Commerce team will set up the Azure Front Door instance (Commerce-hosted).
- The Commerce team will then provide the CNAME record to point your custom domain.
- After the CNAME record is updated, the Commerce-hosted Azure Front Door instance will be able to verify the domain ownership and get the SSL certificate.

For existing/active domains:

- The Commerce team will instruct you to add an `afdverify.<custom-domain>` CNAME record to provide to your domain DNS provider.
- When complete, the Commerce team will add the domain to the Azure Front Door instance and provide additional DNS TXT records to be added to the DNS for the domain.
- After the TXT records are completed, the Commerce team will complete the Azure Front Door updates for the domain that will set up the SSL certificate.

## Apex domains

The Commerce-supplied Azure Front Door instance does not support apex domains (root domains that do not contain subdomains). Apex domains require an IP address to resolve, and the Commerce Azure Front Door instance exists with virtual endpoints only. To use an apex domain, you have two options:

- **Option 1** - Use your DNS provider to redirect the apex domain to a "www" domain. For example, `fabrikam.com` redirects to `www.fabrikam.com` where `www.fabrikam.com` is the CNAME record that points to the Commerce-hosted Azure Front Door instance.



- **Option 2** - Set up a CDN/front door instance on your own to host the apex domain.

**NOTE**

If you are using Azure Front Door, you must also set up an Azure DNS in the same subscription. The apex domain hosted on Azure DNS can point to your Azure Front Door as an alias record. This is the only work around, as apex domains must always point to an IP address.

## Additional resources

[Deploy a new e-commerce tenant](#)

[Set up an online store channel](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

[Enable location-based store detection](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add support for a content delivery network (CDN)

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to add a content delivery network (CDN) to your Microsoft Dynamics 365 Commerce environment.

## Overview

When you set up an e-commerce environment in Dynamics 365 Commerce, you can configure it to work with your CDN service.

Your custom domain can be enabled during the provisioning process for your e-commerce environment. Alternatively, you can use a service request to set it up after the provisioning process is completed. The provisioning process for the e-commerce environment generates a host name that is associated with the environment. This host name has the following format, where *<e-commerce-tenant-name>* is the name of your environment:

*<e-commerce-tenant-name>.commerce.dynamics.com*

The host name or endpoint that is generated during the provisioning process supports a Secure Sockets Layer (SSL) certificate only for \*.commerce.dynamics.com. It doesn't support SSL for custom domains. Therefore, you must terminate SSL for custom domains in your CDN and forward traffic from the CDN to the host name or endpoint that Commerce generated.

Additionally, the *statics* (JavaScript or Cascading Style Sheets [CSS] files) from Commerce are served from the endpoint that Commerce generated (\*.commerce.dynamics.com). The statics can be cached only if the host name or endpoint that Commerce generated is put behind the CDN.

## Set up SSL

To help guarantee that SSL is set up, and that statics are cached, you must configure your CDN so that it is associated with the host name that Commerce generated for your environment. You must also cache the following pattern for statics only:

```
/_msdyn365/_scnr/*
```

After you provision your Commerce environment with the custom domain that is provided, or after you provide the custom domain for your environment by using a service request, point your custom domain to the host name or endpoint that Commerce generated.

As was previously mentioned, the generated host name or endpoint supports an SSL certificate only for \*.commerce.dynamics.com. It doesn't support SSL for custom domains.

## CDN services

Any CDN service can be used with a Commerce environment. Here are two examples:

- **Microsoft Azure Front Door Service** – The Azure CDN solution. For more information about Azure Front Door Service, see [Azure Front Door Service Documentation](#).
- **Akamai Dynamic Site Accelerator** – For more information, see [Dynamic Site Accelerator](#).

## CDN setup

The CDN setup process consists of these general steps:

1. Add a front-end host.
2. Configure a backend pool.
3. Set up rules for routing and caching.

### **Add a front-end host**

Any CDN service can be used, but for the example in this topic, Azure Front Door Service is used.

For information about how to set up Azure Front Door Service, see [Quickstart: Create a Front Door for a highly available global web application](#).

### **Configure a backend pool in Azure Front Door Service**

To configure a backend pool in Azure Front Door Service, follow these steps.

1. Add `<ecom-tenant-name>.commerce.dynamics.com` to a backend pool as a custom host that has an empty backend host header.
2. Under **Load balancing**, leave the default values.

The following illustration shows the **Add a backend** dialog box in Azure Front Door Service with the backend host name entered.

# Add a backend



[← Go back to backend pool](#)

Backends are your application servers where Front Door will route your client requests to. You can assign weights to your backends to define proportion of traffic to be sent and set priority for the backends to define active/stand-by kind of architectures. [Learn more](#)

Backend host type \*

Backend host name \* ⓘ

Backend host header ⓘ

HTTP port \* ⓘ

HTTPS port \* ⓘ

Priority \* ⓘ

Weight \* ⓘ

Status

 Disabled  Enabled

The following illustration shows the **Add a backend pool** dialog box in Azure Front Door Service with the default load balancing values.

# Add a backend pool



A backend pool is a set of equivalent backends to which Front Door load balances your client requests. [Learn more](#)

Name \*



## BACKENDS

Backend host name	Status	Priority	Weight
ecomtenant.commerce.dynamics.com	Enabled	1	50

+ [Add a backend](#)

## HEALTH PROBES

Front Door sends periodic HTTP/HTTPS probe requests to each of your configured backends to determine the proximity and health of each backend to load balance your end user requests. [Learn more](#)

Status

Disabled  Enabled

## LOAD BALANCING

Configure the load balancing settings to define what sample set we need to use to call the backend as healthy or unhealthy. The latency sensitivity with value zero (0) means always send it to the fastest available backend, else Front Door will round robin traffic between the fastest and the next fastest backends within the configured latency sensitivity. [Learn more](#)

Sample size \* ⓘ

Successful samples required \* ⓘ

Latency sensitivity (in milliseconds) \* ⓘ

## Set up rules in Azure Front Door Service

To set up a routing rule in Azure Front Door Service, follow these steps.

1. Add a routing rule.
2. In the **Name** field, enter **default**.
3. In the **Accepted protocol** field, select **HTTP and HTTPS**.

4. In the **Frontend hosts** field, enter **dynamics-ecom-tenant-name.azurefd.net**.
5. Under **Patterns to match**, in the upper field, enter **/\***.
6. Under **Route Details**, set the **Route type** option to **Forward**.
7. In the **Backend pool** field, select **ecom-backend**.
8. In the **Forwarding protocol** field group, select the **Match request** option.
9. Set the **URL rewrite** option to **Disabled**.
10. Set the **Caching** option to **Disabled**.

To set up a caching rule in Azure Front Door Service, follow these steps.

1. Add a caching rule.
2. In the **Name** field, enter **statics**.
3. In the **Accepted protocol** field, select **HTTP and HTTPS**.
4. In the **Frontend hosts** field, enter **dynamics-ecom-tenant-name.azurefd.net**.
5. Under **Patterns to match**, in the upper field, **/\_msdyn365/\_scnr/\***.
6. Under **Route Details**, set the **Route type** option to **Forward**.
7. In the **Backend pool** field, select **ecom-backend**.
8. In the **Forwarding protocol** field group, select the **Match request** option.
9. Set the **URL rewrite** option to **Disabled**.
10. Set the **Caching** option to **Disabled**.
11. In the **Query string caching behavior** field, select **Cache every unique URL**.
12. In the **Dynamic compression** field group, select the **Enabled** option.

The following illustration shows the **Add a rule** dialog box in Azure Front Door Service.

## Add a rule

A routing rule maps your frontend host and a matching URL path pattern to a specific backend pool. [Learn more](#)

\* Name  ✓

Accepted protocol  ▼

Frontend hosts  ▼

### PATTERNS TO MATCH

Set this to all the URL path patterns that this route will accept. For example, you can set this to `/users/*` to accept all requests on the URL `www.contoso.com/users/*`. [Learn more](#)

✓

### ROUTE DETAILS

Once a route for a Front Door is matched, the configuration below defines the behavior of the route - forward and serve from the cache, or redirect. [Learn more](#)

Route type  Forward  Redirect

\* Backend pool  ▼

Forwarding protocol  HTTPS only  
 HTTP only  
 Match request

URL rewrite

Caching

\* Query string caching behavior  ▼

\* Dynamic compression  Enabled  Disabled

### WARNING

If the domain that you will use is already active and live, create a support ticket from the **Support** tile in [Microsoft Dynamics Lifecycle Services](#) to get assistance for your next steps. For more information, see [Get support for Finance and Operations apps or Lifecycle Services \(LCS\)](#).

If your domain is new and is not a pre-existing live domain, you can add your custom domain to the configuration for Azure Front Door Service. This will enable web traffic to direct to your site via the Azure Front Door instance. To add the custom domain (for example, `www.fabrikam.com`), you must configure a Canonical Name (CNAME) for the domain.

The following illustration shows the **CNAME configuration** dialog box in Azure Front Door Service.

## CNAME configuration

A CNAME record is used to specify that a domain name is an alias for another domain. In your scenario, that would be mapping 'www.fabrikam.com' to 'dynamics-ecom-tenant-name.azurefd.net'. Create a CNAME record with your DNS provider using the configuration below.

Source	www.fabrikam.com
Type	CNAME
Destination	dynamics-ecom-tenant-name.azurefd.net

You can use Azure Front Door Service to manage the certificate, or you can use your own certificate for the custom domain.

The following illustration shows the **Custom Domain HTTPS** dialog box in Azure Front Door Service.

### CUSTOM DOMAIN HTTPS

Enable HTTPS protocol for a custom domain that's associated with Front Door to ensure sensitive data is delivered securely via TLS/SSL encryption when sent across internet. [Learn more](#)

Status

Enabled  Disabled

Certificate management type

Front Door managed  Use my own certificate

For detailed instructions on adding a custom domain to your Azure Front Door, see [Add a custom domain to your Front Door](#).

Your CDN should now be correctly configured so that it can be used with your Commerce site.

## Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Enable location-based store detection](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable location-based store detection

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to turn on location-based store detection for your Dynamics 365 Commerce site.

## Overview

Location-based store detection in Commerce lets you provide relevant site content to customers, based on their location. When location-based store detection is turned on, the Commerce rendering service uses the country/region information from the IP address of the customer's web browser to direct the customer to the best geographical site configuration that is available.

## Privacy notice

If you turn on the location-based store detection feature, information from the customer's browser is sent to a Microsoft location service. This information is then used to provide the customer content that is relevant to his or her location. Both the information that is sent from the customer's browser and the location-based information that is returned to the customer are subject to privacy and cookie compliance policies.

## Turn on location-based store detection

To turn on location-based store detection in Commerce, follow these steps.

1. In the authoring tool, go to your site.
2. In the navigation pane on the left, select **Site Management**.
3. Select **Site Settings**.
4. Set the **Enable location based store detection** option to **On**.

## Additional resources

[Configure your domain name](#)

[Deploy a new e-commerce tenant](#)

[Create an e-commerce site](#)

[Associate a Dynamics 365 Commerce site with an online channel](#)

[Manage robots.txt files](#)

[Upload URL redirects in bulk](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user logins](#)

[Configure multiple B2C tenants in a Commerce environment](#)

[Add support for a content delivery network \(CDN\)](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# E-commerce digital gift cards

2/18/2021 • 4 minutes to read • [Edit Online](#)

## IMPORTANT

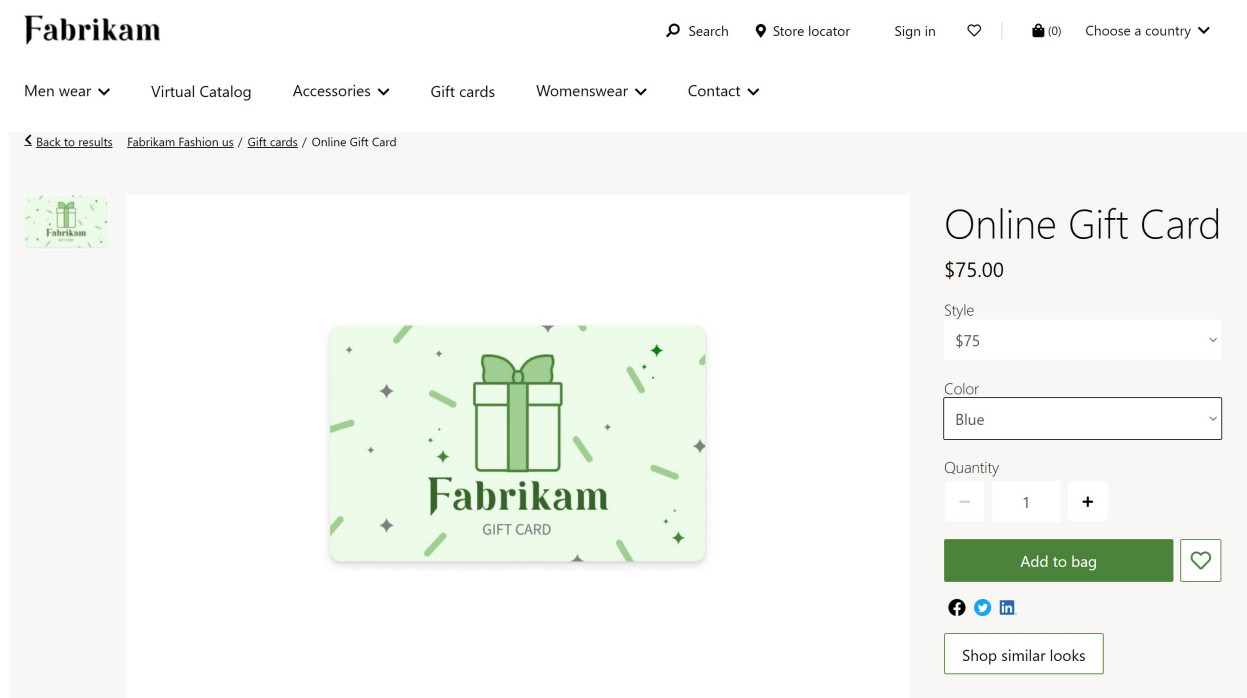
Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic describes how digital gift cards work in the e-commerce implementation of Microsoft Dynamics 365 Commerce. It also provides an overview of important configuration steps.

In Dynamics 365 Commerce, the purchase of digital gift cards follows the same flow as the purchase of other products in the system. No additional modules have to be configured. If multiple gift cards are added to the cart, the gift card items aren't aggregated on a single sales line. This behavior is required because each sales line is invoiced by using a separate gift card number.

The purchase of digital gift cards is supported in the Dynamics 365 Commerce 10.0.16 release and later.

The following illustration shows an example of the product details page (PDP) for a digital gift card on the Fabrikam e-commerce site.



## Turn on the digital gift card feature in Commerce headquarters

For the purchase flow for digital gift cards to work in Dynamics 365 Commerce, the **Purchasing gift card on e-Commerce feature** must be turned on in Commerce headquarters. You can find the feature in the **Feature management** workspace in Commerce headquarters, as shown in the following illustration.

**Feature management**  
Do not enable new features automatically

New: 1 | Not enabled: 444 | Scheduled: 0

Buttons: Enable all | Check for updates

Feature name	Enable date	Feature added	Preview feature	Module
Purchasing gift card on e-Commerce feature	10/8/2020	9/15/2020		Retail and commerce
Purchasing cXML enhancements		11/11/2019		Procurement and sourcing
Purchasing card processing		10/11/2019		Accounts payable

**Purchasing gift card on e-Commerce f...** ✓

Feature added 9/15/2020  
Retail and commerce  
[Learn more](#)

This feature was enabled on 10/8/2020

**Description**  
When this feature is enabled it allows gift cards to be purchased on e-commerce. It ensures a gift card product can be uniquely identified and the necessary workflows for gift card purchase can be triggered on e-commerce.

**Comments**

## Configure a digital gift card in Commerce headquarters

Digital gift card products should be configured in Commerce headquarters. The process resembles the process for other products. However, the following important steps are specific to the configuration of gift cards for purchase. For more information about how to create and configure products, see [Create a new product in Commerce](#).

- When you configure digital gift card products in the **New product** dialog box, set the **Product type** field to **Service**. (To open the dialog box, go to **Retail and commerce > Products and categories > Products by category**, and select **New**.) Products of the **Service** type aren't checked for available inventory before an order is placed. For more information, see [Create a new product](#).
- On the **Commerce parameters** page, on the **Posting** tab, the **Gift card product** field must be set to **Digital Gift Card**, as shown in the following illustration. If the product is an external gift card, see [Support for external gift cards](#) for more information.

Finance and Operations Preview | commerce parameter | USRT

Save | Options

### Commerce parameters

- General
- Posting**
- Accounting export
- Customer orders
- Inventory
- Prices and discounts
- Number sequences
- POS authentication
- Channel deployment
- Configuration parameters
- Transaction validation

**Set up the retail parameters for posting**

**Gift card**

+ New | Delete

Start date	End date	Gift card number
1/1/2015		GC#####

Gift card product: **Digital Gift Card** (highlighted in red)

transactions to print: 0

Journal: [dropdown]

Allow concluding gift card transacti...:  No

Tax on gift cards:  No

Gift card company: USRT

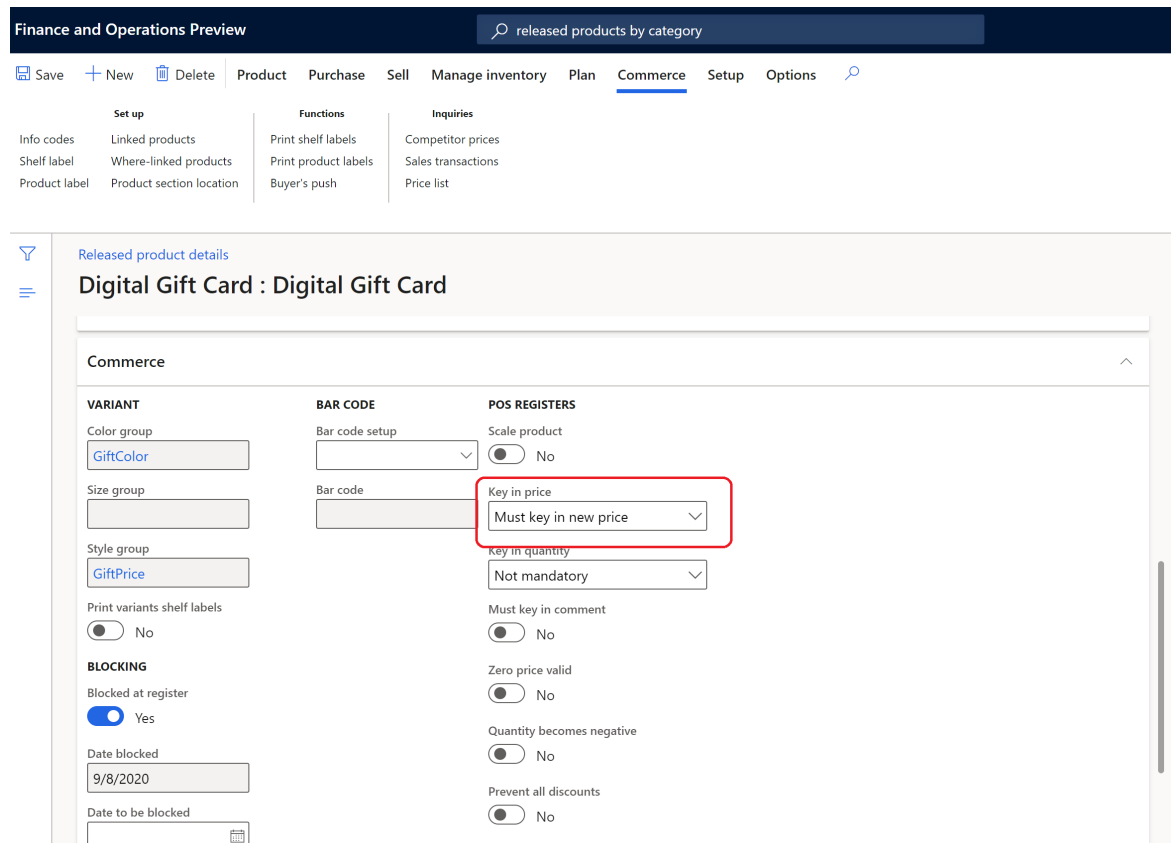
Void journal name: [dropdown]

**Credit memo**

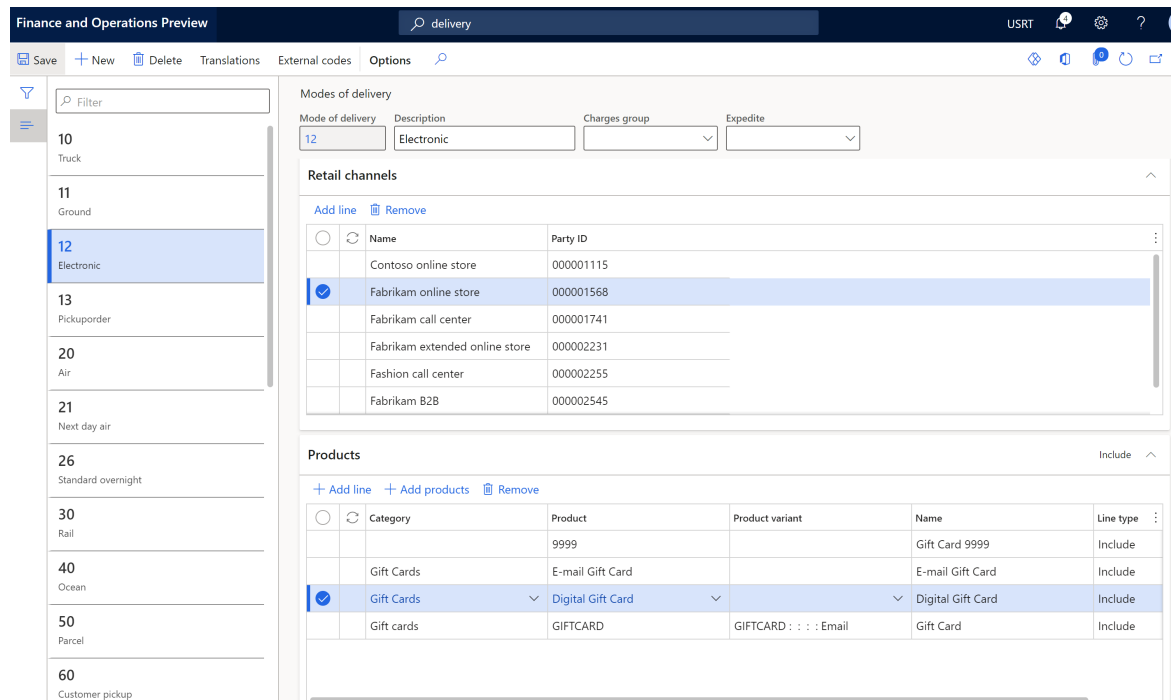
Allow concluding credit memo trans...:  No

- If a gift card must support multiple predefined amounts (for example, \$25, \$50, and \$100), the **Size** dimension should be used to set up those predefined amounts. Each predefined amount will be a variant. For more information, see [Product dimensions](#).
- If customers must be able to specify a custom amount for a gift card, first set up a variant that allows for a custom amount. Next, open the product from the **Released products in category** page, and then, on

the Commerce FastTab, set the **Key in price** field to **Must key in new price**, as shown in the following illustration. This setting ensures that customers can enter a price when they browse the product on a PDP.



- The mode of delivery for a digital gift card must be set to **Electronic**. On the **Modes of delivery** page (**Retail and commerce > Channel setup > Modes of delivery**), select the **Electronic** mode of delivery in the list pane, and then add the digital gift card product to the grid on the **Products** FastTab, as shown in the following illustration. For more information, see [Set up modes of delivery](#).



- Make sure that an online functionality profile has been created and associated with your online store in Commerce headquarters. In the functionality profile, set the **Aggregate products** option to **Yes**. This setting ensures that all items except gift cards are aggregated. For more information, see [Create an online functionality profile](#).

- To ensure that customers receive an email after a gift card is invoiced, create a new email notification type on the [Email notification profiles](#) page, and set the **Email notification type** field to **Issue gift card**. For more information, see [Set up an email notification profile](#).

## Add product images to the Commerce site builder Media library

You must add product images for digital gift card products to the Commerce site builder Media library. Make sure that the file names of the gift card image files follow your site's naming conventions for product images. For more information, see [Upload images](#).

## Configure a custom amount for a digital gift card in Commerce site builder

If a digital gift card is configured to allow for a custom amount, this behavior must also be enabled in the [buy box module](#) that is used on your site's PDPs. The buy box module supports module configuration to allow for custom amounts. You can also define the minimum and maximum amounts that are allowed for custom amounts.

To configure a custom amount for a digital gift card in Commerce site builder, follow these steps.

1. Go to the buy box module that is used on your site's PDPs. This buy box module might be implemented in a fragment, in a template, or on a page.
2. Select **Edit**.
3. In the properties pane on the right, select the **Allow custom price** check box.
4. Optional: To define minimum and maximum amounts for custom amounts, enter amounts under **Minimum price** and **Maximum price**.
5. Select **Finish editing**, and then select **Publish**.

## Additional resources

[Buy box module](#)

[Checkout module](#)

[Cart module](#)

[Create a new product in Commerce](#)

[Set up modes of delivery](#)

[Product dimensions](#)

[Set up an email notification profile](#)

[Create an online functionality profile](#)

[Support for external gift cards](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Browser requirements for Dynamics 365 Commerce sites

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the browser requirements for Microsoft Dynamics 365 Commerce sites.

Dynamics 365 Commerce sites are best supported by the following web browsers. To find the latest release for each web browser, go to the software manufacturer's website.

- [Microsoft Edge](#) (up to the current version minus 1)
- [Mozilla Firefox](#) (up to the current version minus 1)
- [Google Chrome](#) (up to the current version minus 1)
- [Apple Safari](#) (up to the current version minus 1)

## NOTE

Internet Explorer isn't supported. If you use Internet Explorer, some features might not work, and users might have a substandard experience.

## Additional resources

[Create an e-commerce site](#)

[Manage robots.txt files](#)

[Enable location-based store detection](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Home page overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of the home page in Microsoft Dynamics 365 Commerce.

## Overview

The home page is the default page that shoppers go to when they visit an e-Commerce site. Typically, this page showcases products and promotions by using a combination of marketing modules. The home page should be rich with images and text to keep shoppers engaged.

The following illustration shows an example of a home page that was built by using the module library and the "Fabrikam" theme.

Summer collection online now!

# Fabrikam

Search Karen | (1)

Menswear ▼ Womenswear ▼ Accessories ▼ Contact

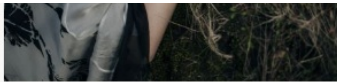
## NEW ARRIVAL

Tasteful accessories walking a fine line between vintage and modern

Shop Now

### The new season has arrived

Explore our new season with soft fabrics as if it's embracing a subtle forest breeze.



[Shop Women](#)



[Shop Men](#)



[Shop Accessories](#)

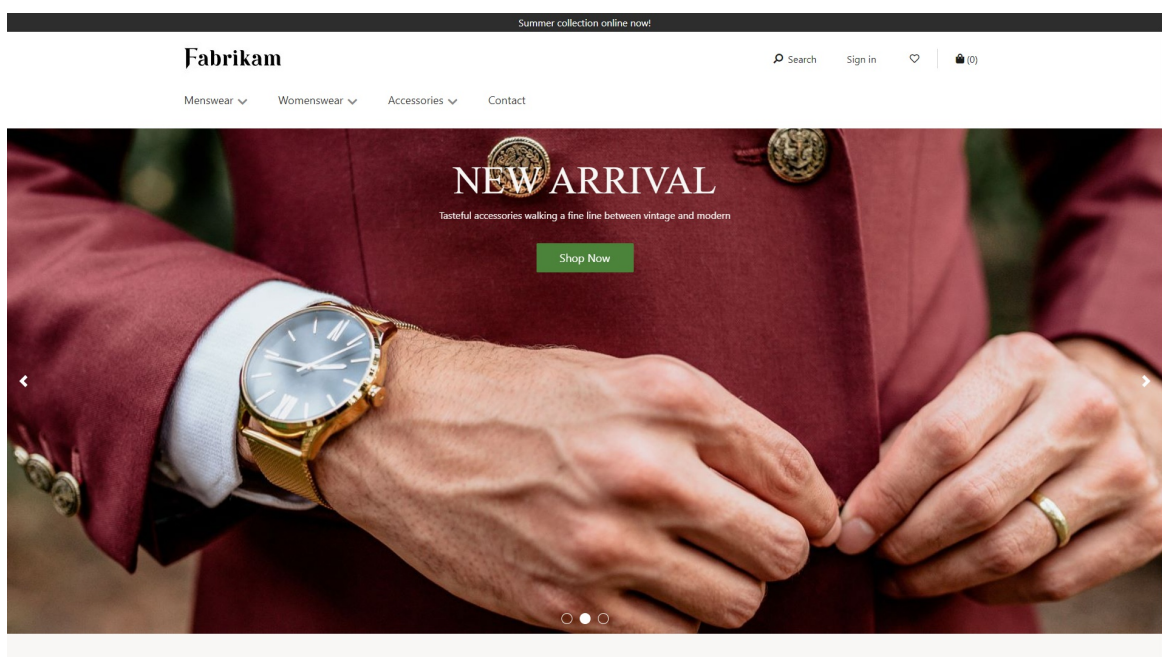


The top of the home page has a header that shows all the product categories and other pages that the retailer wants customers to browse. The bottom of the home page has a footer that contains quick links to various topics that might interest customers.

The main section of the home page can highlight products, categories, or promotions by using various Dynamics 365 Commerce modules:

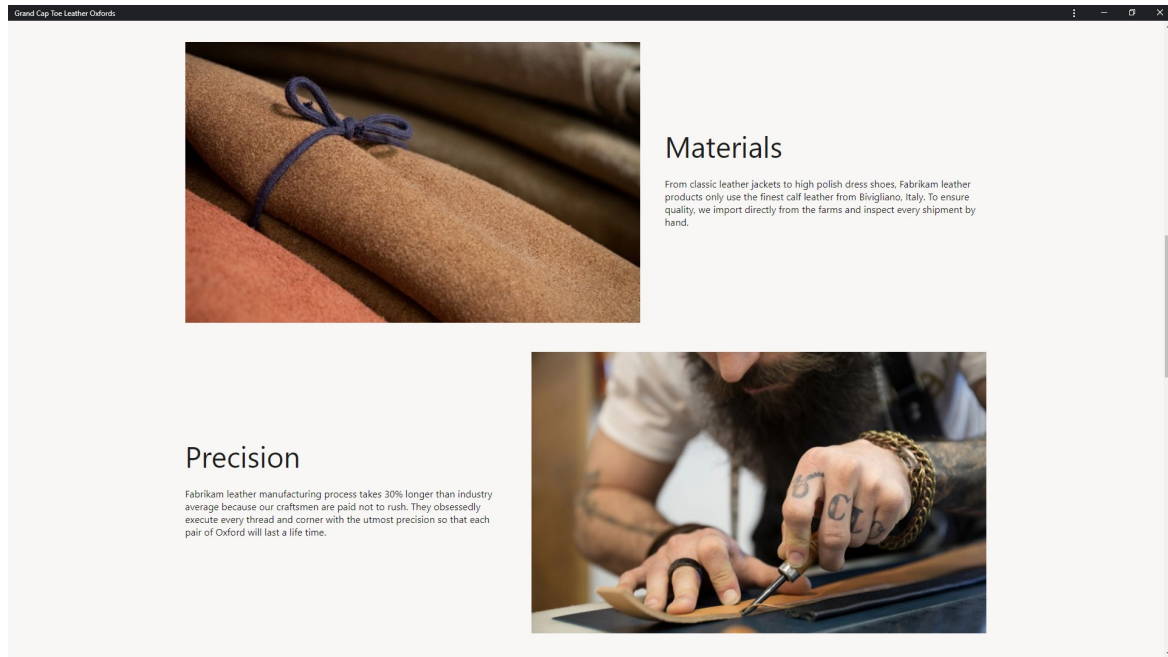
- **Hero** – Typically, the first item at the top of the main section shows one or more "hero" images that highlight new products and promotions in the store. If there are multiple hero images, they are hosted in a carousel module so that users can browse them.

The following illustration shows an example of a home page where the first item in the main section is a hero layout of a content block module that is named "New Arrival."



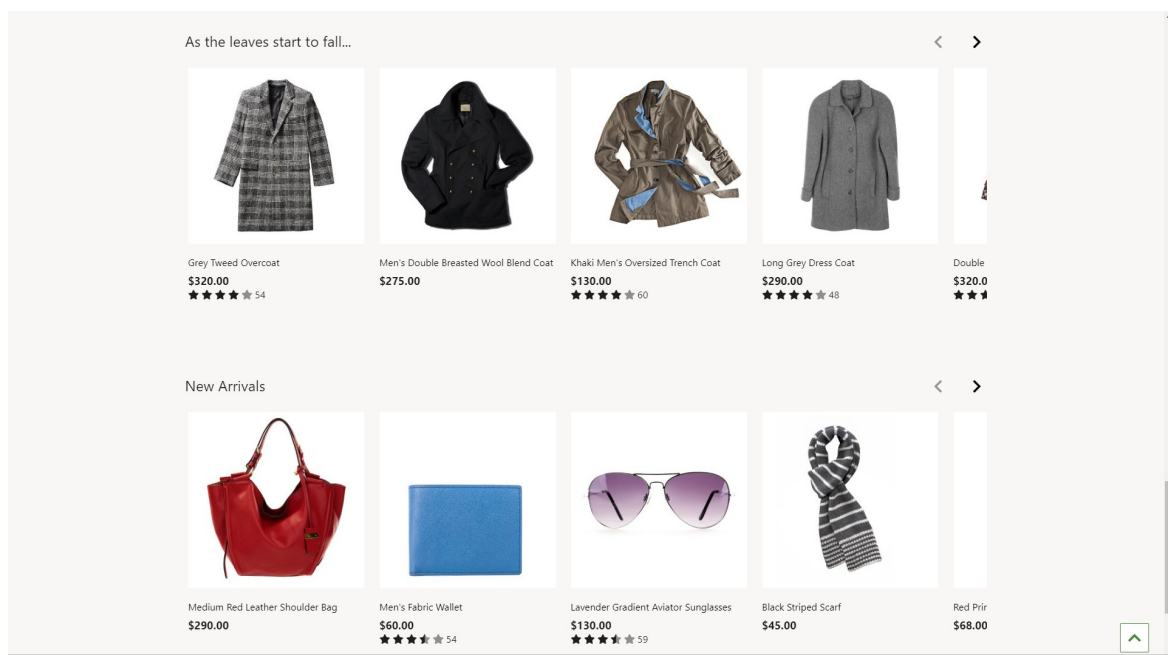
- **Feature** – A feature layout of a content block module is used to market products or promotions by using a combination of images and text. Features layouts can be used independently, or they can be hosted in a carousel module.

The following illustration shows an example of feature layout of a content block module on a home page.



- **Tile** – A tile layout of a content block module is used to showcase multiple products or category of products by using a combination of images and text in a multicolumn layout. In the illustration of a home page that appears earlier in this topic, a tile layout is used for the three-column rendering of the **Shop Women**, **Shop Men**, and **Shop Accessories** items.
- **Video player** – A video player module can be used to showcase video content on the home page. The illustration of a home page that appears earlier in this topic includes a video player module.
- **Text block** – A content rich block module can be used to present text content on the home page in a single-column or multicolumn layout.
- **Product recommendations** – Product recommendations modules are used to show lists, such as **New**, **Trending**, and **Best Selling** on the home page. These lists showcase products based on shopping trends, and they can be algorithmically generated or manually curated. They help customers quickly discover top products and then continue to shop.

The following illustration shows an example of product recommendations modules on a home page.



**NOTE**

All the modules that are listed here can be used on any site page. However, their placement on the home page is important because that page is where customers first interact with your site.

## Additional resources

[Product details pages overview](#)

[Cart and checkout pages overview](#)

[Account management pages overview](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product details pages overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides an overview of product details pages (PDPs) in Microsoft Dynamics 365 Commerce.

## Overview



A PDP provides detailed information about a product, and lets customers select product options such as a size, style, and color. A PDP should showcase all the product information that a customer requires to make a purchase decision.

The following illustration shows an example of a PDP.

Summer collection online now!

**Fabrikam** Search Karen ▾ ♥ 🛒 (1)

[Menswear ▾](#) [Womenswear ▾](#) [Accessories ▾](#) [Contact](#)



### Medium Red Leather Shoulder Bag

\$290.00

A spacious, functional bag meets a trendy red leather, creating the ultimately convenient shoulder bag.

Overnight shipping available

Quantity

– 1 +

[Add to Cart](#) ♥

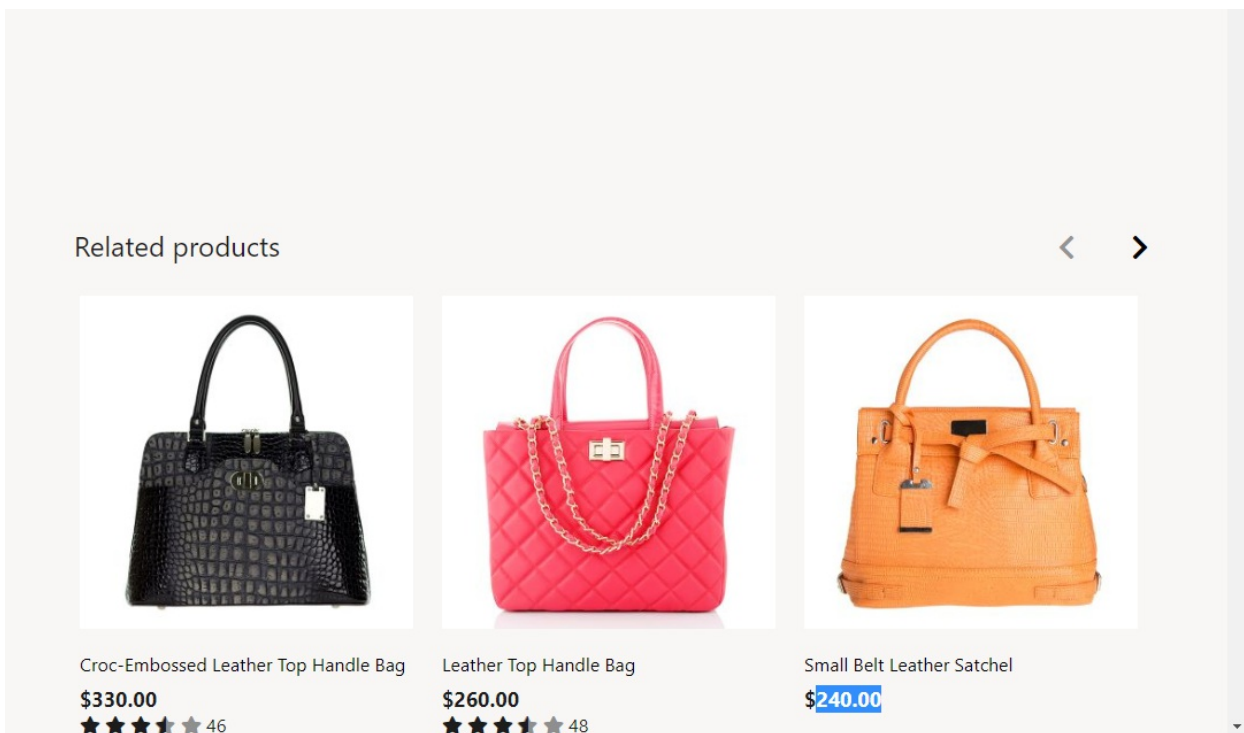
Buy now, pick up in a store

Search for product availability at stores within 50 miles of you.

[Find a store](#)

#### Product specifications

<b>Brand</b>	Northwind Traders
<b>Bag type</b>	Satchel
<b>Closure Type</b>	Snap



## Header and footer modules

The top of a PDP has a header that shows all the product categories and other pages that the retailer wants customers to browse. The bottom of the page has a footer that contains quick links to various topics that might interest customers.

## Buy box module

The most important module on a PDP is the buy box module, which appears as the first item in the main section of the page. A buy box module shows important product information, such as the product name, the product description, the product price, product images, and product ratings.

The buy box module lets the customer select product options (for example, a size, style, and color) and add the product to the cart. It also lets the customer buy the product online and pick it up in a store. The buy online and pick up in store module uses integration with Bing Maps application programming interfaces (APIs) to find nearby stores or stores in another location that the customer specifies.

A buy box module requires a product ID. This ID is derived from the page context. If a buy box module is added to a page where the page context doesn't include a product ID, it won't render the information correctly.

## Product specifications module

The product specifications module can be used to showcase additional details about the product. These details are taken from product attributes in Commerce. The product specifications module shows every attribute where the **visible** property is set to **true**. It requires a product ID to retrieve the product attributes.

## Recommendations module

The recommendations module is an important module on a PDP. While customers browse for products, more product options should be presented to them, so that they can find the correct product and make a purchase. Recommendations help customers easily discover related content and continue to shop.

Different types of recommendation lists are available:

- The **People also like** list is based on machine learning. It uses the transaction history of other customers to

provide recommendations. This list is generated by the recommendations service and resembles "Customers who bought this also bought..." lists. A product ID is required to generate this list.

- A **Related** list can be configured for a product in Commerce. For example, for a brown leather travel handbag, more handbags that are leather-based or designed for travel purposes can be configured for the related list. Other types of related lists, such as **Accessories** and **More like this**, can also be configured in Commerce. A product ID is required to generate this list. Therefore, if it's added to a home page, where the page context doesn't include a product ID, the list will be empty.
- Algorithmically generated recommendation lists, such as **Trending**, **Best Selling**, and **New**, can be used on PDPs. Although these lists might not be directly related to the product on the PDP, they are another way to help customers find products that might interest them. These types of lists don't require a product ID. They are generic lists that are generated based on shopping patterns across the site.
- Editorial lists are manually curated lists. For example, a retailer might decide to manually curate lists of products that it wants to showcase.

## Ratings and reviews modules

Three modules can be used to show and add reviews:

- **Reviews** – This module lists ratings and reviews that have been provided by other customers. Customers can sort and filter the reviews. This module also lets customers like or dislike reviews, and report issues.
- **Write review** – This module lets customers write their own reviews of a product.
- **Ratings histogram** – This module includes a histogram that shows the ratings trend for a product.

For more details, see [Ratings and reviews overview](#).

## Marketing modules

If marketing content is unique to a specific product, any marketing module can be added to the PDP. You can add marketing modules to a PDP by "enriching" the page. For more details, see [Enrich a product page](#).

## Additional resources

[Home page overview](#)

[Cart and checkout pages overview](#)

[Account management pages overview](#)

[Enrich a product details page](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cart and checkout pages overview

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic provides an overview of the cart and checkout pages in Microsoft Dynamics 365 Commerce.

## Overview

The cart page of an e-Commerce website shows all items that a customer has added to the cart. The cart page is built by using the cart module. The cart module is a container that hosts all the modules that are required to showcase items in the cart. The cart module can also use other modules to show the order summary and any promotional codes that have been applied to the customer order.

The checkout page of an e-Commerce website presents a step-by-step flow that customers follow to enter all the information that is required to place an order. A checkout module can include modules that handle the shipping address, shipping methods, billing information, order summary, and other information that is related to customer orders.

## Cart page

The cart page serves as the shopping bag and includes all the items that have been added to the cart.

The following illustration shows an example of a cart page that was built by using the module library and the "Fabrikam" theme.

The screenshot displays the Fabrikam e-commerce website's shopping bag and order summary. At the top, a dark banner reads "Summer collection online now!". The Fabrikam logo is on the left, and navigation links for "Menswear", "Womenswear", "Accessories", and "Contact" are below it. On the right, there are search, user profile ("Karen"), heart, and cart (1) icons.

The main content area is titled "Shopping bag" and features a product card for "Grand Cap Toe Leather Oxfords". The product image shows a brown leather shoe. Details include "Size: 7.5", "Color: Brown", and "Style: Regular". The quantity is set to 1, and the price is \$390.00. Below the product card, there is a "Ship to an address" section with a "Pick this up" button, and links for "Move to Wishlist" and "Remove".

To the right of the product card is the "Order summary" section. It includes a "Promo Code" field with an "Apply" button. The summary shows: Subtotal \$390.00, Estimated shipping "To be calculated", Estimated tax "To be calculated", and Order total \$390.00. Below this, it states "Amount due \$390.00" with a note "\*Before applicable taxes". At the bottom of the summary are two buttons: "Checkout now" (green) and "Back to shopping" (white).

At the bottom of the page, there is a "Frequently bought together" section with navigation arrows.





Tailored Patchwork Dress Shoes

**\$214.00**  
★★★★★ 49



Suede Dress Shoes

**\$195.00**  
★★★★★ 57



Classic Leather Dress Shoes

**\$145.00**

### Trending



Small Belt Leather Satchel

**\$240.00**



Retro Americana Watch

**\$495.00**  
★★★★★ 1



Diamond Patterened Crew Neck Sweater

**\$120.00**

The main body of the cart page shows all the items that the customer has added to the cart. All applicable discounts are showcased. These discounts include complex discounts. Examples include "Buy 3 items and get 10% off" or "Buy a bottle and a backpack to get 10% off." The order summary module shows the amount that is due after discounts, shipping, taxes, and so on, have been applied. There is also a promo code module that lets the customer apply or remove promotional codes.

A customer can shop anonymously or as a signed-in user. If a customer is signed in, items in the cart are preserved between sessions. In this way, the customer can continue to shop from multiple devices.

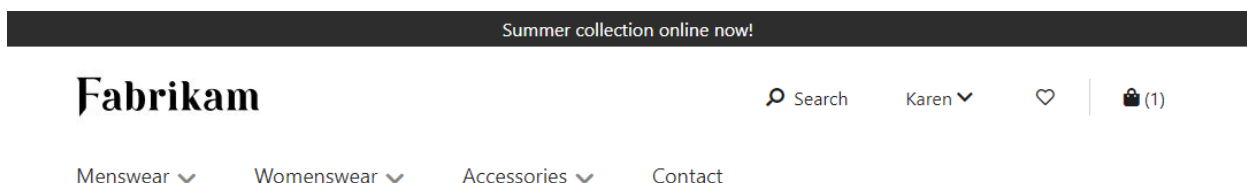
From the cart, the customer can proceed to checkout. A customer can initiate checkout as a guest user or as a signed-in user.

For information about how to author a cart page, see [Add a cart module to a page](#).

## Checkout page

The checkout page is where customers enter the information that is required to place an order.

The following illustration show an example of a checkout page that was built by using the module library.



## Checkout

### 1. SHIPPING ADDRESS

[Change](#)

Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
📞 4257058000

### 2. DELIVERY OPTION

[Change](#)

Standard - Free

### 3. PAYMENT METHOD

[Change](#)

*Notice: Payment information is for demonstration purposes only, and no items will be delivered.*

#### Card information

Karen Berg  
Card ending in 1111  
Expires 10/2020

#### Billing address

Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
📞 4257058000

### 4. CONTACT INFORMATION

[Change](#)

Karen.berg@microsoft.com

[Place order](#)[Back to shopping](#)

## Order summary

Subtotal	<b>\$390.00</b>
Estimated shipping	<b>Free</b>
Estimated tax	<b>\$25.35</b>
Order total	<b>\$415.35</b>

Amount due **\$415.35**

[Place order](#)[Back to shopping](#)[Edit Cart](#)

#### Shipping (1 item)



#### Grand Cap Toe Leather Oxfords

Size: 7.5  
Color: Brown  
Style: Regular  
Quantity: 1  
\$390.00

#### Customer Service

FAQ  
Returns & refunds  
Terms & conditions  
Privacy Policy

#### Fabrikam Store

Store locations  
Store hours  
Store events  
Fabrikam store support

#### About us

Our story  
Careers with Fabrikam  
News

#### Follow us



The main body of the checkout page is where all the order information is collected. This information includes the

shipping address, delivery options, and payment information. Checkout has a step-by-step flow, because the information must be entered in a specific order to be processed. For example, the shipping address must be entered before the shipping costs can be calculated and the payment can be authorized.

### **Shipping address**

A shipping address is required if items must be shipped. The format of shipping addresses for each locale can be configured in Dynamics 365 Commerce. For example, if the items will be shipped to the United States, the shipping address must include a street address, state, and ZIP Code. Some basic input validation is done for shipping address fields, such as validation for alphanumeric characters, maximum length, and numbers. Although the validity of the address itself isn't verified, this verification can be done by using customized third-party services.

The shipping address is applied to all items in the cart that the "ship" option is selected for. If you use the checkout flow that is provided in the module library, individual cart items can't be shipped to different addresses. If you require this capability, it can be implemented through customization of the checkout modules.

After the shipping address is provided, the shipping methods that are available from the Dynamics 365 Commerce online store are shown. The shipping methods and the addresses that they support can be configured in Commerce.

### **Payment**

The next step in the checkout flow is payment. In e-Commerce, multiple methods of payment can be used to place orders, such as credit cards, gift cards, and loyalty points. A combination of these payment methods can also be used. Depending on the payment methods that are used, additional information might be required. For example, a credit card payment requires a billing address. Credit card payments are processed by using the Adyen Payment Connector.

#### **Loyalty points**

During the checkout flow, a customer who is a member of a loyalty program and who has accrued loyalty points can redeem those loyalty points for an order. The loyalty points module is shown only if the customer has an existing loyalty membership. For non-members and guest users, this module is hidden.

#### **Gift cards**

The module library lets internal gift cards be redeemed for an order. To apply an internal gift card, the customer must be signed in. For additional security, we recommend that you customize the flow by using a personal identification number (PIN) for internal gift cards.

### **Signed-in and guest users**

The customer can complete the checkout process as a guest user or a signed-in user. If the customer signs in, account information such as saved shipping addresses and saved credit card details is automatically retrieved.

### **Order summary**

Checkout shows a summary of the line items in the cart, so that the customer can verify the order before he or she places it. The line items can't be edited during the checkout flow. However, a link to the cart is provided in case the user wants to go back and edit line items.

After the customer provides shipping and billing information, the order summary reflects the amount that is due after loyalty points, gift cards, and other payments have been applied.

### **Order confirmation and email**

When the customer places an order, a confirmation number is provided. At this point, the payment has been authorized but not charged. The payment will be charged only when the order is fulfilled (that is, when it's either shipped or picked up).

After an order is created, an order confirmation email is sent to the customer.

For more information about how to author a checkout page, see [Add a checkout module to a page](#).

## Additional resources

[Home page overview](#)

[Product details pages overview](#)

[Account management pages overview](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Account management pages overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of account management pages in Microsoft Dynamics 365 Commerce.

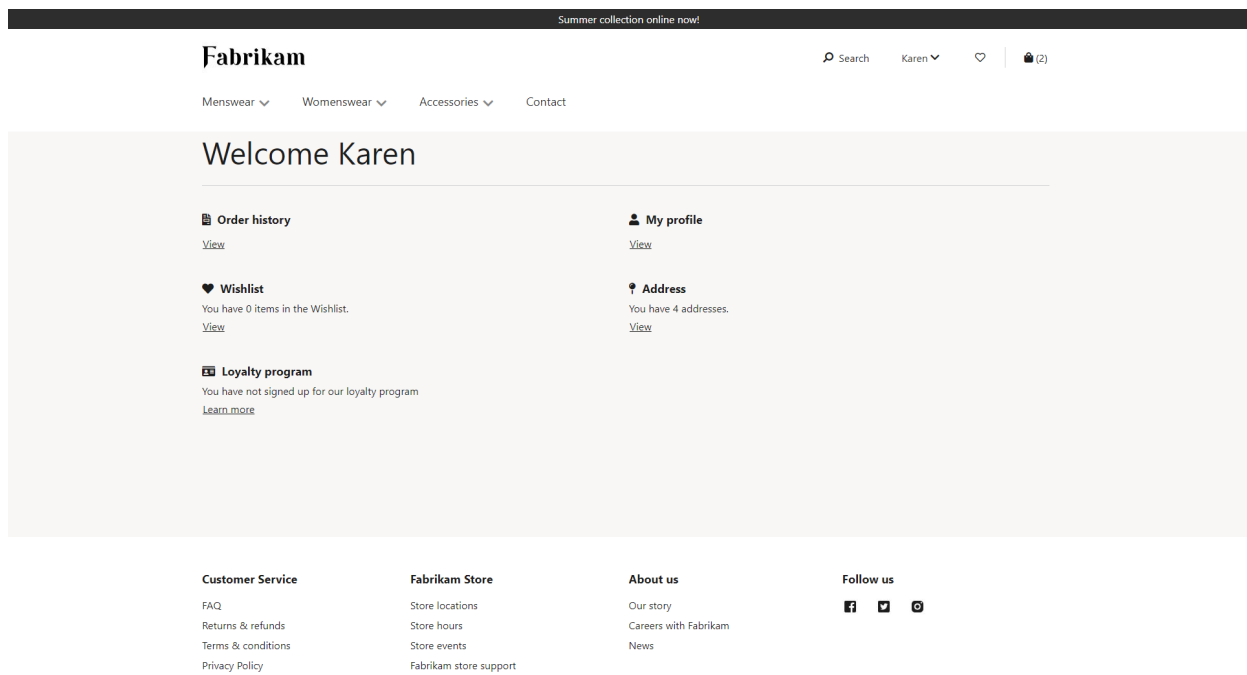
## Overview

Account management pages let customers view information that is related to their account and orders. Account management pages include the account management landing page, and pages for the user's profile, addresses, order history, order details, loyalty points, and wish list.

### Account management landing page

When a customer signs in and selects **My Account**, the account management landing page is opened. This page provides a quick summary of all account-related information, such as the user's profile, orders, wish list, addresses, loyalty points. From this page, the customer can access more details for each area.

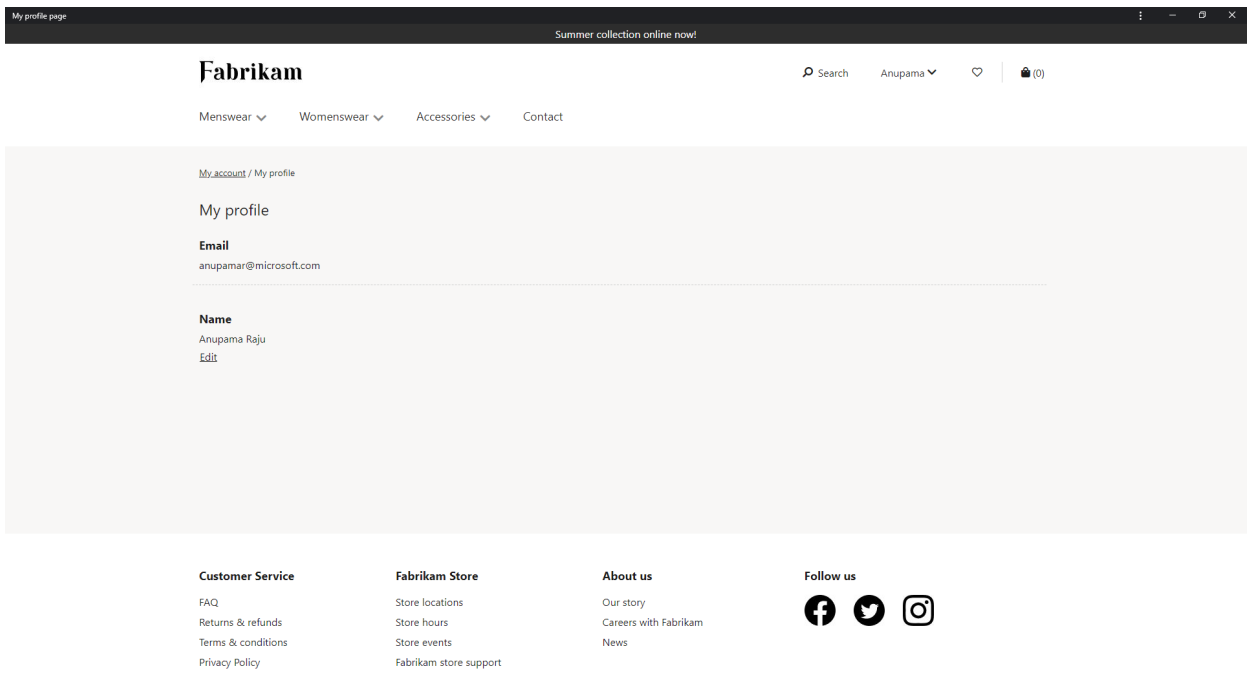
The following illustration shows an example of the account management landing page.



### My profile page

The **My profile** page shows customer's account information, such as his or her name and phone number. The customer can update his or her profile information on this page. This page can be customized so that it includes additional customer account preferences, such as an option for opting in to marketing email.

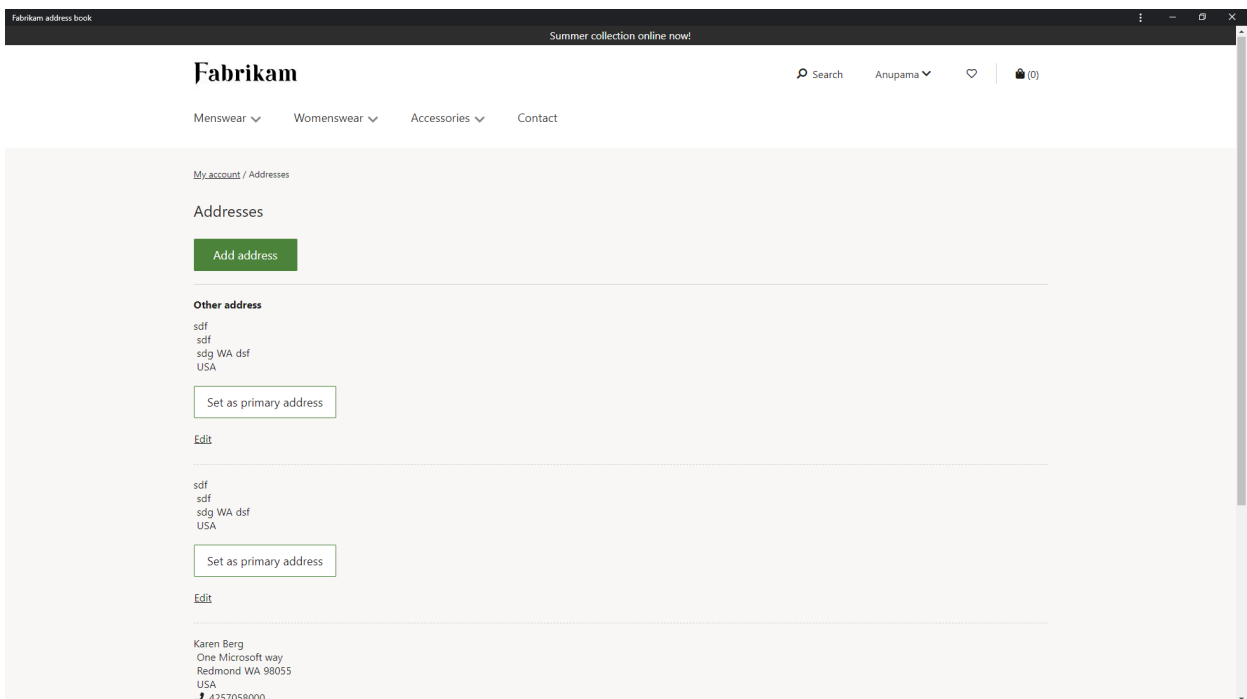
The following illustration shows an example of a **My profile** page that was built by using the module library.



## Addresses page

The **Addresses** page lets the customer add addresses to his or her account. It also shows the list of addresses that the customer has previously added or saved to the account. These addresses are addresses that the customer entered either on this page or while placing an order.

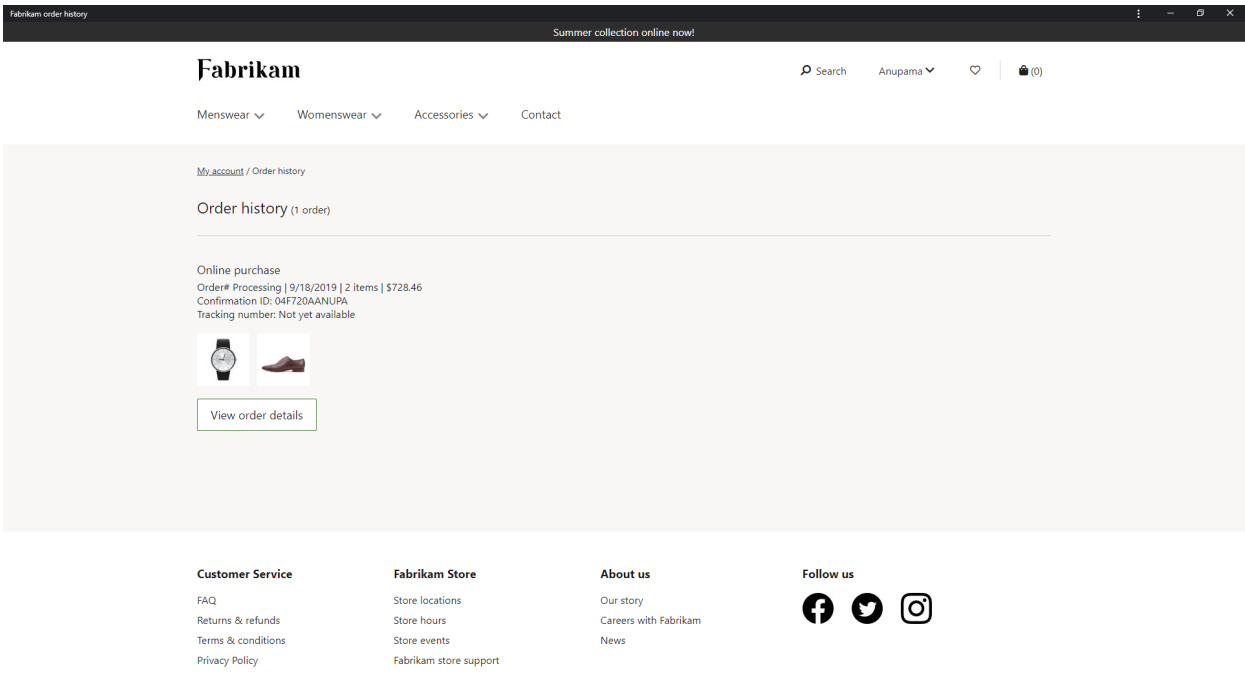
The following illustration shows an example of the **Addresses** page.



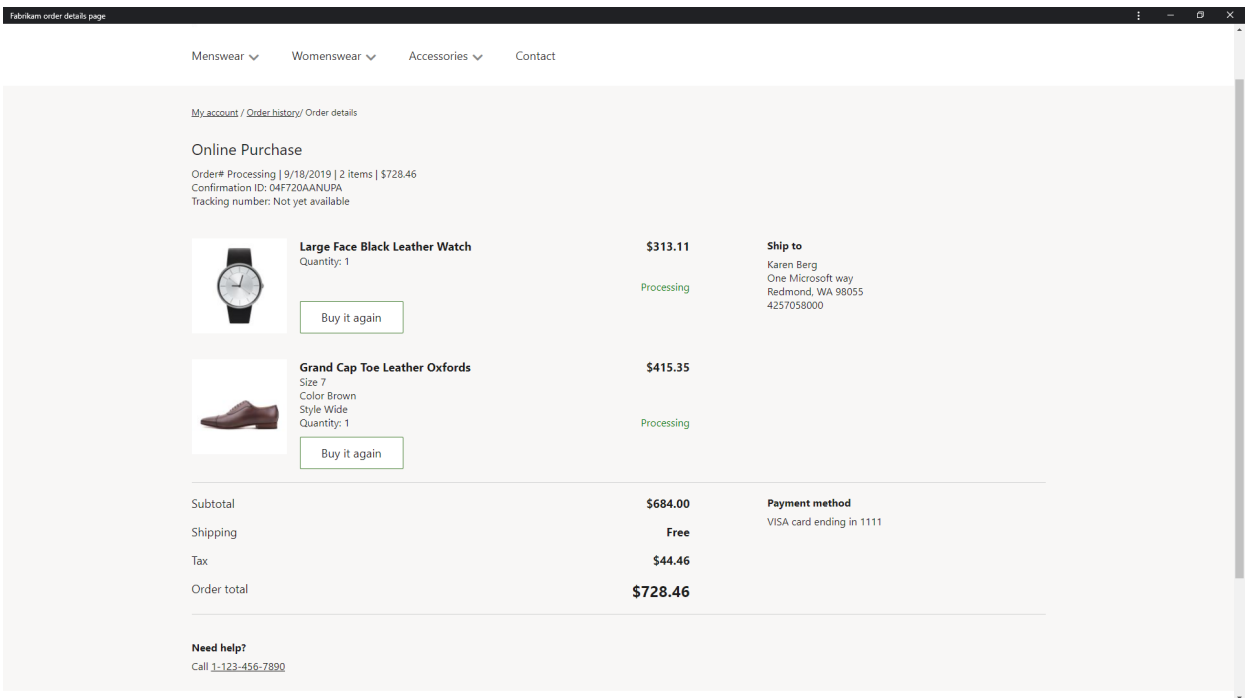
## Order history and Order details pages

The **Order history** page shows a summary of all orders that the customer has submitted by using his or her account. It gives a quick summary of the items that were ordered, the confirmation number, sales ID, tracking information, and other information. If the customer wants to view a more detailed breakdown of each order, there is an **Order details** page. This page includes information such as the shipping address, payment information, discounts, taxes, and shipping costs for the order.

The following illustration shows an example of the **Order history** page.



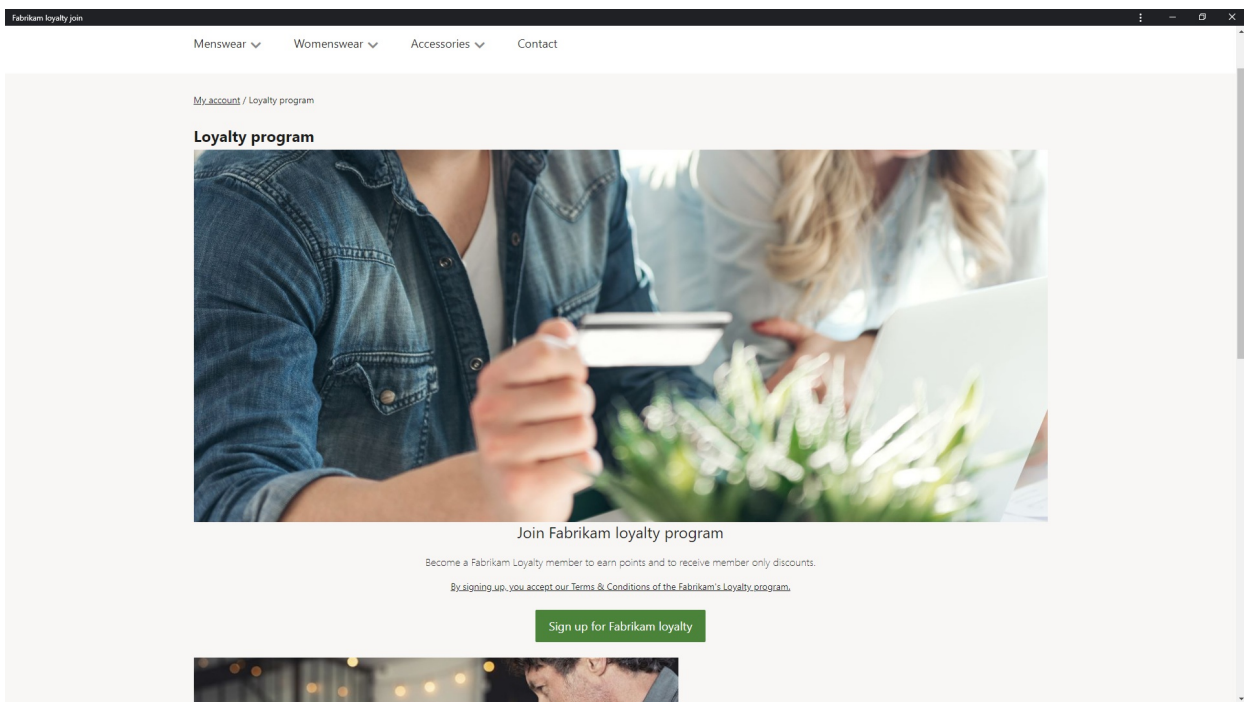
The following illustration shows an example of the **Order details** page.



## Loyalty program page

The **Loyalty program** page lets the customer become a member of a loyalty program. After a customer has signed up for a loyalty program, the **Loyalty program** page include details such as the number of points that have been earned and the number of points that have been redeemed.

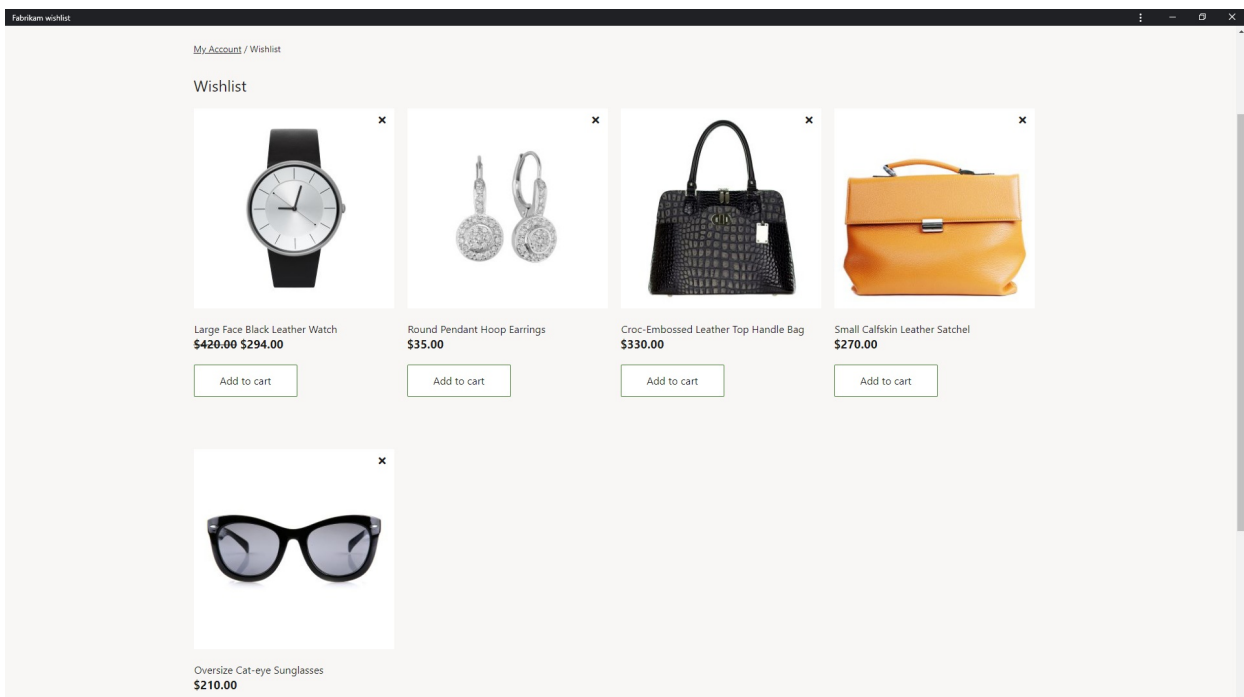
The following illustration shows an example of a **Loyalty program** page.



## Wishlist page

The **Wishlist** page shows a list of the items that the customer has added to his or her wish list. Both products and product variants can be added to the wish list. From this page, the customer can remove an item from the wish list or add an item directly to the cart.

The following illustration shows an example of a **Wishlist** page.



For more information about account management modules and how to author them, see [Account Management](#).

## Additional resources

[Home page overview](#)

[Product details pages overview](#)

[Cart and checkout pages overview](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Authoring page overview

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic provides an overview of the authoring page in Microsoft Dynamics 365 Commerce.

## Overview

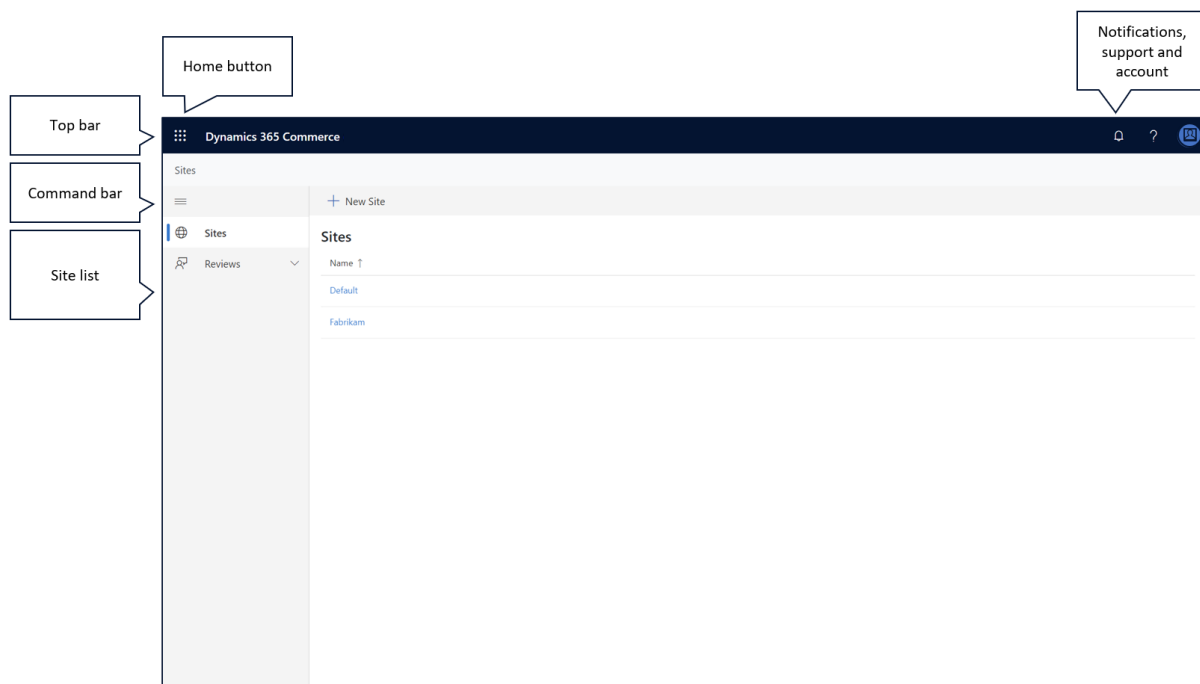
Websites can be created to support various business needs. They can represent a whole business, offer a single channel of the business, or target a specific audience or segment of an audience. For example, a clothing manufacturer might have a website that showcases all the brands that it owns. The same clothing manufacturer might then have a separate website for each of those brands, and also a set of websites that feature luxury fashion, outdoor fashion, and children's fashion.

Dynamics 365 Commerce supports the creation and management of multiple websites, and each website can have its own appearance and content. The authoring page serves as a common access point for these websites. You can use it to sign in to and out of the system, and to create new websites.

Currently, the authoring page consists of the following sections.

- **Top bar** – The top bar appears at the top of the authoring page. It provides easy access to e-Commerce tools, notifications, support links, and user sign-in.
- **Command bar** – The command bar appears below the top bar. It can be used to create new websites.
- **Site list** – The site list fills all the space below the command bar. It provides a comprehensive list of websites and the domains that are associated with them.

The following illustration shows the authoring page.



## Use the Home button to select a tool

The **Home** button is in the upper-left corner of the authoring page. It provides easy access to other e-Commerce tools. When you select this button, a menu of the tools that you can use is opened. When you select a tool, the menu is closed, and the selected tool is loaded in the browser.

## View and clear notifications

The **Notifications** button is one of the buttons in the upper-right corner of the authoring page. It looks like a bell. By selecting this button, you can view all the notifications that have been sent to you.

Notifications are used throughout the authoring tool to inform you when actions have been completed. For example, a notification might state, "Your page has been published" to inform you that a publish action was successful.

Notifications can also inform you about errors that were encountered while an action was being performed. Select the error notification to open its message. The information in this message can help you resolve the error.

You can clear notifications from the notification menu by selecting **Remove** at the bottom of the notification message. To clear notifications in bulk, select **Remove all** at the bottom of the notification menu.

## Get help with the authoring tool

The **Help** button is another button in the upper-right corner of the authoring page. It looks like a question mark. When you select this button, a menu of the following predefined options is opened:

- **Site development help** – If you select this option, the documentation for creating a new website is opened on a new browser tab.
- **Feedback and support** – Select this option to open a Microsoft Yammer channel where you can leave feedback about the authoring tool or request support.
- **Privacy and cookies** – If you select this option, the Microsoft privacy statement is opened on a new browser tab.
- **About** – Select this option to open a message box that contains information about the authoring tool and the version that you're currently using.

## Sign in to and out of the authoring tool

The **My account** button is another button in the upper-right corner of the authoring page. It looks like a colored circle. By selecting this button, you can see which account you used to sign in, and you can also sign out of that account as you require.

To sign in to or out of the authoring tool, follow one of these steps.

- If you aren't already signed in to the authoring tool, select **My account > Sign in** to sign in.
- If you're already signed in and want to sign out, select **My account > Sign out**.

## Change the display language of the authoring tool

You can also use the **My account** button to change the language of the text strings that appear in the authoring tool.

To change the display language, follow these steps.

1. Select **My account > Change language**. A dialog box appears.
2. Select one of the user languages, and then select **Save**.

## Create a new website

Dynamics 365 Commerce supports the creation and management of multiple websites, and each website can have its own appearance and content.

To create a new website, follow these steps.

1. On the command bar, select **New Website**. A dialog box appears.
2. Enter the following required information for the new website:
  - **Site name** – Enter the name of the website. This name isn't shown to website customers. Instead, it's used in the site list and other places in the authoring tool.
  - **Site administers security group** – Enter the full name of the Microsoft Azure Active Directory (Azure AD) security group that contains the users who should have administrative access to the website. The admin group name, together with the other permissions for the website, can be changed after the website is created.
  - **Default channel** – Enter the default merchandizing channel that should be associated with the website. The default channel determines the products that can be sold through the website.
  - **Default language** – After you specify the default channel, select the default language for the channel. The default channel defines the language that products are shown in if the customer doesn't specify a preferred language.
  - **Default market** – Enter the default market for the website. The default market defines the market that is shown if the customer doesn't specify a preferred market.
  - **Domain** – Select the web domain that should be associated with the website. This domain is the domain that the website's customers will go to in their browser.
3. Select **OK**. The new website is created.

#### NOTE

Creation of a new website can take up to 60 seconds. After the process is completed, a notification appears in the notification area. Additionally, the website appears in the site list and has the site name that you entered.

## Select a website to author

The site list provides a comprehensive list of the websites that are associated with the e-Commerce system. Websites appear in alphabetical order. The domain that is associated with each website is also shown. To view the contents of a website and start to author pages, select the name of the website. The authoring tool and the content for the website are loaded.

After the authoring tool is loaded, you can select **Home** to return to the authoring page.

## Additional resources

[Manage e-Commerce users and roles](#)

[Search engine optimization \(SEO\) considerations for your site](#)

[Manage Content Security Policy \(CSP\)](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage e-Commerce users and roles

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to grant users access to the authoring environment for your Microsoft Dynamics 365 Commerce site.

To help control user access and grant users permission to perform specific tasks, the site authoring environment uses security groups that you create in Microsoft Azure Active Directory (Azure AD). You first assign a new or existing security group from Azure AD to each role in the authoring environment. You then grant or revoke permissions for individual users by either adding those users to an appropriate security group or removing them from a security group.

## Overview of roles in the authoring environment

The Dynamics 365 for Commerce authoring environment supports the following roles.

ROLE	DESCRIPTION
System Administrator	Users who have this role have all privileges for all tools, and for all ratings and reviews. They can also create sites.
Administrator	Users who have this role have all privileges for all tools and RnR in a given site structure.
Web Producer	Users who have this role can create pages, fragments and templates, upload and manage assets, and enrich products and categories.
Reader	Users who have this role can view pages, templates, assets, fragments, layouts and settings, but may not make changes.
RnR Moderator	Users who have this role can moderate product reviews.

## System Administrator role

When you provision Dynamics 365 Commerce in the Microsoft Dynamics Lifecycle Services (LCS) environment, you're asked to provide a security group for the **System Administrator** role. This role is then automatically applied to all sites that you create in the environment that you're configuring. The security group for this role can be updated only in LCS. On the **Site Administration** page for all sites, it appears as read-only and is for informational purposes only.

## Administrator role

When you create a new site in Commerce, you're asked to provide a security group for the **Administrator** role. See the table earlier in this topic for an overview of the permissions that this role grants.

## Add or update security groups

After your site is created, only users who are in the security groups that are associated with the **System Administrator** and **Administrator** roles can access the authoring environment for that site. To assign users to the **Web Producer**, **RnR Moderator**, and **Reader** roles, you must assign security groups to those roles. To add

a security group to a role, or to update a security group that is currently assigned to a role, follow these steps.

1. Go to the site that you want to update.
2. In **Site management**, open the **Security** page.
3. Select the role to modify.
4. Add security groups to roles, or remove security groups from roles.

## Additional resources

[Add script code to site pages to support telemetry](#)

[Search engine optimization \(SEO\) considerations for your site](#)

[Manage Content Security Policy \(CSP\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Apply inventory settings

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic covers inventory settings and describes how to apply them in Microsoft Dynamics 365 Commerce.

## Overview

Inventory settings specify whether inventory should be checked before products are added to the cart. They also define inventory-related merchandising messages, such as "In stock" and "Only a few left." These settings ensure that a product can't be purchased if it's out of stock.

Dynamics 365 Commerce provides estimates of on-hand availability for products. For information about how estimated on-hand availability is calculated, see [Calculate inventory availability for retail channels](#).

In Commerce site builder, inventory thresholds and ranges can be defined for a product or a category. They determine whether inventory can be classified as in stock, low stock, or out of stock. For details, see [Configure inventory buffers and inventory levels](#).

### NOTE

Support for inventory thresholds and ranges is available in the Dynamics 365 Commerce 10.0.12 release.

## Inventory settings

In Commerce, inventory settings are defined at **Site Settings > Extensions > Inventory Management** in site builder. There are four inventory settings, one of which is obsolete (deprecated):

- **Enable stock check in app** – This setting turns on a product inventory check. Buy box, cart, and pick up in store modules will then check product inventory, and will allow a product to be added to the cart only if inventory is available.
- **Inventory level based on** – This setting defines how inventory levels are calculated. The available values are **Total Available**, **Physical Available**, and **Out of stock threshold**. In Commerce, inventory threshold and ranges can be defined for each product and category. The inventory APIs return product inventory information for both the **Total Available** property and the **Physical Available** property. The retailer decides whether the **Total Available** or **Physical Available** value should be used to determine the inventory count and the corresponding ranges for in-stock and out-of-stock statuses.

The **Out of stock threshold** value of the **Inventory level based on** setting is an old (legacy), obsolete value. When it's selected, the inventory count is determined from the results of the **Total Available** value, but the threshold is defined by the **Out of stock threshold** numeric setting that is described later. This threshold setting applies to all products across an e-commerce site. If inventory is below the threshold number, a product is considered out of stock. Otherwise, it's considered in stock. The capabilities of the **Out of stock threshold** value are limited, and we don't recommend that you use it in version 10.0.12 and later.

- **Inventory ranges** – This setting defines the inventory ranges that message are shown for on site modules. It's applicable only if either the **Total Available** value or the **Physical Available** value is selected for the **Inventory level based on** setting. The available values are **All**, **Low and out of stock**, and **Out of stock**.

- When **All** is selected, messages for all inventory ranges, from in stock ("Available" message) to out of stock ("Out of stock" message), will be shown.
- When **Low and out of stock** is selected, messages for all inventory ranges except in stock ("Available" message) will be shown.
- When **Out of stock** is selected, only the "Out of stock" message will be shown.
- **Out of stock threshold** – This old numeric setting will take effect only if the **Out of stock threshold** value is selected for the **Inventory level based on** setting.

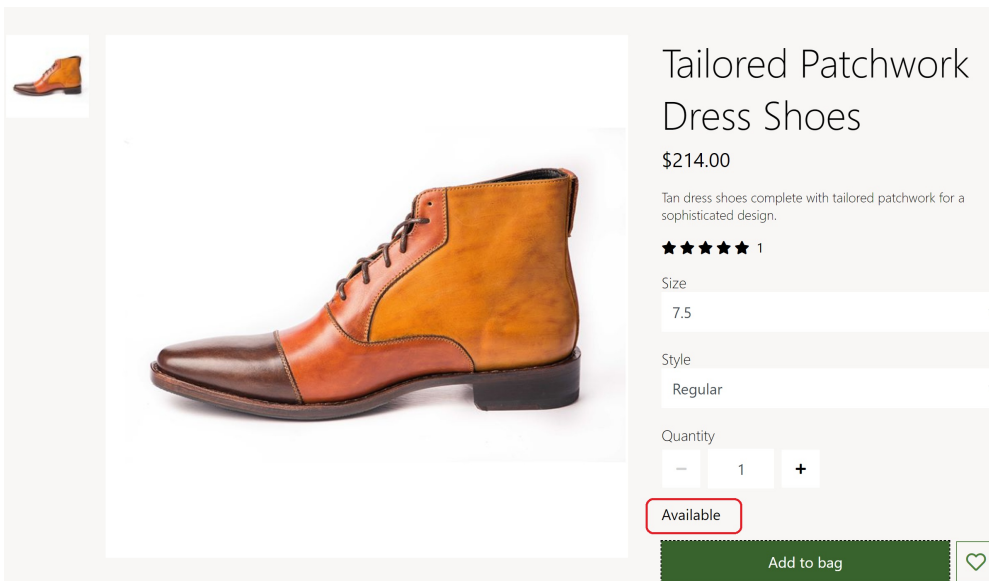
#### IMPORTANT

These settings are available in the Dynamics 365 Commerce 10.0.12 release. If you are updating from an older version of Dynamics 365 Commerce, you must manually update the appsettings.json file. For instructions on updating the appsettings.json file, see [SDK and module library updates](#).

## Modules that use inventory settings

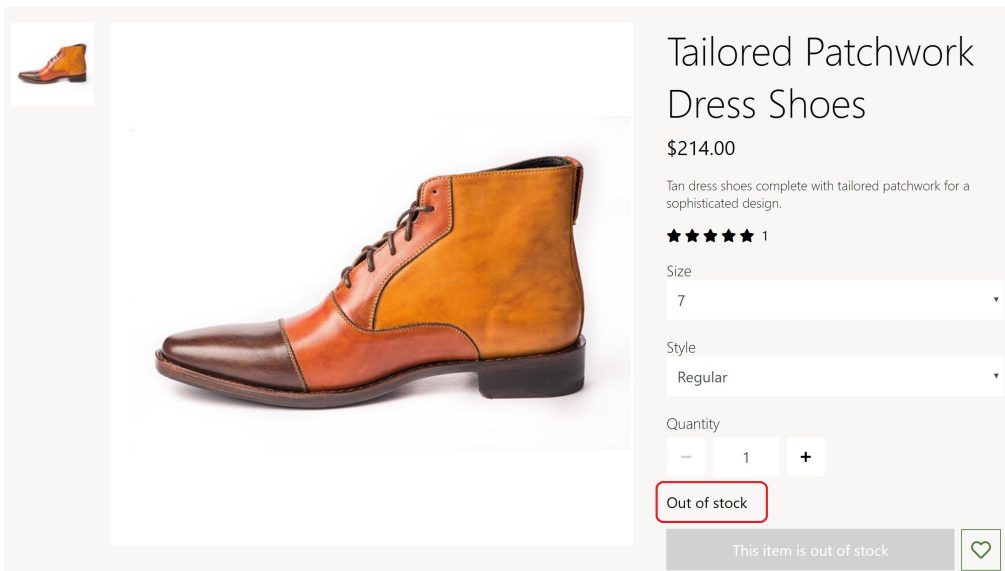
Buy box, wishlist, store selector, cart, and cart icon modules use inventory settings to show the inventory ranges and messages.

The following image shows an example of a product details page (PDP) that is showing an in-stock ("Available") message.

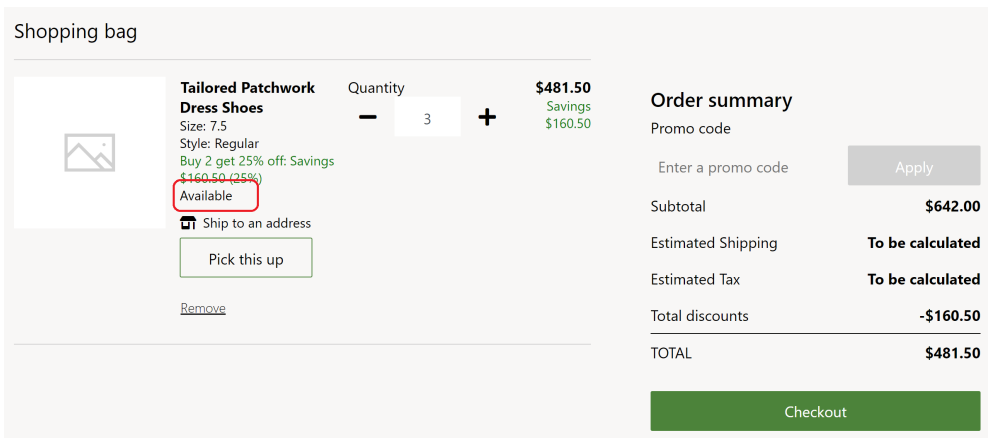


The following image shows an example of a PDP that is showing an "Out of stock" message.





The following image shows an example of a cart that is showing an in-stock ("Available") message.



## Additional resources

[Module library overview](#)

[Configure inventory buffers and inventory levels](#)

[Cart module](#)

[Buy box module](#)

[Account management pages and modules](#)

[Store selector module](#)

[SDK and module library updates](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Search engine optimization (SEO) considerations for your site

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers search engine optimization (SEO) considerations for your site from development to production.

## A site that is under development

While a site is under development, all site pages should have the **NOINDEX** and **NOFOLLOW** meta tags, so that search engines don't index the pages and store development versions of your site in their cache. To do this configuration, you must add the default meta tags module to the site page template. The default meta tags properties will then be available in the SEO properties section in the page editor. You can use these properties to manage the meta tags.

## Soft launch of a site

During a "soft launch," a website is made available to a limited audience or market before the full launch occurs. If you do a soft launch of your website, you should consider leaving the **NOINDEX** meta tags in place. In this way, you help guarantee that the soft launch remains restricted to the limited audience that you want to reach.

## A site that is in production

When a site is in production, you should make sure that all site pages are correctly tagged. Microsoft Dynamics 365 Commerce uses the information that is entered for a page to render all the SEO information on that page. The following modules provide this functionality: category page summary, list page summary, and product page summary.

To optimize search engine indexing, the rendering framework uses both information from the SEO properties that are configured in Dynamics 365 Commerce and module-specific information. For a site that is in production, you should make sure that the robots.txt file allows for indexing of your whole site, and that it contains links to your published site map document. You should turn on the site map generation functionality at **Site Settings > Site maps enabled**.

### **Page SEO settings for internal preview, limited audiences, and all audiences**

Because Dynamics 365 Commerce supports "what you see is what you get" (WYSIWYG) authenticated previews in visual page builder, authors can prepare their page content without having to worry that the information will become visible to site visitors. If a page must be published, but its exposure must be limited, it should have the **NOINDEX** meta tag, so that it won't be indexed by search engines. Then, when the page is ready for all audiences, all the basic SEO metadata should be present, to maximize the efficiency of search engine indexing. Additionally, the **NOLIMIT** meta tag should be removed.

## Additional resources

[Manage e-Commerce users and roles](#)

[Add script code to site pages to support telemetry](#)

[Manage Content Security Policy \(CSP\)](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage Content Security Policy (CSP)

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to manage Content Security Policy (CSP) in Microsoft Dynamics 365 Commerce.

## Overview

CSP is an additional layer of security that helps detect and mitigate some types of web attacks. The purpose of these attacks can range from data theft, to site defacement, to the distribution of malware. CSP provides an extensive set of policy directives that help you control the resources that a site page is allowed to load. Each directive defines the restrictions for a specific type of resource.

When CSP is turned on for an e-Commerce site, it helps enhance security by blocking connections, scripts, fonts, and other types of resources that originate from unknown or malicious sources. In Dynamics 365 Commerce, CSP is turned on by default. However, it will likely require additional configuration for most sites. The Dynamics 365 Commerce online software development kit (SDK) provides a default list of allowed source URLs that style, script, and application programming interface (API) calls can be made from. You can edit this list on the **Extensions** tab in the site builder tool.

For more information about CSP, see [Content Security Policy Reference](#).

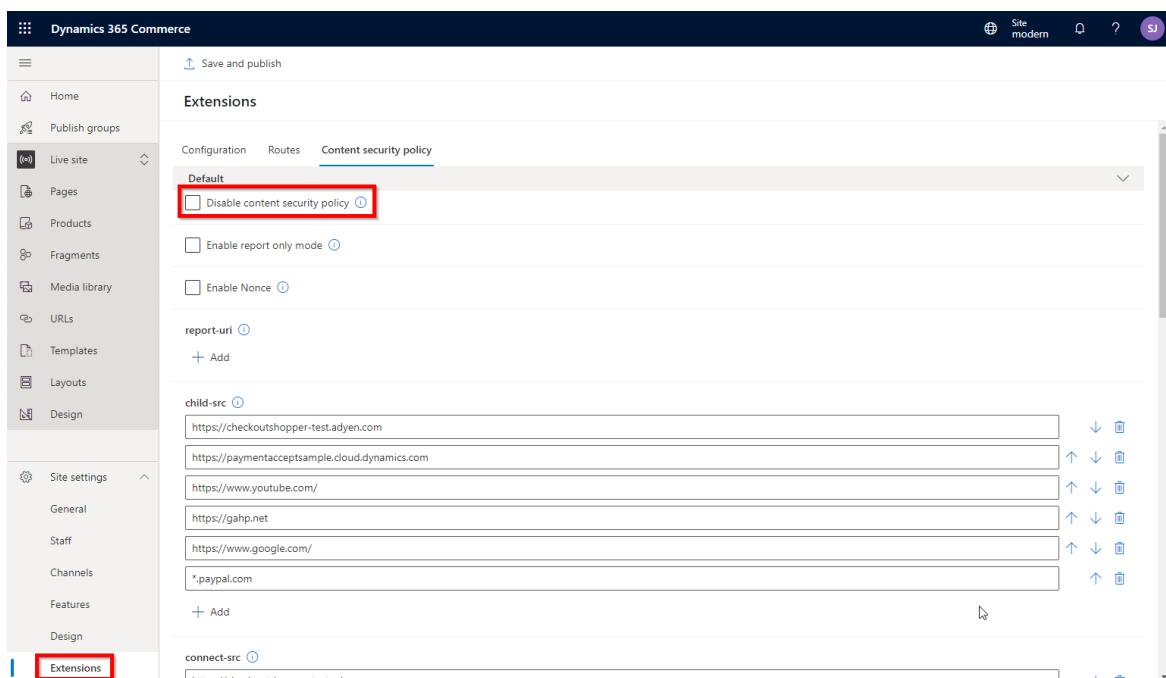
## CSP settings

### Turn off CSP for a site

To prevent CSP from applying policies to your site, you can turn it off for that site in site builder.

To turn off CSP for a site, follow these steps.

1. In site builder, select the site you are working on.
2. Select **Site settings**, and then select the **Extensions** tab.
3. On the **Content security policy** tab, select the **Disable content security policy** check box.



4. Select **Save and publish**.

### Enable report only mode

If CSP is enabled, content security policy will not be enforced, but any violations will be reported to URIs specified by the `report-uri` directive.

To enable report only mode, follow these steps.

1. In site builder, select the site you are working on.
2. Select **Site settings**, and then select the **Extensions** tab.
3. On the **Content security policy** tab, select the **Enable report only mode** check box.

### Enable nonce

Enabling nonce (number used once) will block the execution of all inline scripts except those specified within the [inline script](#) module. A unique cryptographic nonce is generated and added to each script specified in the CSP header.

To enable nonce, follow these steps.

1. In site builder, select the site you are working on.
2. Select **Site settings**, and then select the **Extensions** tab.
3. On the **Content security policy** tab, select the **Enable Nonce** check box.

## CSP directives in Commerce

The following CSP directives can be used on Commerce sites.

DIRECTIVE	DESCRIPTION
child-src	This directive defines valid sources of web workers and nested browsing contexts that are loaded by using elements such as <code>&lt;frame&gt;</code> and <code>&lt;iframe&gt;</code> .
connect-src	This directive defines the URLs that AJAX requests can be made from.
font-src	This directive defines valid sources of fonts.
frame-ancestors	This directive specifies valid parents that may embed a page using <code>&lt;frame&gt;</code> , <code>&lt;iframe&gt;</code> , <code>&lt;object&gt;</code> , <code>&lt;embed&gt;</code> , or <code>&lt;applet&gt;</code> elements. Setting this directive to "none" is similar to specifying the "X-Frame-Options: DENY" directive (which is also supported in older browsers).
frame-src	This directive defines valid sources for nested browsing context loading using elements such as <code>&lt;frame&gt;</code> and <code>&lt;iframe&gt;</code> .
img-src	This directive defines valid sources of images.
media-src	This directive defines valid sources of audio and video, such as HTML5 <code>&lt;audio&gt;</code> and <code>&lt;video&gt;</code> elements.
object-src	This directive defines valid sources of plug-ins, such as <code>&lt;object&gt;</code> , <code>&lt;embed&gt;</code> , and <code>&lt;applet&gt;</code> elements.

DIRECTIVE	DESCRIPTION
report-uri	This directive defines URI(s) that the browser will post CSP violation reports to. These violation reports consist of JSON documents sent via an HTTP POST request to the specified URI.
script-src	This directive defines valid sources of JavaScript.
style-src	This directive defines valid sources of stylesheets.

### Example: Configure a CSP directive

The following example procedure shows how to configure a CSP directive so that an external script can be called from your site.

1. In site builder, select the site you are working on.
2. Select **Site settings**, and then select the **Extensions** tab.
3. On the **Content security policy** tab, under **script-src**, select **Add**, and then enter the full URL of the external script that should be called.

The screenshot shows the Dynamics 365 Commerce Site Management interface. The breadcrumb navigation is 'Sites > Fabrikam > Site Management'. The page title is 'Dynamics 365 Commerce'. The left sidebar shows 'Extensibility' selected. The main content area is titled 'Content Security Policy' and includes a 'Save and Publish' button. Under the 'Default' section, there is a checkbox for 'Disable content security policy'. Below this, a 'Required property' section lists several CSP directives: 'child-src', 'connect-src', 'font-src', 'img-src', 'media-src', 'object-src', 'script-src', and 'style-src'. Each directive has an '+ Add' button. The 'script-src' directive is highlighted with a red box, and its input field contains the URL 'https://www.example.com/scripts/example.js'. A trash icon is visible to the right of the input field. The bottom right corner of the page shows the time '11:27 AM' and the date '10/4/2019'.

#### 4. Select Save and publish.

## Interpret and fix CSP errors

When you first configure CSP for a site, some pages probably won't be loaded at all or won't work as intended, because CSP is blocking external connections, scripts, fonts, and other types of resources from being loaded. Fortunately, CSP logs some helpful errors that you can use to fix, tune, and clean up unwanted or unneeded requests.

The following illustration shows an example of CSP errors in a web browser's developer tools.

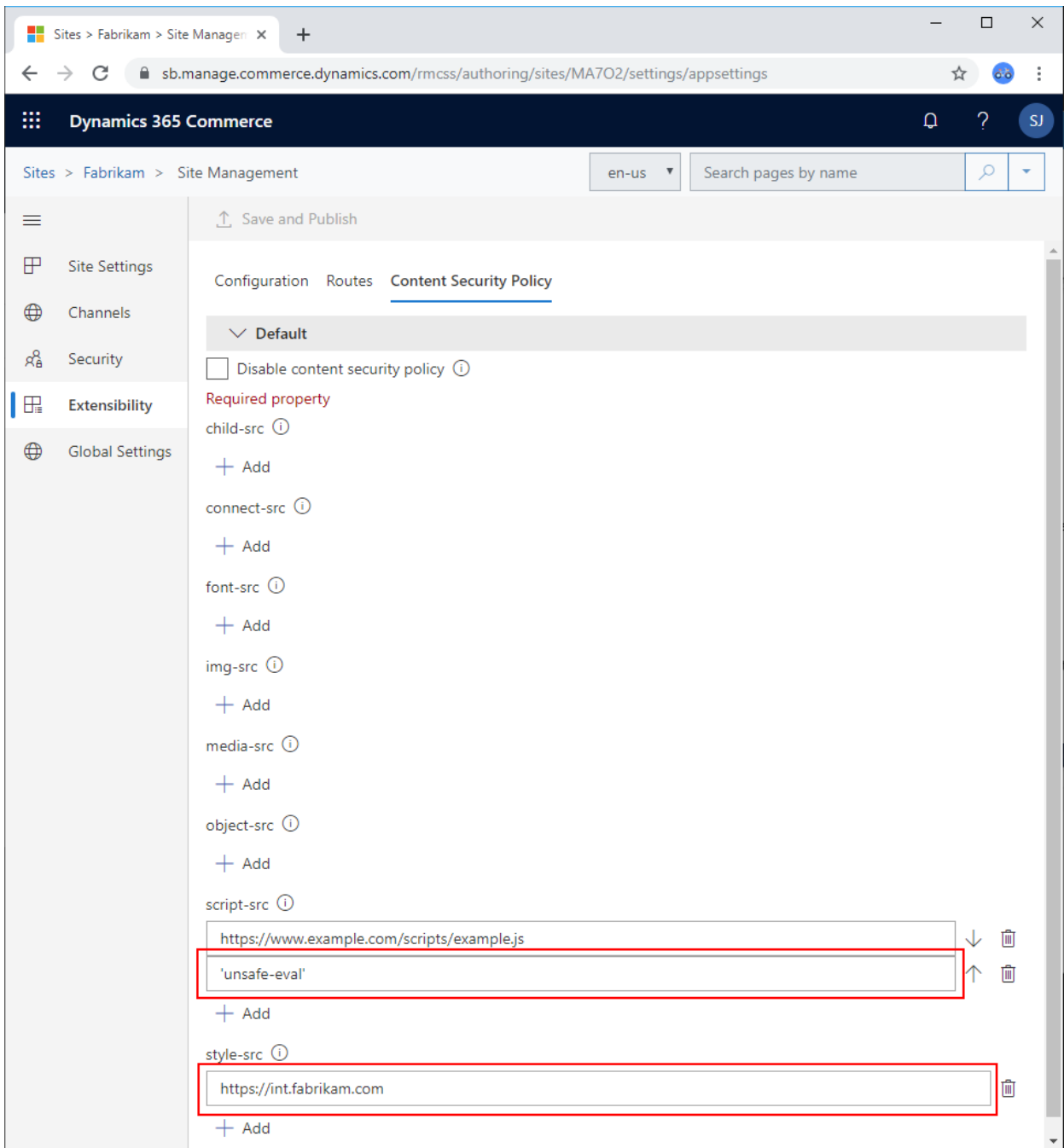


```
Uncaught EvalError: Refused to evaluate a string as JavaScript because 'unsafe-eval' is not an allowed source of script in the following Content Security Policy directive: "script-src 'self' 'unsafe-  
https://*.vo.msecnd.net/scripts/ https://dc.services.visualstudio.com/ https://checkoutshopper-test.adyen.com".  
at eval (anonymous)  
at jquery-bundle-1.10.2-ce=001-0008dc=RV1:3  
at Function.globalEval (jquery-bundle-1.10.2-ce=001-0008dc=RV1:3)  
at init.domManip (jquery-bundle-1.10.2-ce=001-0008dc=RV1:3)  
at init.prepend (jquery-bundle-1.10.2-ce=001-0008dc=RV1:3)  
at Array.<anonymous> (authorize?o=b2c-1-silent-ver=5.3.0.0:16)  
at Array.<anonymous> (jquery-bundle-1.10.2-ce=001-0008dc=RV1:3)  
at a (jquery-bundle-1.10.2-ce=001-0008dc=RV1:3)  
at Object.fireWith [as resolveWith] (jquery-bundle-1.10.2-ce=001-0008dc=RV1:3)  
at Object.<anonymous> (jquery-bundle-1.10.2-ce=001-0008dc=RV1:3)  
Refused to load the stylesheet 'https://int.fabrikem.com/_msdyn365/scrn/53e9550d-6ed3-4e6c-9603-7fefa76731fd/static/css/fabrikem/fabrikem.min.css' because it violates the following Content Security  
'unsafe-inline' https://*.commerce.dynamics.com https://checkoutshopper-test.adyen.com'. Note that 'style-src-elem' was not explicitly set, so 'style-src' is used as a fallback.
```

There are two CSP errors in this example:

- The **Eval** function is blocked by default, because it can cause arbitrary JavaScript execution. To allow this function, add **'unsafe-eval'** to your site's **script-src** directive. The single quotation marks are required.
- The external stylesheet is blocked. To allow a stylesheet to be loaded from an external domain, add the URL to your site's **style-src** directive.

The following screenshot shows what the fixed settings look like on the **Content Security Policy** tab in Commerce.



## Update page mocks that use CSP

If you're testing modules by using the online SDK in a development environment, you can also add CSP by using page mocks. In a page mock, you must either add a top-level **"appContext"** property or go to the existing top-level **"appContext"** property, and create a property under it that is named **"contentSecurityPolicy"**. There, you can add key/value pairs of directives to policies, as shown in the following example.

```
"appContext": {
  "contentSecurityPolicy": {
    "script-src": ["https://www.w3schools.com/js/myScript.js"],
    "font-src": ["https://*.commerce.dynamics.com"]
  }
}
```



**NOTE**

If you add CSP policies in a page mock, the page mock won't include any of the default CSP policies that are provided by the platform.

You can turn off CSP in a page mock by using the following code.

```
"appContext": {  
  "contentSecurityPolicy": {  
    "disableContentSecurityPolicy": true  
  }  
}
```

## Additional resources

[Manage e-Commerce users and roles](#)

[Add script code to site pages to support telemetry](#)

[Search engine optimization \(SEO\) considerations for your site](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Opt out of web activity event collection

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how you can let customers opt out of web activity event collection in Microsoft Dynamics 365 Commerce.

## Overview

Dynamics 365 Commerce lets site administrators analyze the web activity of users of their e-commerce sites. In that way, they can better understand how their sites are used, and they can optimize the sites to provide an improved user experience and meet business objectives.

## Ways for administrators to implement an opt-out experience

Administrators have three ways to implement an opt-out experience.

### Opt out on behalf of users

In Account management in Commerce headquarters (HQ), administrators can opt out on behalf of users.

1. In the HQ client, on the **All customers** page, search for and select a customer.
2. On the customer details page, on the **Retail** FastTab, in the **Privacy** section, set the **Do not track web activity** option to **Yes**.

Retail			
RETAIL	RECEIPT	LOYALTY ENROLLMENT	PRIVACY
Amount charged, not posted 0.00	Receipt option Standard receipt	Block customer for loyalty enrollment <input type="radio"/> No	Disable personalization <input type="radio"/> No
Post as shipment <input type="radio"/> No	Receipt email _____		Do not track web activity <input type="radio"/> No

3. Select **Save**, and then close the page.

### Module-based opt-out experience

Administrators can let authenticated users opt out of web activity event collection by themselves. To provide this opt-out experience, add the user opt-out module to customer account profile pages.

### Custom extensions

Administrators can create their own extensions to manage the opt-out experience for users. For more information, see [Call Retail Server APIs](#) and [Online channel extensibility](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Ways to add content

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview and links to documentation about how to manage content using the Microsoft Dynamics 365 Commerce site builder web authoring toolset.

## Overview

There are many ways to change the look, feel, and content of your site. Depending on the required level of customization, many of these changes can be implemented by non-developers within site builder, the web authoring toolset included with Dynamics 365 Commerce. Site builder enables you to build templates, select themes, and select and configure modules without writing any code. By contrast, development skills are required to create a new theme or module, because the e-Commerce software development kit (SDK) and the Microsoft Dynamics Lifecycle Services (LCS) deployment workflow must be used.

The following topics are good jumping off points to start understanding how to add and manage site content. Most of the topics listed focus on areas of your site that don't require a developer. Some address basic content editing, while others focus on site administrator tasks. Each of these topics will denote specific tasks might require SDK work. Each topic assumes that you have already provisioned a site and been granted access to the site builder toolset for your site.

Select one of the following topics to get started.

- To familiarize yourself with the content management terminology used in site builder and within this documentation, see [Page model glossary](#).
- To understand how modules work within content management workflows, see [Work with modules](#).
- To change the text, images, or video on an existing site page, see [Work with modules](#).
- To see how fragments can make content management more efficient and flexible, see [Work with fragments](#).
- To help ensure a successful on-brand authoring experience for web content authors, see [Templates and layouts overview](#) and [Work with templates](#).
- To rearrange sections on a site page, see [Work with layouts](#).
- To change the fonts, colors, and general look of site pages, see [Select a site theme](#) or [Work with CSS override files](#).
- To rearrange or add new navigation options, see [Customize site navigation](#).
- To learn how to stage, preview, and publish a broad set of concurrent web content changes, see [Work with publish groups](#).

## Additional resources

[Authoring page overview](#)

[Page model glossary](#)

[Document states and lifecycle](#)

[Enable and use cross-channel sharing](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Page model glossary

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes the various elements that are used on the pages of a Microsoft Dynamics 365 Commerce site.

## Page element definitions

The following table provides a summary of terms that you should be familiar with when you change the look, feel, and content of your site. For more thorough explanations and procedures, follow the links.

TERM	DESCRIPTION AND NOTES
<a href="#">Module</a>	<p><b>Definition:</b> Modules are building block that can be authored and make up the skeleton of a webpage. Examples include header, hero, and carousel modules.</p> <p><b>Where it's selected:</b> Deployed modules can be selected and configured in various stages of the site authoring workflow, such as the template, layout, page, and fragment authoring stages.</p> <p><b>Where it's edited:</b> Custom modules are created in code by using the software development kit (SDK). They are then uploaded to your site, where they become available for selection.</p>
Module property	<p><b>Definition:</b> Module properties are specific settings that are defined by the module. They can be edited in the e-Commerce authoring tools. For example, module properties are used to set the heading and background image of a banner module.</p> <p><b>Where it's configured:</b> Module properties are selected and configured in the property pane that appears in the authoring environments (editors) for templates, layouts, pages, fragments, and app settings.</p>
<a href="#">Template</a>	<p><b>Definition:</b> Templates define the module combinations and options that should be used for a category of pages (for example, marketing pages, category pages, and product pages).</p> <p><b>Where it's selected:</b> Templates can be selected during page or layout creation workflows.</p> <p><b>Where it's edited:</b> Templates are authored in the template editor. No code is required to create or modify them.</p>

TERM	DESCRIPTION AND NOTES
<p>Layout</p>	<p><b>Definition:</b> Layouts define the final selection and arrangement of modules from the parent template's set of options. A layout can be configured for a single page (<i>custom layout</i>), or it can be shared by multiple pages (<i>preset layout</i>).</p> <p><b>Where it's selected:</b> Layouts can be selected during new page creation or when a different layout is required for an existing page.</p> <p><b>Where it's edited:</b> Layouts are authored in the layout editor. No code is required to create or modify them.</p>
<p>Page instance</p>	<p><b>Definition:</b> Page instances define the final, page-specific localized content for a single page. This content is derived from the values of module properties.</p> <p><b>Where it's selected:</b> Pages are selected when URLs are assigned.</p> <p><b>Where it's edited:</b> Pages are edited in the page editor. No code is required to create or modify them.</p>
<p>Theme</p>	<p><b>Definition:</b> Themes define the Cascading Style Sheet (CSS), and determine the look and feel of the modules that are rendered on a page.</p> <p><b>Where it's selected:</b> After a theme is uploaded to your site by using Microsoft Dynamics Lifecycle Services (LCS), it can be selected as a property of the page container module.</p> <p><b>Where it's edited:</b> Themes are currently created and edited by using the SDK. They are then uploaded to your site by using LCS.</p>
<p>Fragment</p>	<p><b>Definition:</b> Fragments are fully configured modules that have localized content that can be reused and centrally updated across multiple pages. For example, a fragment that is created from a header module can be used in all templates and on all pages across your site, and centrally updated in one place.</p> <p><b>Where it's selected:</b> Fragments can be selected wherever modules can be selected. They can be substituted for a module to help increase efficiency through reusable and centralized authoring.</p> <p><b>Where it's edited:</b> Fragments are edited in the fragment editor. No code is required to create or modify them.</p>

TERM	DESCRIPTION AND NOTES
URL	<p><b>Definition:</b> Uniform resource locators (URLs) are addresses that point to webpages or other URLs.</p> <p><b>Where it's selected:</b> URLs are selected if links between pages are required.</p> <p><b>Where it's edited:</b> URLs are edited in the URL editor. No code is required to create or modify them.</p>
Asset	<p><b>Definition:</b> Assets are file binaries that have an extension such as .jpg, .docx, .pdf, or .mpg.</p> <p><b>Where it's selected:</b> Assets are selected as module properties for modules that require them.</p> <p><b>Where it's edited:</b> Assets are uploaded, and associated metadata is edited in the asset manager.</p>

## Additional resources

[Ways to add content](#)

[Document states and lifecycle](#)

[Work with publish groups](#)

[Enable and use cross-channel sharing](#)

[Work with modules](#)

[Work with fragments](#)

[Templates and layouts overview](#)

[Customize site navigation](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Document states and lifecycle

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the various document states of page elements in Microsoft Dynamics 365 Commerce.

## Document state descriptions

The [Page elements](#) topic lists various documents types in the content management system (CMS). These document types can have several states in the authoring tool. The document states help prevent data conflicts and enforce version control. They determine who can change the documents, when the documents can be changed, and when changes can be viewed by other people.

The following table shows the possible document states of page elements in Commerce.

DOCUMENT STATE	SITE BUILDER ACTION	DESCRIPTION
Checked out	Select <b>Edit</b> .	The applicable document is checked out to you. While a document is in this state, it can't be changed by other authenticated system users, and any changes that you make to the document are visible only to you.
Saved	Select <b>Save</b> .	Changes that have been made to a checked-out document are saved to the database, but the document isn't yet checked in or published. The saved changes aren't visible to other authenticated system users until the author selects <b>Finish editing</b> . They aren't visible to external users until the item is published.
Discarded check out	Select <b>Discard edits</b> .	All changes to the checked-out document are discarded, and the item reverts to the last version that was checked in.
Checked in	Select <b>Finish editing</b> .	The edited document is checked in. All changes are visible to other authenticated system users, and those users can then edit the document. Each check-in creates a document version record in the item's history.
Published	Select <b>Publish</b> .	The document is published, and the changes are pushed to your live site and become discoverable by external users. Items can be published only if they have first been checked in by selecting <b>Finish editing</b> .

## Additional resources



[Ways to add content](#)

[Page model glossary](#)

[Work with publish groups](#)

[Enable and use cross-channel sharing](#)

[Work with modules](#)

[Work with fragments](#)

[Templates and layouts overview](#)

[Customize site navigation](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Work with publish groups

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes the publish groups feature in Microsoft Dynamics 365 Commerce.

## Overview

E-commerce websites are constantly updated with new content throughout the year. Updates are often published in batches around busy e-commerce events such as holidays, seasonal marketing campaigns, or promotional launches. These updates often require that groups of website content (for examples, pages, images, fragments, and templates) be staged, validated, and published concurrently in a single action.

The ability to group items into logical sets that publish items together, where each set has its own lifecycle, provides many advantages to site authors. In Commerce, these logical sets are known as publish groups. They let site authors track sets of updates as their own configurable, testable, and publishable entities.

Authors can preview updates in a staged publish group without affecting the live site or other self-contained publish groups. Authors can then schedule the publish action to simultaneously publish all the items in the publish group to the live site. The ability to group, preview, and schedule updates for publishing is important for many enterprise-level companies that generate considerable annual revenue around event-based site update milestones.

Companies can incur costs from slow or invalidated content rollouts that don't go smoothly. Publish groups help guarantee that launches are organized, validated, and published on time. Whether they are large or small, publish groups provide a valuable toolset that helps authors organize and simplify ongoing site update tasks.

## When to use publish groups

You can use publish groups whenever you must stage and publish multiple documents together. For example, if your website updates content every season, you can create publish groups for these seasonal marketing motions. Your "Autumn Seasonal Update" publish group might contain new seasonal images, fragments that have seasonal marketing messages, pages that include seasonal product collections, or other seasonal website updates.

An advantage of publish groups is that you can stage multiple updates in parallel. For example, soon after the update for the "Autumn Seasonal Update" publish group, there might be a content update for a specific holiday weekend. In this case, you can stage content for the "Autumn Seasonal Update" publish group at the same time that you stage content for a subsequent "Autumn Holiday Update" publish group. Each publish group contains its own unique set of pages, images, fragments, templates, and so on. You can stage, preview, and validate these two publish groups independently but on a concurrent timeline. Each publish group can then be scheduled to go live on your site at specific dates and times.

## Turn on the publish groups feature

The publish groups feature is optional and must be turned for your site.

To turn on the publish groups feature for your site in the Commerce authoring tools, follow these steps.

1. In the left navigation pane, select **Site Settings** to expand it.
2. Under **Site Settings**, select **Features**.
3. Set the **Publish groups** option to **On**.

# Create a publish group

To create a publish group for your site in the Commerce authoring tools, follow these steps.

1. In the left navigation pane, select **Publish Groups**.
2. In the top command bar, select **New**.
3. In the **Create Publish Group** dialog box, under **Publish Group Name**, enter a descriptive name. Then select **OK**.

## Set the publish group authoring context

In Commerce, the default authoring context is the live site context. The live site authoring context is the default view where you can view and make changes directly to your website without using a publish group. It represents the latest direct updates to the published version of your site. If the context control under **Publish Groups** in the left navigation pane shows the name **Live site**, you're working in the live site authoring context. **Live site** is the default name of the context control. Otherwise, the context control shows the name of a publish group.

To work in a publish group, you must switch to the publish group authoring context for it. To set the publish group context, follow one of these steps.

- In the left navigation pane, select the context control directly under **Publish Groups**, and then select the name of the publish group in the list of options that appears. The context control is renamed and shows the name of the publish group.
- In the left navigation pane, select **Publish Groups**, and then, under **Publish Groups**, select the name of the publish group. The context control is renamed and shows the name of the publish group.

After you set your publish group authoring context, you're working in that publish group context when you preview and edit site content.

To return to the default live site authoring context, select the context control, and then select **Live site**.

## Add pages or other items to a publish group

After you've selected a publish group authoring context, and the context control in the left navigation pane shows its name, you can create content just as you do in the default live site context. You can also add existing pages or other items from other publish groups, or from the live site context.

To copy existing pages to a publish group, follow these steps.

1. Select the authoring context to copy from, and then, in the left navigation pane, select **Pages**.
2. Select the page to add to a publish group.
3. In the command bar, select **Copy to Publish group**.
4. In the **Select a Publish Group** dialog box, select the publish group to add the page to, and then select **OK**.

You can use the same basic steps to create customized product pages, URLs, templates, layouts, fragments, and media library assets, or to add existing items of these types to a publish group.

## Validate a publish group

To make sure that all dependencies in publish group content are satisfied, and that all validations are passed, you can run validation to identify any issues that must be addressed before you schedule a publish action.

To validate your publish group before you schedule it, follow these steps.

1. In the left navigation pane, select **Publish Groups**.

2. Select the publish group to validate.
3. In the command bar, select **Validate**.

Validation is run on all content in the publish group. Any issues that will prevent a successful publish action are shown in a notification box that appears in the upper right.

#### **NOTE**

Validation is always run automatically when a publish group is scheduled. However, the **Validate** button in the command bar is useful because it helps identify issues that you must fix before you try to schedule a publish group to go live.

## Schedule a publish group to go live

To schedule a publish group to go live on your site, follow these steps.

1. In the left navigation pane, select **Publish Groups**.
2. Under **Publish Groups**, select the publish group to schedule.
3. In the command bar, select **Edit Schedule**. The **Edit Schedule** dialog box appears.
4. Select the date and time when your publish group should go live, and then select **OK**.

To unschedule a publish group, follow the same steps, but select **Unschedule publish group** in the **Edit Schedule** dialog box.

#### **NOTE**

Very large publish groups might take up to a minute or two to be published when their scheduled time arrives. Be aware that a publish action isn't instantaneous, and that smaller publish groups will be published faster.

## Publish groups FAQ

### How many items can be in a publish group?

Currently, there is a limit of 2,000 items per publish group. Be aware that very large publish groups might take several minutes to be published when their scheduled time arrives.

### Are publish groups like code "branches" in software development terminology?

Yes and no. Publish groups can be thought of as forked versions of your site. In that way, they do act like branches. However, there is no concept of a merge at the level of individual items. The item that is published last just overwrites what previously existed, and the most recent publish action always supersedes previous publish actions.

### Can I schedule two publish groups to go live at the same time?

No. For performance and conflict reasons, the system will force you to stagger scheduled publish groups at least five minutes apart.

### Can I use publish groups to schedule omnichannel back-office items, such as discounts and product updates?

Currently, the publish groups feature supports only website content. However, Microsoft is aware that integration with back-office merchandising scenarios could be valuable for the coordination and automation of omnichannel campaign launches.

## Additional resources

Ways to add content

Page model glossary

Document states and lifecycle

Enable and use cross-channel sharing

Work with modules

Work with fragments

Templates and layouts overview

Customize site navigation

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable and use cross-channel sharing

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to enable and use the cross-channel sharing feature of Microsoft Dynamics 365 Commerce site builder.

## Overview

Cross-channel sharing lets retailers reuse and share content among multiple channels of a site. This capability is useful when the site channels have a compatible base language, or when they have numerous content items in common.

Cross-channel sharing works by enabling a default channel that will be searched for available content when a channel-specific version of the requested content isn't found. Content that is intended to be shared among channels is created in the default channel. That content can be localized for any locale that is used on any site channel.

## When to use cross-channel sharing

Cross-channel sharing is useful when multiple channels on a single site can share content. For example, a retailer that has multiple brands and storefronts that are grouped under a single site can share some content among some or all of the storefronts. This shared content can include pages for terms and conditions, payment terms, shipment methods, and frequently asked questions (FAQ).

Cross-channel sharing also supports fragments. Therefore, a content page that contains channel-specific fragments can be created as cross-channel content. In this case, although most of the content will be shared among channels, channel-specific fragments on a cross-channel page will be rendered only when they are requested from the corresponding storefront channel.

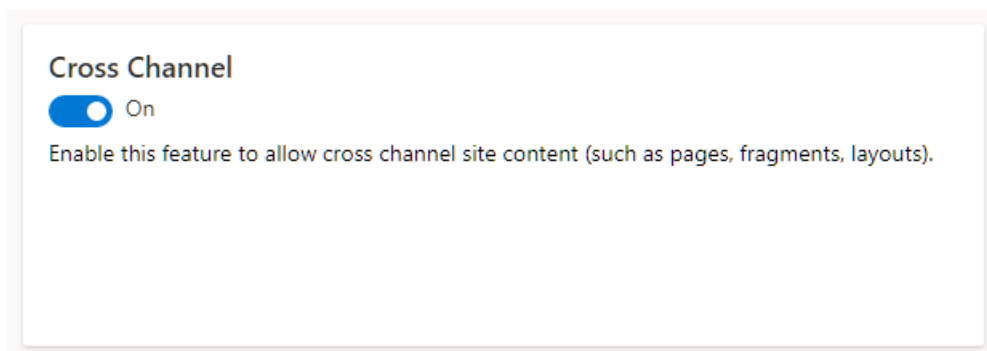
Sites that have only one channel, or sites that have multiple channels that can't share content, won't benefit from cross-channel sharing.

## Enable cross-channel sharing

Cross-channel sharing is enabled at the site level. This operation is one-way. In other words, after cross-channel sharing is enabled, it can't be disabled.

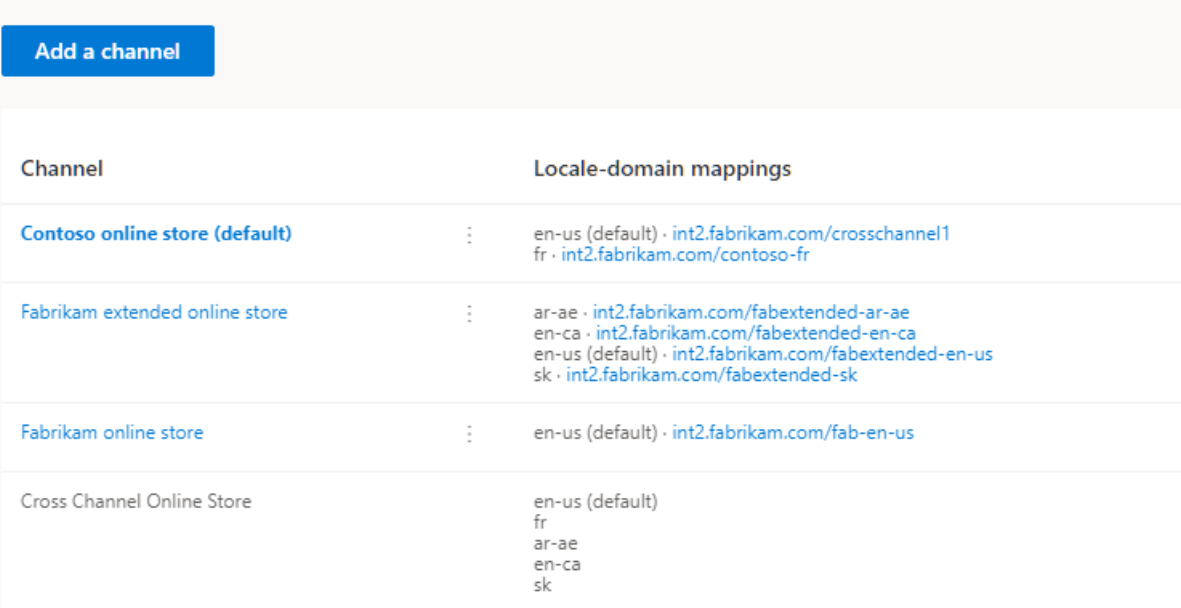
To enable cross-channel sharing in Commerce site builder, follow these steps.

1. Go to **Site settings > Features**.
2. Set the option for the **Cross Channel** feature to **On**.



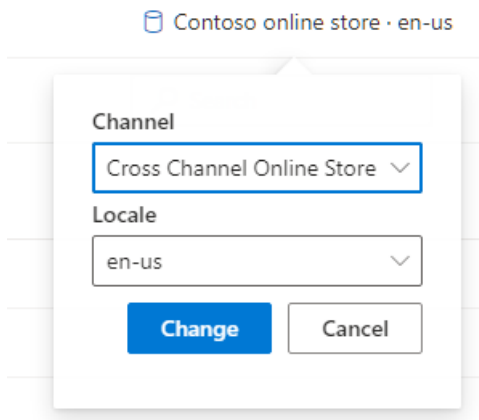
After you enable cross-channel sharing, cross-channel information will appear in the **Channels** section at **Site settings > Features**, as the example in the following illustration shows.

## Channels



Channel	Locale-domain mappings
Contoso online store (default)	en-us (default) · <a href="https://int2.fabrikam.com/crosschannel1">int2.fabrikam.com/crosschannel1</a> fr · <a href="https://int2.fabrikam.com/contoso-fr">int2.fabrikam.com/contoso-fr</a>
Fabrikam extended online store	ar-ae · <a href="https://int2.fabrikam.com/fabextended-ar-ae">int2.fabrikam.com/fabextended-ar-ae</a> en-ca · <a href="https://int2.fabrikam.com/fabextended-en-ca">int2.fabrikam.com/fabextended-en-ca</a> en-us (default) · <a href="https://int2.fabrikam.com/fabextended-en-us">int2.fabrikam.com/fabextended-en-us</a> sk · <a href="https://int2.fabrikam.com/fabextended-sk">int2.fabrikam.com/fabextended-sk</a>
Fabrikam online store	en-us (default) · <a href="https://int2.fabrikam.com/fab-en-us">int2.fabrikam.com/fab-en-us</a>
Cross Channel Online Store	en-us (default) fr ar-ae en-ca sk

Additionally, after you enable cross-channel sharing, the **Channel** field in the upper right of Commerce site builder will include a **Cross Channel Online Store** option that you can use to manage cross-channel content, as shown in the following illustration.



## Create and use cross-channel content

You can create and use cross-channel content in multiple ways. For example, you can create cross-channel fragments, create cross-channel pages that use cross-channel and channel-specific content, and override cross-channel fragments with channel-specific versions of fragments.

### Create a cross-channel fragment

To create a cross-channel fragment in Commerce site builder, follow these steps.

1. Go to **Fragments**, and select **New** to create a new fragment.
2. In the **New fragment** dialog box, select the **Promo banner** module, and then, under **Fragment name**, enter a name (for example, **Cross-channel banner**). Then select **OK**.
3. In the property pane for the **Promo banner** module, select **Add Message**, and then select **Message**.
4. In the **Message** dialog box, under **Text**, enter **Cross-channel**, and then select **OK**.

5. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

This cross-channel fragment can be used on cross-channel or channel-specific pages that are created on any site channel.

### Create a cross-channel page that uses cross-channel content

Cross-channel pages can be used on any channel of your site. Therefore, you can create a shared content page one time and make any subsequent updates in a single place. For example, a cross-channel **Terms and conditions** page that has the URL `/toc` can be shared among all the channels of a site. If the base URLs for the site channels are `www.fabrikam.com/brand1` and `www.fabrikam.com/brand2`, the same cross-channel, shared **Terms and conditions** page will be available from both site channel URLs, at `www.fabrikam.com/brand1/toc` and `www.fabrikam.com/brand2/toc`, respectively. If the **Terms and conditions** page must be updated later, you have to update only the single, shared page.

To create a cross-channel page in Commerce site builder that uses cross-channel content, follow these steps.

1. Go to **Pages**, and select **New** to create a new page.
2. In the **Choose a template** dialog box, select a template, such as **Marketing**.
3. Under **Page name**, enter a name for the page (for example, **Cross-channel page**).
4. Under **Page URL**, enter a page URL (for example, **examplepage**), and then select **OK**.
5. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add fragment**.
6. In the **Add fragment** dialog box, select the cross-channel fragment that you created earlier that has a promo banner, and then select **OK**.
7. Select **Save**, and then select **Preview** to preview the page. You should see the promo banner that says, "Cross-channel."
8. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

### Create a channel-specific page that uses cross-channel content

By using cross-channel content on channel-specific pages, you can create a shared content fragment one time and then use it on channel-specific pages. This "single sourcing" is useful for shared content such as terms and conditions, payment terms, or contact information.

To create a channel-specific page in Commerce site builder that uses cross-channel content, follow these steps.

1. From within a specific channel, such as **Fabrikam extended online store**, go to **Pages**, and then select **New** to create a new page.
2. In the **Choose a template** dialog box, select a template, such as **Marketing**.
3. Under **Page name**, enter a name for the page (for example, **Channel-specific page**).
4. Under **Page URL**, enter a page URL (for example, **channelspecificpage**), and then select **OK**.
5. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add fragment**.
6. In the **Add fragment** dialog box, under **Channel**, select **Cross Channel Online Store**. The cross-channel fragment that you created earlier should appear in the list. Select it, and then select **OK**.
7. Select **Save**, and then select **Preview** to preview the page. You should see the promo banner that says, "Cross-channel."
8. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

### Create a channel-specific version of a cross-channel page

Cross-channel sharing supports overrides of cross-channel content. For example, all but one of your site channels share the same piece of content. That one site channel requires different content. To implement the different content for it, you override the cross-channel content with channel-specific content by creating a channel-specific version of the cross-channel page.

To create a channel-specific version of a cross-channel page in Commerce site builder, follow these steps.



1. In the **Channel** field in the upper right, select **Cross Channel Online Store**.
2. Open the cross-channel page that you created earlier.
3. In the **Channel** field in the upper right, select the channel that should have specific content. The page editor shows a message that prompts you to create a new page variant.
4. Select **Create page variant**.
5. In the **Main** slot of the page variant, select the ellipsis (...), and then select **Add Module**.
6. In the **Add Module** dialog box, select the **Promo banner** module, and then select **OK**.
7. In the property pane for the **Promo banner** module, select **Add Message**, and then select **Message**.
8. In the **Message** dialog box, under **Text**, enter **Channel-specific**, and then select **OK**.
9. Select **Save**, and then select **Preview** to preview the page. You should see the promo banner that says, "Channel-specific."
10. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

Now, if you use the base URL of the channel and go to the URL of the cross-channel page on that site, you will see the channel-specific content instead of the cross-channel content.

## Additional resources

[Ways to add content](#)

[Page model glossary](#)

[Document states and lifecycle](#)

[Work with publish groups](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add a logo

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add a logo to your site in Microsoft Dynamics 365 Commerce.

## Overview

When you build your site, one of the first things that you will probably do is add your company or brand logo to the site's header. The Dynamics 365 Commerce online module library provides a module that makes this task easy.

You can add a logo directly to a template, layout, or page. In this way, you can easily change the logo that appears on specific pages or groups of pages. However, this topic covers the most frequent scenario, where you add your logo to a header fragment that can be reused across all the pages of your site.

## Prerequisites

Before you can add a logo to all the pages of your site, you must complete these tasks.

1. Upload your logo to the Media Library.
2. Create a header fragment. For more information about how to create and use fragments, see [Work with fragments](#).
3. Include the header fragment in the template that the pages of your site use for their layout and module options. For more information about templates, see [Work with templates](#).

## Add a logo to a header fragment

To add a logo to the header fragment for your site, follow these steps.

1. In the navigation pane on the left, select **Fragments**.
2. Select the header fragment that you created, and then select **Edit**.
3. Expand the header module.
4. In the property pane for the header module, provide an image and link for the logo.
5. Save the header fragment, finish editing it, and publish it.

After you publish the updated header fragment, all site pages that use the template that contains the header fragment will show your logo.

## Additional resources

[Select a site theme](#)

[Work with CSS override files](#)

[Add a favicon](#)

[Add a welcome message](#)

[Add a copyright notice](#)

[Add languages to your site](#)

[Add script code to site pages to support telemetry](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Select a site theme

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to set or change your site's theme in Microsoft Dynamics 365 Commerce.

## Overview

A site's layout and style (for example, fonts, sizes, and colors) are defined by the theme that you select and apply to the site. A theme is created and deployed by a developer at your company. For an overview of themes, see [Theming overview](#). For more information about how to create and deploy themes, see [Create a new theme](#).

By default, when you first create a site, it uses a theme that is named **Fabrikam**. This default theme is provided as part of the Commerce module library. After you've deployed additional themes for your site, you can configure the site so that it uses one of them instead.

## Select the site theme

To select the theme that is applied to your site, follow these steps.

1. In the site authoring environment, go to your site.
2. Go to **Site Management > Extensibility**.
3. Under **Theme**, select a theme on the drop-down menu.
4. To apply the selected theme to your site, select **Save and publish**.

### NOTE

The theme that you select is published to your site as soon as you select **Save and publish** on the **Extensibility** page. To preview a theme on your site before you apply it, you can use your development or sandbox environment.

## Additional resources

[Add a logo](#)

[Work with CSS override files](#)

[Add a favicon](#)

[Add a welcome message](#)

[Add a copyright notice](#)

[Add languages to your site](#)

[Add script code to site pages to support telemetry](#)

[Theming overview](#)

[Create a new theme](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Work with CSS override files

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes why, when, and how to use Cascading Style Sheets (CSS) override files in Microsoft Dynamics 365 Commerce.

## Overview

Permanent site styles should usually be handled through a site's theme. Themes provide the foundational CSS and style settings for the modules on any page of your site. Themes are created by using the Dynamics 365 Commerce online software development kit (SDK), and they are deployed to your websites through Microsoft Dynamics Lifecycle Services (LCS). Theme debugging capabilities and module interface configurations in the SDK help site developers create customizable and cohesive site design packages. When these design packages are deployed to a site, site authors can focus on creating, editing, and publishing content instead of site development.

Given the usual lifecycle of a theme, the dependency on developers to make style changes through the SDK and LCS deployment pipeline can be prohibitive in some scenarios. Site prototypes or hotfixes can seem cumbersome if the SDK isn't configured, or if you don't have time to wait for an LCS deployment.

In these scenarios, CSS override files can help. In the Commerce authoring tools, authenticated users can upload a CSS file, preview it, and then activate it to override a site's theme. The overhead of SDK or LCS deployment isn't required. The overrides that are specified in a CSS override file can be as small as a change to a single text style or as wide-ranging as a complete brand overhaul.

Before you use CSS override files, be aware of the following limitations:

- Only one CSS override file can be active on a site at a time. Therefore, all active overrides must be present in a single override file.
- Although you can preview the overrides in the Commerce authoring tools, there are no dedicated debugging features to help identify any bugs that the overrides introduce. In other words, when you use CSS override files, you don't have the same toolset that the SDK provides for module and authoring validation.

Nevertheless, CSS override files provide a quick way to prototype a design or implement a hotfix before a full theme update is developed and deployed.

## Create a CSS override file

To create a CSS override file, you can use any integrated development environment (IDE), text editor, or source code editor. A typical approach is to use standard web debuggers in your browser to identify style selectors, properties, and values on your existing site. Most browsers let you change values and preview them in the debugger. After you've identified the changes that you want to make, you can save them to your own CSS file.

## Upload a CSS override file

To upload a CSS file to your site in Commerce, follow these steps.

1. In the authoring tools, go to your site.
2. In the navigation pane on the left, select **Design**.

#### NOTE

Depending on the version of the Commerce authoring tools that you're using, you might have to expand **Settings** in the navigation pane before you can select **Design**.

3. At the top of the main design pane, select the **CSS override** tab, if it isn't already selected.
4. Under **Available CSS overrides**, select **Upload CSS file**. A File Explorer window appears.
5. In File Explorer, browse to and select a CSS file, and then select **Open**. The uploaded CSS file now appears under **Available CSS overrides**.

## Preview a CSS override file

To preview a CSS override file before you make it active on your live site, follow these steps.

1. In the navigation pane on the left, select **Design**, and then, on the **CSS override** tab, under **Available CSS overrides**, find the CSS file that you want to preview.
2. Next to the CSS file name, select **Preview site**.
3. In the **Select a URL** dialog box, select the URL of the site that you want to see the override applied to, and then select **OK**.
4. If there are multiple variants for the selected URL, select the desired variant in the **Preview variations** dialog box that appears.

A new browser tab or window is opened, where you can validate your style overrides against your site. You can then share the URL with other authenticated Commerce users for review and feedback.

## Activate a CSS override file on your live site

After you've previewed and approved the CSS override file, you can activate it on your live site.

#### NOTE

Only one CSS override file can be active on your site at a time. If you activate a new override file, the previous override file is inactivated. Therefore, make sure that all required overrides are present in a single CSS override file.

To activate a CSS override file, follow these steps.

1. In the navigation pane on the left, select **Design**, and then, on the **CSS override** tab, under **Available CSS overrides**, find the CSS file that you want to activate.
2. Under the CSS file name, select **Activate**. The override file immediately becomes active on your live site.

## Deactivate a CSS override file on your live site

To deactivate a CSS override file on your site, follow these steps.

1. In the navigation pane on the left, select **Design**, and then, on the **CSS override** tab, under **Available CSS overrides**, find the CSS file that you want to deactivate.
2. Under the CSS file name, select **Deactivate**. The override file immediately becomes inactive on your live site.

**TIP**

To access additional options for CSS override files, select the ellipsis (...) next to the CSS file name. The **Download**, **Rename**, and **Replace** options are useful for quick changes to an existing CSS override file.

## Additional resources

[Add a logo](#)

[Select a site theme](#)

[Work with style presets](#)

[Add a favicon](#)

[Add a welcome message](#)

[Add a copyright notice](#)

[Add languages to your site](#)

[Add script code to site pages to support telemetry](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Work with style presets

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to work with site style presets in Microsoft Dynamics 365 Commerce site builder.

## Overview

A style preset is a stored set of all authorable style values across a site's theme. It can be used to immediately change the look of a site from site builder. Style presets let Commerce site builder authors quickly change, preview, and activate a set of style values across their site, without having to use Cascading Style Sheets (CSS) or deploy themes. Font styles, button styles, and site colors are typical examples of style variables that can be managed through style presets.

The set of style variables that is available in a site is determined by the theme and module library that is deployed to a site's tenant. The Dynamics 365 Commerce online software development kit (SDK) lets developers implement as many (or as few) authorable style variables as they require for a given theme. By enabling more style variables, a theme developer can put final choices about site styles into the hands of site builder authors. Therefore, non-developers can update and preview site styles by using the toolset. The capability is also useful for any scenario where direct changes to themes or CSS will cause unnecessary overhead.

Themes where authorable style variables are enabled require a default style preset. They can optionally include additional preset options as part of a deployed theme package. For example, a theme can be deployed that has a single default "modern light" style preset. Alternatively, it might include additional style preset options besides the default preset, such as "modern dark," "vintage light," or "vintage dark." These built-in theme presets are created by developers and can be used as starting points for new site designs.

In site builder, authors can select among a theme's built-in presets, or they can create their own style presets and customizations by using the enabled style variables. A style preset can be previewed in site builder before it's activated on the live site. After an author's style changes are reviewed, the style preset can then be set to "active" for the live site.

## Preview a style preset

To preview a style preset on your site in site builder, follow these steps.

1. In the left navigation pane for your site, go to **Site Settings > Design**.
2. On the **Style presets** tab at the top of the design editor, in the **Available presets** list, select a preset, and then select **View** to go to the preset editor.

If there are currently no presets in the **Available presets** list, see [Create a custom style preset](#) for information about how to create a custom style preset.

### NOTE

Presets that were included with the theme are indicated by a **Built-in** badge. These built-in presets are read-only. To copy a built-in preset as a new customizable preset, select the ellipsis button (...) for the preset, and then select **Save as**.

3. On the command bar, select **Preview**.
4. Select a URL from your site to use to preview the style preset, and then select **OK**.

5. Select the channel-specific and locale-specific URL variant to preview by selecting the variant's name. A new preview browser window is opened, where the selected style preset is applied to the specified page.

#### NOTE

The preview URL is persistent and authenticated. Therefore, you can copy, paste, and send it to other authenticated co-workers for review before you set it to "active" on your live site. The preview URL is also useful for checking styles on different devices, in different browsers, and on different screens.

#### TIP

While you edit a style preset, you might find it helpful to leave the preview browser window for it open in a separate browser window, so that you can quickly validate your changes. After you save your changes to a preset, refresh the open preview browser window to validate the user experience.

## Create a custom style preset

To create a custom style preset in site builder, follow these steps.

1. In the left navigation pane for your site, go to **Site Settings > Design**.
2. On the **Style presets** tab at the top of the design editor, on the command bar, select **New preset**.
3. Enter a name and description for the new preset, and then select **Save**. A new customizable preset is created that uses the theme's default values as a starting point.

#### NOTE

You can also create a new custom style preset from any existing preset by selecting the ellipsis button (...) for that preset and then selecting **Save as**. Alternatively, select **Save as** on the command bar in the preset editor.

## Modify global and module type style values

Some of a theme's style variables are shared among multiple module types. These style variables are referred to as *global* style variables. Examples include primary site colors, default font styles, and button styles. By setting global variables, you might change the look across many different module types.

Some style values can be unique to a module type, or they might have to optionally override a default global value. If a site's theme has implemented module type style variables, site builder authors can customize the style of a module type independently of the global settings. Module type variables can either augment or override the global style variables in a theme.

#### NOTE

The hierarchy of style values in a site behaves in the following manner. Style values that appear farther to the right override the style values to the left of them.

Theme default values < Global style values < Module type style values < CSS override

To change a style preset's global or module type values in site builder, follow these steps.

1. In the left navigation pane for your site, go to **Site Settings > Design**.
2. On the **Style presets** tab at the top of the design editor, select **View** for any style preset to go to the preset editor.

3. Select **Preview**, and then follow the URL selection steps to open a full-browser-window preview for your preset. Leave this preview browser window open.
4. In the preset editor, select **Edit** in the upper right.
5. Use the style variable controls in the editor to change some global style values.
6. At the top of the editor, on the **Modules** tab to the right of the **Global** tab, select a module type that must be styled.
7. Use the style controls to change some values for the module type.
8. When you're ready to preview your changes, select **Save** on the command bar.
9. Return to the open preview browser window, and refresh it. The full-browser-window preview is useful for checking style changes at different view breakpoints, in different browsers, and on different device platforms.
10. When all changes have been completed and validated, select **Finish editing** in the upper right of the editor.

#### NOTE

If you're editing the style preset that is currently active on your site, you will see a blue **Active** badge in the editor. This badge indicates that the preset is currently live on your website. If you change the active preset, select **Publish** to push those changes to your live site.

## Make a new style preset active on your live site

To set a style preset as the new active preset on your site, follow these steps.

- Select the **Set as active button** in one of these places:
  - The command bar in the style preset editor
  - The ellipsis menu (...) for any available preset in the main view on the **Style presets** tab at **Site Settings > Design**

The preset's style values are made active across your public-facing website.

## Additional resources

[Add a logo](#)

[Select a site theme](#)

[Work with CSS override files](#)

[Add a favicon](#)

[Add a welcome message](#)

[Add a copyright notice](#)

[Add languages to your site](#)

[Add script code to site pages to support telemetry](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add a favicon

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to add a favicon to your site.

## Overview

A favicon is a small graphics file that is shown on a web browser tab, in the Address bar, in the browsing history, and in bookmarks or favorites, among other places. We recommend that you add a favicon to your site, because it represents and reinforces your brand, and helps distinguish your site from other sites that your customers visit.

Although you can add multiple favicons of various sizes and file types to your site, this topic shows how to add a single favicon. However, the same process and location are used to add more favicons.

## Upload a favicon to your site's asset collection

To upload a favicon to your site's asset collection, follow these steps.

1. In the left navigation pane, select **Media Library**.
2. On the command bar, select **Upload > Upload Media Items**.
3. In the File Explorer window, browse to the favicon image file that you want to upload, select it, and then select **Open**.
4. In the **Upload Media Item** dialog box, enter the required title and alt text.
5. If you want to publish the image immediately after upload, select the **Publish media items after upload** check box.

### NOTE

If you don't select the **Publish media items after upload** check box, you must return to **Media items** page and manually publish the favicon later.

6. Select **OK**.
7. In the property pane on the right, copy the public URL of the favicon. You will use this URL later.

## Create the HTML for your favicon

To create the HTML for the favicon, use the following HTML string. For the **href** attribute, replace **Public\_URL\_for\_your\_favicon** with the public URL that you copied earlier.

```
<link rel="shortcut icon" href="Public_URL_for_your_favicon">
```

## Create a fragment that contains a metatag for your favicon

To create a fragment that contains a metatag for your favicon, follow these steps.

1. Go to **Fragments**, and select **New**.
2. In the **New fragment** dialog box, select **Metatags** as the module that the fragment is based on.

3. Enter a name for the fragment, and then select **OK**.
4. In the fragment hierarchy tree, select the **Default metatags** child.
5. In the right pane, under **Meta Tags**, select **Add**, and then enter the HTML string that you created earlier for the favicon.
6. Select **Finish editing**, and then select **Publish** to publish the fragment.

## Add the metatag fragment to the HTML head section of your pages

To add the metatag fragment to the HTML **head** section of your pages, follow these steps.

1. Go to **Templates**, open the template for the pages that you want to add your favicon to, and then select **Edit**.
2. In the template hierarchy tree, select the ellipsis (...) button to the right of the **HTML head** container, and then select **Add fragment**.
3. In the **Select fragment** dialog box, select the metatag fragment that you created earlier, and then select **OK**.
4. Select **Finish editing**, and then select **Publish** to publish the template.

### NOTE

If your site uses more than one template, you must add the metatags fragment to all of them.

When you preview pages that are based on the template that you added the metatags fragment to, you should now see the favicon on the browser tab.

## Additional resources

[Add a logo](#)

[Select a site theme](#)

[Work with CSS override files](#)

[Add a welcome message](#)

[Add a copyright notice](#)

[Add languages to your site](#)

[Add script code to site pages to support telemetry](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add a welcome message

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add a welcome message to your Microsoft Dynamics 365 Commerce website.

## Overview

A welcome message on your e-Commerce website can inform visitors about ongoing sales, site updates, or availability of seasonal collections. The welcome message is set by using the alert module.

The alert module should be added to the **Error/Information messages** slot of the header fragment. The alert module lets you specify the text that is shown, the text color, and the alignment. It also lets you specify whether visitors to the site can dismiss the message.

When a welcome message is added to a shared header fragment, it will be shown on every page that uses the template where that shared header fragment is used.

To add a welcome message to your site, follow these steps.

1. In Commerce site builder, go to your site.
2. Select **Fragments**.
3. Select the header fragment to add the message to.
4. In the outline tree, expand **Error/Information messages**.
5. Select the alert module, and then select **OK**. If an alert module doesn't yet exist, first select the ellipsis button (...) next to **Error/Information messages**, and then select **Add module**.
6. In the property pane on the right, on the **Data** tab, select **Add Data Source**, and then select **Content**.
7. In the **Input Text** field, enter the text of the welcome message.
8. Select **Save**, select **Finish editing** to check in the header fragment, and then select **Publish** to publish it.

The welcome message will now appear at the top of every site page that uses the selected header fragment.

## Additional resources

[Add a logo](#)

[Select a site theme](#)

[Work with CSS override files](#)

[Add a favicon](#)

[Add a copyright notice](#)

[Add languages to your site](#)

[Add script code to site pages to support telemetry](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add a copyright notice

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add a copyright notice to your e-Commerce website.

## Prerequisites

Before you can add a copyright notice to your site, you must have the following items:

- A template that includes a shared footer fragment.
- A page that uses that template.

## Add a copyright notice

To add a copyright notice to the bottom of every page that uses a specific template, follow these steps.

1. Go to **Fragments**, and then select **New**.
2. In the **New fragment** dialog box, select the **Footer** module, and name the fragment. For example, enter **Footer-Copyright**.
3. Select **OK**.
4. In the navigation pane, select the ellipsis button (...) next to **Footer**, and then select **Add Module**.
5. In the dialog box, select **Footer category**, and then select **OK**.
6. In the navigation pane, select the ellipsis button next to **Footer category**, and then select **Add Module**.
7. In the dialog box, select **Text block**, and then select **OK**.
8. In the navigation pane, select **Text block**.
9. In the properties pane on the right, in the **Paragraph** field, add your copyright message. For example, enter **(C) Fabrikam 2019**.
10. Select **Save**, select **Finish editing**, and then select **Publish**.
11. Go to **Templates**, select the template, and then select **Edit**.
12. Under **Page Outline**, expand **Body**, and then expand **Default Page**.
13. Select the ellipsis button next to **Footer Slot**, and then select **Add Fragment**.
14. Select the fragment that you created earlier, and then select **Select**.
15. Select **Finish editing** to check in the template, and then select **Publish** to publish it.

The footer that contains the copyright notice automatically appears at the bottom of all pages that use the selected template.

## Additional resources

[Add a logo](#)

[Select a site theme](#)

[Work with CSS override files](#)

[Add a favicon](#)

[Add a welcome message](#)

[Add languages to your site](#)

## Add script code to site pages to support telemetry

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Add languages to your site

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to add support for additional languages to a Microsoft Dynamics 365 Commerce site.

## Overview

You can localize your website into any language that Commerce supports. (The list of supported languages appears later in this topic.) To add a language on your website, you must first add it to an online store that is bound to your site.

## Add a language to an online store

To add a language to an online store, follow these steps.

1. Open the Dynamics 365 Commerce environment for your site.
2. Go to **Retail and Commerce > Channels > Online stores** to access the list of online stores that are configured for your environment. Alternatively, enter **online stores** as a search term.
3. Select the online store to add a language for.
4. On the **Languages** FastTab, select **Add**.
5. In the **Language** field, select the language to add.

The language that you added will now be available so that you can configure your site to use it in the site authoring environment.

### Languages that are supported by Dynamics 365 Commerce

- af
- ar
- ar-ae
- ar-bh
- ar-dz
- ar-eg
- ar-iq
- ar-jo
- ar-kw
- ar-lb
- ar-ly
- ar-ma
- ar-om
- ar-qa
- ar-sa
- ar-sy
- ar-tn
- ar-ye
- be
- bg
- ca

- cs
- da
- de
- de-at
- de-ch
- de-li
- de-lu
- el
- en-029
- en-au
- en-bz
- en-ca
- en-gb
- en-ie
- en-in
- en-jm
- en-my
- en-nz
- en-sg
- en-tt
- en-us
- en-za
- es
- es-ar
- es-bo
- es-cl
- es-co
- es-cr
- es-do
- es-ec
- es-gt
- es-hn
- es-mx
- es-ni
- es-pa
- es-pe
- es-pr
- es-py
- es-sv
- es-tr
- es-uy
- es-ve
- et
- eu
- fa
- fi

- fo
- fr
- fr-be
- fr-ca
- fr-ch
- fr-lu
- hi
- hr
- hu
- is
- it
- it-ch
- ja
- lt
- lv
- mk
- ms
- mt
- nb-no
- nl
- nl-be
- nn-no
- pl
- pt-br
- ro
- ru
- ru-ru
- sk
- sl
- sq
- sr
- sr-la
- sv
- sv-fi
- th
- tn
- tr
- uk
- ur
- xh
- zh-hans
- zh-hk
- zh-sg
- zu

Additional resources

[Add a logo](#)

[Select a site theme](#)

[Work with CSS override files](#)

[Add a favicon](#)

[Add a welcome message](#)

[Add a copyright notice](#)

[Add script code to site pages to support telemetry](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add script code to site pages to support telemetry

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to add client-side script code to your site pages to support the collection of client-side telemetry.

## Overview

Web analytics are an essential tool when you want to understand how your customers interact with your site and make decisions that will help optimize the experience for maximum conversion. Many web analytics packages are available to help you achieve these goals, such as Google Analytics, Clicky, Moz Analytics, and KISSMetrics. Most web analytics packages require that you add client-side script code in the `<head>` element of the HTML for all pages of your site.

### NOTE

The instructions in this topic also apply to other custom client-side functionality that Microsoft Dynamics 365 Commerce doesn't natively offer.

## Create a reusable fragment for your script code

A fragment allows you to reuse inline or external script code across all pages on your site, regardless of the template they use.

### Create a reusable fragment for your inline script code

To create a reusable fragment for your inline script code in site builder, follow these steps.

1. Go to **Fragments**, and then select **New**.
2. In the **New fragment** dialog box, select **Inline script**.
3. Under **Fragment name**, enter a name for the fragment, and then select **OK**.
4. Under the fragment that you created, select the **Default inline script** module.
5. In the property pane on the right, under **Inline script**, enter your client-side script. Then configure other options as you require.
6. Select **Save**, and then select **Finish editing**.
7. Select **Publish**.

### Create a reusable fragment for your external script code

To create a reusable fragment for your external script code in site builder, follow these steps.

1. Go to **Fragments**, and then select **New**.
2. In the **New fragment** dialog box, select **External script**.
3. Under **Fragment name**, enter a name for the fragment, and then select **OK**.
4. Under the fragment that you created, select the **Default external script** module.
5. In the property pane on the right, under **Script source**, add an external or relative URL for the external script source. Then configure other options as you require.
6. Select **Save**, and then select **Finish editing**.
7. Select **Publish**.

#### NOTE

If content security policy (CSP) is enabled for your site, ensure that all external URLs are added to the `script-src` CSP directive in Commerce site builder. For more information, see [Manage Content Security Policy \(CSP\)](#).

## Add a fragment that includes script code to a template

To add a fragment that includes script code to a template in site builder, follow these steps.

1. Go to **Templates**, and open the template for the pages that you want to add your script code to.
2. In the left pane, expand the template hierarchy to show the **HTML Head** slot.
3. In the **HTML Head** slot, select the ellipsis button (...), and then select **Add fragment**.
4. Select the fragment that you created for your script code.
5. Select **Save**, and then select **Finish editing**.
6. Select **Publish**.

## Add an external script or inline script directly to a template

If you want to insert an inline or external script directly into a set of pages that are controlled by a single template, you don't have to create a fragment first.

### Add an inline script directly to a template

To add an inline script directly to a template in site builder, follow these steps.

1. Go to **Templates**, and open the template for the pages that you want to add your script code to.
2. In the left pane, expand the template hierarchy to show the **HTML Head** slot.
3. In the **HTML Head** slot, select the ellipsis button (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select **Inline script**.
5. In the property pane on the right, under **Inline script**, enter your client-side script. Then configure other options as you require.
6. Select **Save**, and then select **Finish editing**.
7. Select **Publish**.

### Add an external script directly to a template

To add an external script directly to a template in site builder, follow these steps.

1. Go to **Templates**, and open the template for the pages that you want to add your script code to.
2. In the left pane, expand the template hierarchy to show the **HTML Head** slot.
3. In the **HTML Head** slot, select the ellipsis button (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select **External script**.
5. In the property pane on the right, under **Script source**, add an external or relative URL for the external script source. Then configure other options as you require.
6. Select **Save**, and then select **Finish editing**.
7. Select **Publish**.

## Additional resources

[Add a logo](#)

[Select a site theme](#)

[Work with CSS override files](#)

[Add a favicon](#)

[Add a welcome message](#)

[Add a copyright notice](#)

[Add languages to your site](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Work with modules

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes how and when to use modules in Microsoft Dynamics 365 Commerce.

## Overview

Modules are logical building blocks that make up your page structure, and they have various purposes and scopes. Some modules are high-level containers, and their only purpose is to hold and organize other modules (child modules). Other modules, such as a simple image placement module, have a very specific purpose. Other modules, such as a carousel module, fall somewhere between those two categories.

By default, your Dynamics 365 Commerce site includes a module library that lets you achieve most basic e-Commerce scenarios. You should be able to construct an end-to-end e-Commerce site just by using these modules. However, you might also want to customize these modules or build new, custom modules for specific needs. If you want to build custom modules, a module design software development kit (SDK) is available to help you create a custom module library.

## Container modules and slots

As was mentioned earlier, some modules are designed to hold child modules. These modules are known as *containers*, and they allow for hierarchies of nested modules. Container modules include *slots*. Slots are used to handle the layout and purpose of child modules in the container. An example is a basic page container module (a top-level module for any page) that defines several important slots:

- A header slot
- A sub-header slot
- A main slot
- A footer slot
- A sub-footer slot

The module's developer defines these slots, and determines which child modules and how many child modules can be put directly inside it. For example, the header slot might support only one module of the **Header Module** type, whereas the body slot might support an unlimited number of modules of any type (except other page container modules).

In the authoring tools, page authors don't have to know in advance which modules can and can't be put in each slot. When page authors select a slot and then try to select a module to add to it, they see a filtered view of module types that are supported for that slot.

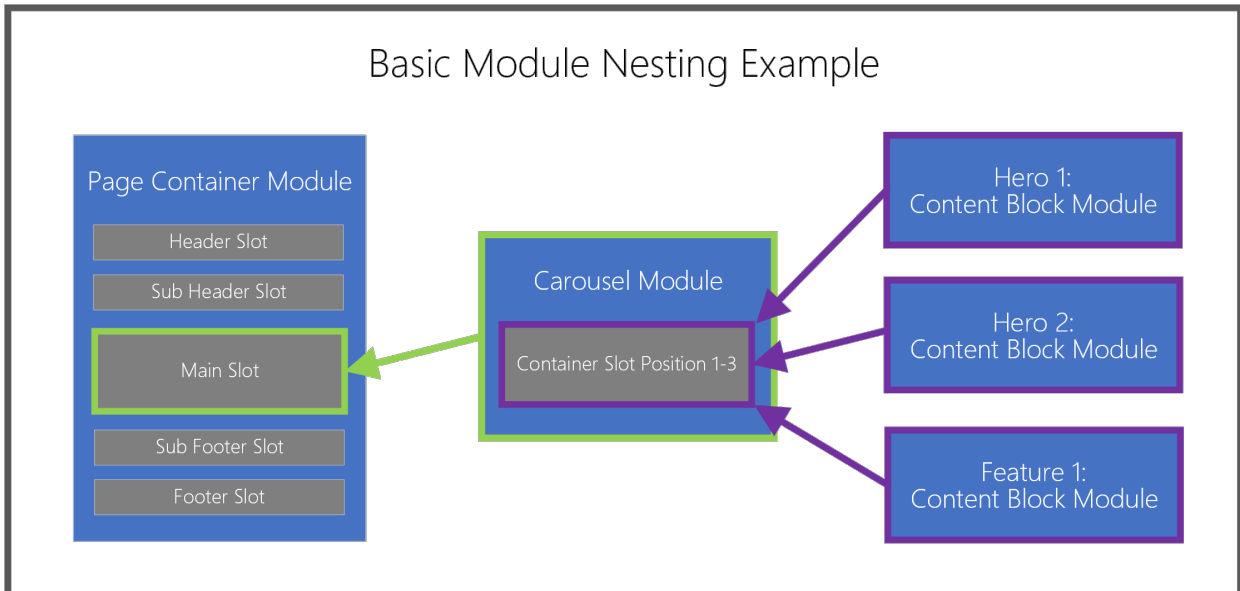
## Content modules

Content modules contain content and media elements, such as text (for example, headlines, paragraphs, and links) or asset references (for example, images, video, and PDFs). Typical content module types include content block, text block, and promo banner modules. Modules of these three types can contain text or media, and they don't require any child modules to make something visible on a page.

The majority of typical, day-to-day page and content authoring activities involve content modules, primarily because these modules define the actual content that is rendered in their parent container modules. Many content modules are available, and these modules are typically the last pieces that you will add to a page's hierarchy of nested modules.



The following illustration shows how modules are nested inside parent container module slots.



## Add or remove modules

The following procedures describe how to add and remove modules.

### Add a module

To add a module to a slot or container on a page, follow these steps.

1. In the outline pane on the left or directly in the main canvas, select a container or slot to which a child module can be added.

#### NOTE

The module designer defines the list of modules types that can be added to a specific module slot. Template authors can then refine the allowed module options to help guarantee consistent search engine optimization (SEO) and authoring efficiency for all the pages that are built from a specific template. When adding a module to a slot, the **Add Module** dialog box is automatically filtered so that it shows only modules that are supported in the selected container or slot. This list of allowed modules is determined by the page's template or the container's module definition.

2. If using the outline pane, select the ellipsis (...) next to the module name, and then select **Add Module**. If using the controls directly within the canvas, select the plus symbol (+) in an empty slot or adjacent to the currently selected module, and then select **Add Module**.

#### NOTE

If a container or slot doesn't support new child modules, the **Add Module** option is unavailable.

3. In the **Add Module** dialog box, select a module to add to your page.

#### TIP

**Content block** is a good module type for beginners to work with.

4. Select **OK** to add the selected module to the selected container or slot on your page.

### Remove a module

To remove a module from a slot or container on a page, follow these steps.

1. In the outline pane on the left, select the ellipsis (...) next to the name of the module to be removed, and then select the trash can symbol. Alternately, in the main canvas you can select the trash can symbol on a selected module's toolbar.
2. When prompted to confirm that you want to remove the module, select **OK**.

## Move a module to a new position

To move a module to a new position within your page, use any of the following methods.

### Move a module using the outline pane

To move a module using the outline pane, follow these steps.

1. Select and hold the module you want to move in the outline pane, then drag the module to a new position in the outline. The blue line in the outline and on the canvas indicates where the module can be placed.
2. Release the module to drop it into the new position.

### Move a module directly within the canvas

To move a module directly within the canvas, follow these steps.

1. Select the module you want to move in the canvas.
2. Select either an upward or downward pointing arrow symbol in the module's toolbar, and then drag the arrow to a new position on the page. The blue line in the canvas and outline indicates where the module can be placed. If a module cannot be moved up or down, that arrow symbol will be grayed out.
3. Release the module to drop it into the new position.

### Move a module using the ellipsis menu

To move a module using the ellipsis menu, follow these steps.

1. Select a module in either the outline or the canvas.
2. Select the ellipsis (...) next to the module's name in the outline pane, or in the module's toolbar in the canvas.
3. If the module can be moved up or down within the container or slot, you will see options for **Move up** or **Move down**. Select the desired move option to move the module up or down relative to its siblings.

## Configure modules

The following procedures describe how to configure content and container modules.

### Configure a content module

To configure a content module on a page, follow these steps.

1. In the outline pane on the left, expand the tree and select any content module (for example, **Content block**). Alternately, you can select the module in the main canvas.
2. In the module properties pane on the right, enter properties for any desired module controls.
3. On the command bar, select **Save**. This will also refresh the preview canvas.

### Edit module text properties

Module text properties that are not read-only can be edited directly in the canvas.

To edit module text properties, follow these steps.

1. Select the text control in the canvas, then place your cursor where you wish to edit text.
2. Enter your text content.
3. Select anywhere outside the text content to continue editing other content.

## Inline image selection

Module images that are not read-only can be changed directly from the canvas.

To choose a new image for a content module, follow these steps.

1. In the canvas, double-click the image. This will bring up the media picker window.
2. Find and select a new image you want to use, and then select **OK**. The new image is now rendered in the canvas.

## Configure a container module

To configure a container module on a page, follow these steps.

1. Select a container module on your page (for example, a carousel or fluid container module).
2. In the properties pane on the right, expand the nested controls by selecting the headers, and set any required control values.
3. In the outline pane on the left, select the ellipsis button next to the name of either the container or any slots inside the container, and then select **Add Module**. Then, add child modules to the selected container. For more information, see the [Work with modules](#) section earlier in this topic.
4. If multiple child modules exist as siblings in a parent container, you can change their display order in the parent container. Select the ellipsis button for a module, and then use the up arrow and down arrow buttons.

## Additional resources

[Templates and layouts overview](#)

[Work with templates](#)

[Work with preset layouts](#)

[Work with fragments](#)

[Add a container module to a page](#)

[Work with publish groups](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Work with fragments

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes why, when, and how to use fragments in Microsoft Dynamics 365 Commerce.

## Overview

Fragments allow for a centralized authoring experience for module configurations that must be reused throughout your site. For example, headers, footers, and banners are often configured as fragments, because they are shared across many pages. You can think of fragments as miniature webpages that can be inserted into other pages on your site. Fragments have their own lifecycle. In other words, they are created, referenced, updated, and deleted as independent entities in the authoring tools.

After fragments are configured, they can be used wherever modules can be used in your site structure. Fragments can be referenced on pages, in layouts, in templates, and in other fragments.

### NOTE

Fragments can be nested up to seven levels deep inside other fragments.

For example, if you want to promote a seasonal event cross many pages on our site, you can use a fragment. The first step in the process of creating a new fragment is to select the type of module that you want to start from. For this example, you can build the fragment from a hero module.

### NOTE

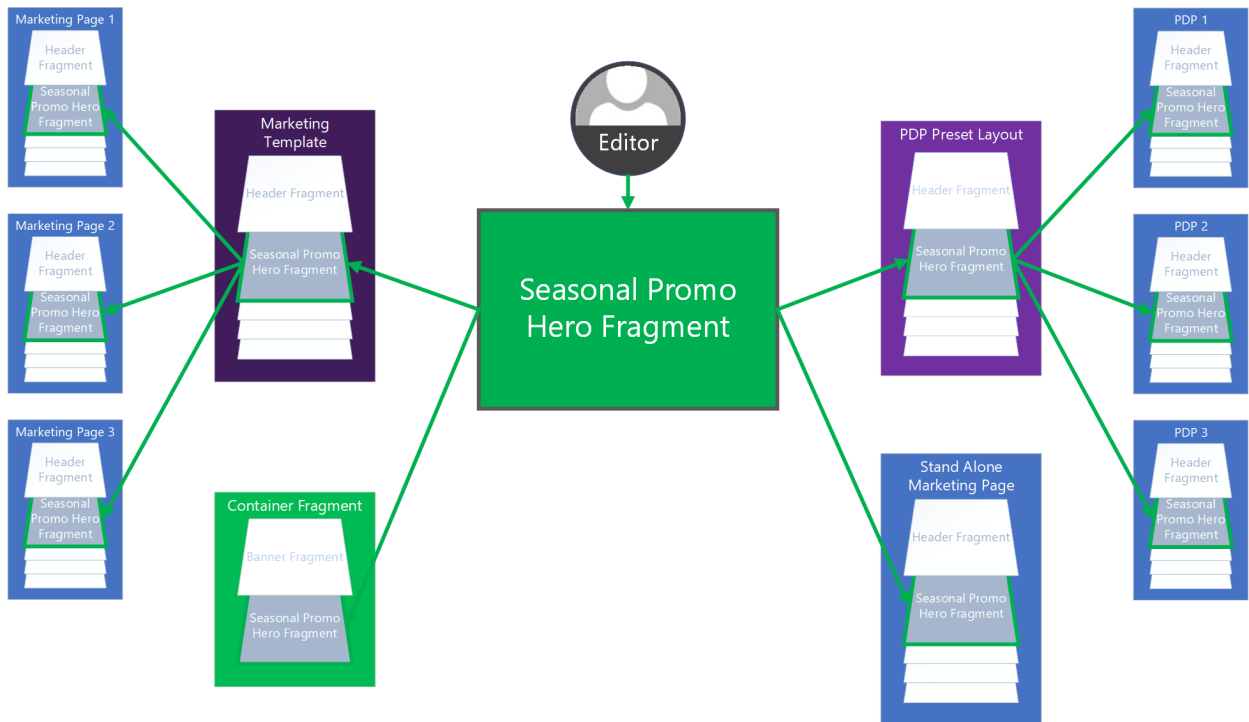
Fragments can be built from any module type.

You can then configure the hero fragment with your specific promotional content. You can also localize it as you require. The new stand-alone hero fragment can then be consumed as a preconfigured module throughout your site. You can easily add it to templates, to specific pages, or to other fragments that can contain hero modules.

All the places where the fragment is added are references to the central hero fragment that you created. If you publish changes to the fragment, those changes are immediately reflected in all the places where the fragment is referenced across the site. Therefore, fragments provide a powerful and efficient way to reuse and centrally manage module configurations on a site. By effectively using them, you can significantly increase agility and help reduce the cost that is associated with managing site content.

The following illustration shows how fragments can be used to centralize authoring of shared module configurations across an e-Commerce site.

Fragments enable centralized authoring for module configurations that are re-used across your site



## Create a fragment

You can either create a new fragment or save an existing module configuration as a fragment.

### Save an existing module configuration as a fragment

To convert a previously configured module to a reusable fragment in Commerce site builder, follow these steps.

1. Open a page or template that contains the module that you want to convert to a fragment.
2. In the outline pane on the left or directly in visual page builder, select the previously configured module.
3. Select the ellipsis (...) next to the name of the module in either the outline pane or the selected module's toolbar in visual page builder.
4. Select **Share as fragment**.
5. In the **Save as fragment** dialog box, enter a name for the fragment.
6. Select **OK** to save the module configuration as a fragment that can be added to other pages.

### Create a new fragment

To create a new fragment in Commerce site builder, follow these steps.

1. In the navigation pane on the left, select **Fragments**.
2. Select **New**. A **New fragment** dialog box appears that shows all the available module types. As was mentioned earlier, fragments can be created from any module type.
3. Select a module type for your fragment.

#### TIP

By selecting a generic container module type, you get the most flexibility when you need to update and configure your fragment later.

## Add, remove, or edit fragments on a page

The following procedures describe how to add, remove, and edit fragments.

### Add a fragment

To add a fragment to a page in Commerce site builder, follow these steps.

1. In the outline pane on the left or directly in visual page builder, select a container or slot to which child modules can be added.
2. Select the ellipsis (...) next to the name of the container or slot. Alternately, if using visual page builder, select the plus symbol (+).
3. Select **Add fragment**.

#### NOTE

If the container or slot doesn't support new child modules, the **Add fragment** option is unavailable.

4. In the **Select fragment** dialog box, search for and select a fragment to add. If no available fragments are listed, you might first have to create a fragment from a module type that the selected container or slot supports.
5. Select your desired fragment to add it to the container or slot on your page.

#### NOTE

The modules that are allowed in a container or slot are defined by the page's template or the modules' own definitions.

### Remove a fragment

To remove a fragment from a slot or container on a page in Commerce site builder, follow these steps.

1. In the outline pane on the left, select the ellipsis (...) next to the name of the fragment to be removed, and then select the trash can symbol. Alternately, you can select the fragment in visual page builder and select the trash can symbol in the fragment's toolbar.
2. When you're prompted to confirm that you want to remove the fragment, select **OK**.

#### NOTE

When you remove a fragment from a page, you just remove the reference to it from that page. You do **not** delete the fragment from your site. To delete fragments from your site, you must use the fragment inspector user interface (UI). You can delete fragments from a site only if they aren't currently referenced by any pages, templates, or other fragments.

### Edit a fragment

To edit fragments, you must use the fragment editor UI. This restriction is by design. It helps guarantee that authors don't confuse the process of editing the modules for a specific page with the process of editing fragments that might be shared across many pages.

To edit a fragment in Commerce site builder, follow these steps.

1. In the navigation pane on the left, select **Fragments**.
2. Under **Fragments**, select the fragment to edit.
3. Edit the fragment's module properties and structure as you require. The process resembles the process for editing modules are edited in the page editor view.

You can also edit a fragment by selecting it on a page, in a template, or in a parent fragment, and then selecting **Edit Fragment** in the properties pane on the right.

# Additional resources

[Templates and layouts overview](#)

[Work with templates](#)

[Work with preset layouts](#)

[Work with publish groups](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

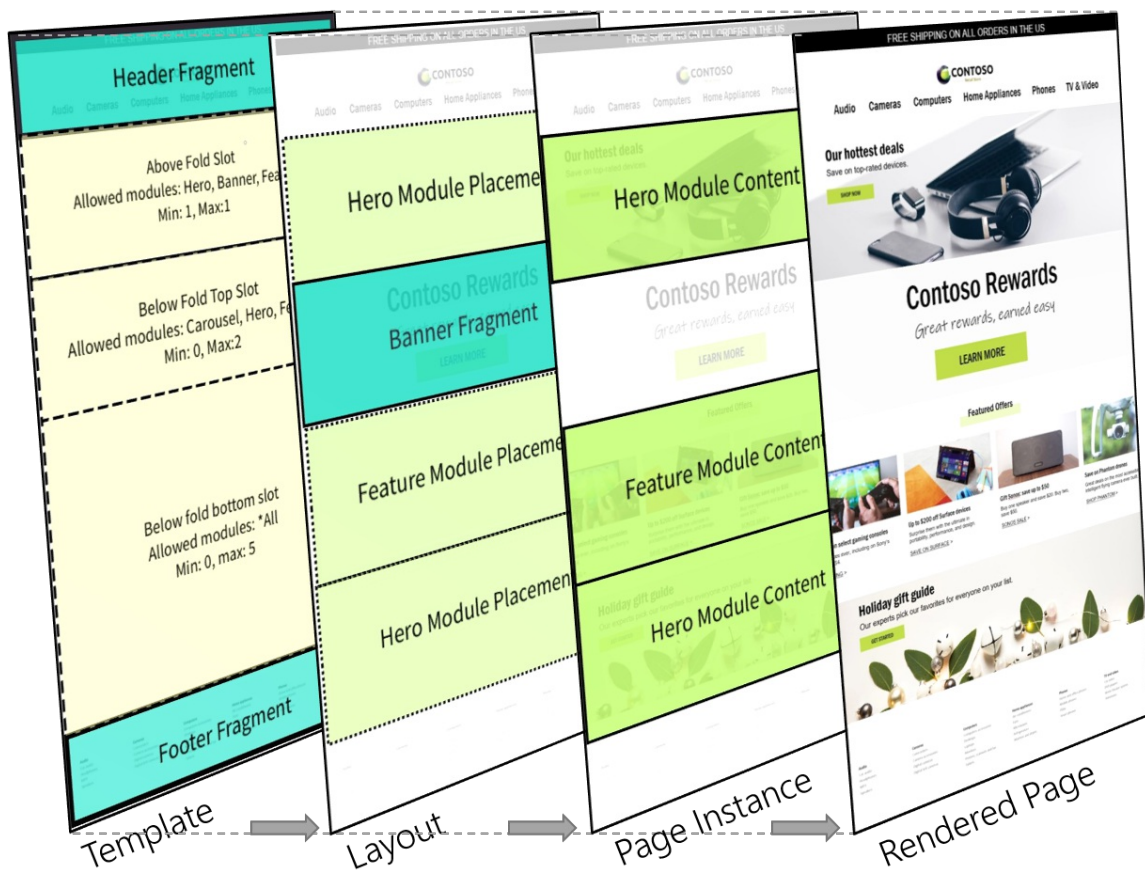
# Templates and layouts overview

2/18/2021 • 7 minutes to read • [Edit Online](#)

Templates are a foundational element of the Microsoft Dynamics 365 Commerce page model. If your goal is to maximize efficiency and consistency for site authoring workflows, it's important that you learn how to take advantages of templates for your website. Early decisions about template structure are important, and can significantly affect cost and agility for daily, seasonal, and site-wide brand updates. Well-structured templates have other benefits too. For example, they help improve site-wide search engine optimization (SEO) scores and minimize bug counts.

A good way to start to work with templates is to understand the functional benefits of templates and layouts, the differences between them, and the hierarchy.

The following illustration shows the page model hierarchy behind a rendered webpage.



ENTITY	BASIC FUNCTION
Template	Templates define the module options and basic scaffolding for a set of layouts and page instances.
Layout	Layouts define the final selection and arrangement of modules for a page or a set of pages.
Page instance	Page instances define the data and content for specific pages.



# Templates

Templates are at the top of the Dynamics 365 Commerce page model hierarchy and represent an important early step for site configuration. Conceptually, templates help control consistency across a family of child layouts and pages by defining the base structure and authoring options for downstream layout creation and page creation workflows. Templates can help simplify the content authoring process through predefined, centrally managed elements (such as headers and footers) and guided authoring flows that help guarantee that module configuration choices are on-brand.

## Controlling consistency

When you design a template, the biggest business decision that you must make is how much control the template should have over the page creation process. A template that leaves everything open for a downstream author is the easiest type of template to create, but it might have long-term consequences for the maintenance of pages that are created from it. A well-written template provides guidance and a streamlined authoring experience, but it also gives authors enough flexibility so that they can complete their task. All these aspects depend on the level of control that the template enforces.

Templates can help content authors be more efficient and stay on-brand in the following ways:

- Limit the modules that can be used on a page.
- Suggest default module and configuration choices.
- Explicitly make some module and configuration choices that are controlled at the template level. This process is also known as *locking a setting*.

The following example shows how a basic template (template X) can be configured:

- All child layouts of template X must have a header container, a body container, and a footer container.
- In template X, the configuration of the header container is locked and can be changed only in template X itself. All child layouts and pages always have this header.
- The body container requires at least one module and up to a maximum of ten modules. These modules are defined by downstream layouts and pages.
- For the body container, the hero, feature, carousel, and banner modules are available.
- A footer container is configured in template X, but it can be overridden by downstream layouts and pages.

The template in this example defines a simple structure and set of options for downstream content authors. Notice that some parts of a page (in this case, the header) are fully defined and locked in the template, and they can't be changed by downstream authors. Other parts (in this case, the body) can be defined by downstream authors within specific guidelines (in this case, a minimum number and maximum number of modules of specific types). And other parts (in this case, the footer) are defined in the template but can be overridden by downstream authors.

An important initial step for site and brand admins is to determine the correct balance between constraint and flexibility for child layout and page authors. When templates are used, this balance is completely configurable. It affects whether page elements are centrally updated (locked in the template) or left to individual child levels that are lower in the page hierarchy.

To start to use templates, [Work with templates](#).

## Layouts

Layouts are the next level in the page model hierarchy, below templates. Whereas a template defines all the modules that are allowed for a page, a layout is an explicit selection and arrangement of modules. Pages are the next level in the page model hierarchy, below layouts. They define the localized content for the modules that are selected in the layout.

The following example builds on the template example from the previous section, and shows how a basic layout can be configured:

- The parent template of the layout requires that the body container have between one and ten modules. These modules can be only hero, feature, carousel, and banner modules. Therefore, the layout can define the following selection and arrangement of modules:
  - The first module in the body container is a banner module, and it's followed by a hero module and two feature modules.
  - The first feature module is left-aligned, and the second feature module is right-aligned.
- Even though a default footer is inherited from the parent template, the template author left the footer unlocked. Therefore, the layout can override it by defining a different footer fragment.

The layout in this example defines the final arrangement of modules for child pages. Like a template, a layout can define default or locked module properties that will always be inherited by child pages (for example, the alignment of the feature modules). The actual content or data for every module in the layout is then defined farther down the hierarchy, in each child page instance. An important distinction here is that layouts don't directly contain localizable content, whereas their child pages do. The layout's primary function is to define the final arrangement and default configuration of modules for its child pages.

This hierarchy is powerful for two reasons. First, layouts that share the same parent template are treated as compatible for layout switching scenarios. Therefore, the layout for any page can be changed to another layout from the same template hierarchy without requiring that page-level content be reauthored. You can take advantage of this capability to do seasonal design updates, experiment, or do a permanent site redesign. Second, layouts provide another way to centrally modify shared elements for a group of pages without requiring updates to individual pages. For example, if a product category has 1,000 pages that share the same layout, the modules can be reordered in the layout, and this change will immediately be reflected in all 1,000 child pages.

By understanding this hierarchy, you can deliver an agile and efficient site structure that helps save cost, is scalable, and produces better results as the site evolves over time.

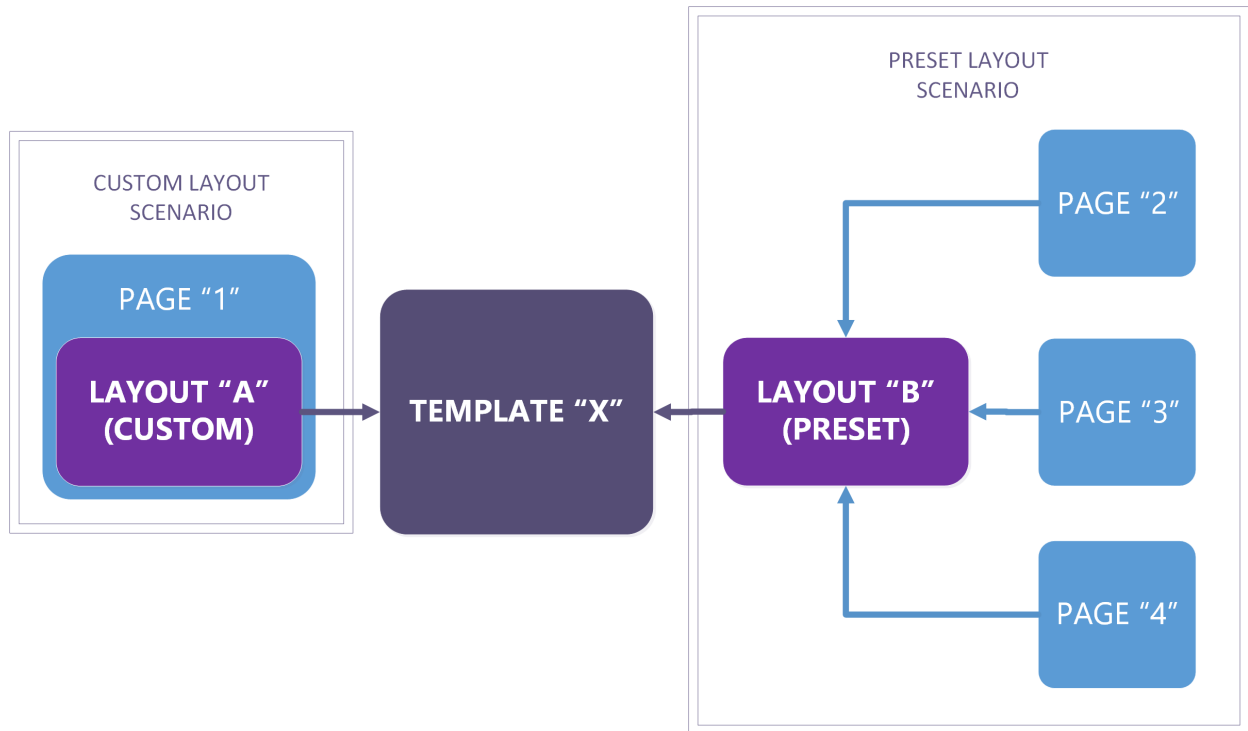
### **Preset and custom layouts**

Layouts on your site can be either *preset* or *custom*:

- **Preset layouts** allow for a page creation workflow where all modules are already selected and arranged, and only data entry is required. This approach can help save time when many pages must be authored that have the same layout requirements. Preset layouts have a one-to-many relationship with their child pages. Therefore, a single preset layout can be used to centrally control the module arrangement for hundreds or thousands of child pages.
- **Custom layouts** are essentially single-use layouts that are embedded in one page. They aren't exposed as an option when other new pages are created or in layout switching scenarios. The benefit of this approach is that an author can experiment by authoring a page that uses a custom layout. Then, if the author wants to reuse the layout for other pages, it can easily be converted to a preset layout. The new preset layout is then exposed as an option in page creation workflows and in layout switching scenarios for pages from the same template hierarchy. Conversely, preset layouts can be branched into custom layouts. In this way, an author can break a page away from the preset layout and create a new single-use custom layout. (This new custom layout is still bound by any constraints in the parent template.)

Preset layout and custom layouts are edited in different parts of the authoring toolset. Because custom layouts have no dependencies on other pages, they are edited directly in the page editor. In this case, the existence of a layout is mostly transparent to the user and is exposed only in page-level properties and through the actions for layout options. However, because changes to preset layouts can affect many child pages, they must be edited in the layout editor, where publish actions consider the full downstream impact on child pages.

The following illustrations shows scenarios for preset and custom layouts.



To start to use preset layouts, see [Work with preset layouts](#).

## Additional resources

[Work with templates](#)

[Work with preset layouts](#)

[Work with publish groups](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Work with templates

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes how to work with templates in Microsoft Dynamics 365 Commerce.

## Overview

As was discussed in [Templates and layouts overview](#), templates define the set of options that is available to downstream authors. Templates are useful to an enterprise's web authoring team for several reasons, and well-structured templates can help with all the following goals:

- Simplify the authoring experience for day-to-day content editor roles.
  - Filter module options so that only relevant modules are shown for a specific page section. (For example, a marketing section of a template can be configured to filter out irrelevant modules that should never be used in that context, and that will just complicate content authoring tasks if they are shown.)
  - Configure default module setting to help improve authoring efficiency.
  - Define default page fragments to help improve authoring efficiency. (For example, header and footer fragments in a template will automatically appear on every downstream page.)
- Keep enterprise sites on-brand by defining an approved set of module arrangement and configuration options.

### TIP

Successful e-Commerce sites provide customers with familiar, repeatable, and on-brand user experience (UX) design patterns. By using templates, you help control consistency across your site.

- Improve search engine optimization (SEO) scores by ensuring repeatable and programmatically defined page definitions and metadata.

### NOTE

Although templates are designed to control consistency across a site, they can theoretically be configured so that they don't enforce any consistency. Brand and site administrators can define any level of variability for the pages on their site. For example, a template can be left entirely open, so that content authors can create any page design that they choose. In this case, none of the benefits in the preceding list are applicable.

## Modify a template

Templates are modified by using the template editor.

To open the template editor, follow one of these steps:

- In the navigation pane of your site, select **Templates**, and then select the template to modify.
- In the page editor for an existing page, select the top node in the outline tree on the left. Then, in the property pane on the right, select **Edit Template**.

The outline tree view on the left shows the module options and structures that are available to child layouts and pages. When you select a module in the outline tree, you can view the template properties for the selected

module in the property pane on the right. Some of these properties are unique to template editing. The following table describes these properties.

PROPERTY NAME	DESCRIPTION
Min Occurs	This property defines the minimum number of occurrences for the selected module. For example, if the value is set to <b>1</b> , the module is required for downstream authors, whereas if the value is set to <b>0</b> (zero), the module is optional.
Max Occurs	This property defines the maximum number of occurrences for the selected module. For example, if the value is set to <b>1</b> , the module can be added only one time.
Min Modules (Containers)	For modules that contain other modules (that is, for <i>containers modules</i> ), this property defines the minimum number of total modules that should be added as children. For example, for a carousel module, the value might be set to a number that is more than 1.
Max Modules (Containers)	For container modules, this property defines the maximum number of total modules that should be added as children. For example, for a carousel module, the value might be set to a number that is less than 10.
Locked	A <b>Locked</b> Boolean control appears next to all core module properties. It lets the template author lock a module setting in the template. A module setting that is locked can't be overridden by any child layouts or pages. It becomes a centrally editable property value for all layouts and pages that use the template.

## Create a new template

To create a new template, follow these steps.

1. In the navigation pane of your site, select **Templates** to open the template inspector view.
2. Select **New Template**.
3. In the template creation dialog box, enter a name and description for the template. The values that you enter will be shown to authors when they create new pages. Therefore, enter metadata that will be useful to page authors. For example, enter **Use this template to create general marketing pages** as the description. This metadata can be edited later.
4. Select **OK** to create the new template and open the template editor. The template editor shows an outline tree on the left and a property pane on the right.
5. In the outline tree, expand the nodes, and select the **HTML Head** slot.
6. If there aren't yet any modules in this slot, select the ellipsis button (...), and then select **Add Module**.
7. In the **Add Module** dialog box, select **Default page summary**, and then select **OK**.
8. In the outline tree, select the new module, and then, in the property pane, enter any default settings that should be automatically configured for all child pages of the template. If you don't want any default settings, leave the values blank.
9. In the outline tree, select the **Body** slot, select the ellipsis button, and then select **Add Module**.
10. Select a page container module (there might be only one option), and then select **OK**.

Under the new page container module, you will see a new set of slots (**Header**, **Main**, and so on). Here, you can add and configure the module options that will be available to authors when they create pages from this

template. By default, if you don't add any modules to a slot, all available modules types are supported for that slot.

The template is now technically valid, and it can be saved, checked in, and used to create new pages. However, the next three sections describe some other default settings that you might want to configure first.

## Add a header and a footer

If your site already has a header fragment, follow these steps to add a header and a footer to a template.

1. In the outline tree, expand the **Body** slot and its child page module.
2. Select the **Header** slot.
3. Select the ellipsis button for the **Header** slot, and then select **Add Fragment**.
4. Search for and select your site's header fragment, and then select **OK**.

All pages that use the template will now automatically inherit this header fragment.

If your site doesn't yet have a header fragment, see [Create a fragment](#) for information about how to create it, and then complete the previous procedure.

## Change the template theme

To set the default theme for all pages that use a template, follow these steps.

1. In the outline tree on the left, expand the **Body** slot.
2. In the **Body** slot, select the page container module (for example, **Default Page**).
3. In the property pane on the right, in the **Theme** field, select a theme.

By default, all new pages will now use the selected theme. To prevent pages from overriding this setting at the layout or page level, set the **Locked** Boolean control to **True**.

## Add a script to a template

You can add HTML `<script>` elements that contain JavaScript to your template. In this way, you can provide default script behaviors to the HTML head, body begin, and body end sections of your pages.

To add a script to a template, follow these steps.

1. In the outline tree on the left, select the slot where you want to add the `<script>` element (for example, the HTML head, body begin, or body end).
2. Select the ellipsis button for the slot, and then select **Add Module**.
3. In the **Add Module** dialog box, select a script module (for example, **External Script** or **Inline Script**).
4. In the property pane on the right, in the appropriate script property control (for example, **Inline Script** or **Script tags**), enter your script.
5. In the property pane, enter any other optional settings that you want to configure.

### TIP

If you want to reuse any of your script modules for other templates, you can convert them to fragments. In this way, you help make the authoring process more efficient, and you centralize the update process. For information about how to convert a script module to a fragment, see [Save an existing module configuration as a fragment](#).

## Save, check in, preview, and publish a template

To save and check in a template, follow these steps.

1. Select **Save** at the top of the template editor. Saved changes don't affect downstream pages until they are checked in.
2. Select **Finish editing**. Your changes are now discoverable for downstream workflows.

To preview your changes, either open an existing page that uses the template or create a new page from the template.

After you've previewed the changes to your template, follow one of these steps to publish the template to your live site:

- Go to **Templates**, select the template, and then select **Publish**.
- Select the layout name to open the layout editor, and then select **Publish**.
- Publish a page that references the unpublished template. The template is automatically published.

#### **WARNING**

When a template, or any other content management system (CMS) item, is published, it's discoverable on the internet. Don't publish documents or assets until you're ready to make them public. Document versions that have been saved and checked in, but that haven't been published, are discoverable only to authenticated system users.

## Additional resources

[Templates and layouts overview](#)

[Work with preset layouts](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Work with preset layouts

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to work with preset layouts in Microsoft Dynamics 365 Commerce.

## Overview

Before you complete the procedures in this topic, be sure to read [Preset and custom layouts](#). For a general overview, see [Templates and layouts overview](#).

## Create a new preset layout

There are two methods for creating a preset layout. You can save an existing custom layout as a new preset layout, or you can create a preset layout from scratch.

### Create a preset layout from an existing custom layout

To create a preset layout from an existing custom layout, follow these steps.

1. Open an existing page that doesn't currently use a preset layout, and that has a module structure that you want to reuse for other pages on your site.
2. Select **Edit** to check out the page.
3. Select **Save as new layout**. The **Save as new layout** dialog box appears.
4. Enter a name and description for your preset layout. The values that you enter will be shown to other authors when they create new pages from your layout or switch to it. Therefore, enter values that will be useful to page authors.
5. Select **OK**.

The preset layout will now be available when you create new pages or select a different layout for a page in the same template hierarchy.

#### TIP

To quickly see whether a specific page is currently bound to a preset layout, select the page in the pages list view, and inspect the layout metadata in the property pane on the right.

### Create a new preset layout

To create a preset layout from scratch, follow these steps.

1. In the navigation pane on the left, select **Layouts**.
2. Select **New Layout**. The **New layout** dialog box appears.
3. Select the template to use for your preset layout.
4. Enter a name and description for your preset layout. The values that you enter will be shown to other authors when they create new pages from your layout or switch to it. Therefore, enter values that will be useful to page authors.
5. Select **OK** to create the new preset layout and open the layout editor.
6. In the layout editor, add and configure modules by using the outline tree on the left and the property pane on the right. (The process resembles the process for adding and configuring modules for a template in the template editor.) The arrangement of modules and any locked default settings become the centralized module configuration for any pages that use this preset layout.



# Modify a preset layout

To modify a preset layout, follow these steps.

1. In the navigation pane on the left, select **Layouts**.
2. In the layout editor, in the outline tree on the left, select the module to modify. Then follow any of these steps:
  - To move a module up or down inside its parent, select the ellipsis button (...) for the module, and then select **Move up** or **Move down**.
  - To change a module's default settings, use the property pane to enter default values and optionally lock them for all downstream pages.
  - To add new modules or fragments to container modules, select the ellipsis button, and then select **Add module** or **Add fragment**.
  - To remove a module from the layout, select the ellipsis button, and then select **Delete**.

## Change a preset layout theme

A typical practice is to set a default theme for all pages that use a preset layout.

To set or change the theme for all child pages that use your preset layout, follow these steps.

1. In the layout editor, in the outline tree on the left, select the page container module. (Typically, this module is the second node and is named **Default page**.)
2. In the property pane on the right, in the **Theme** field, select a theme.

## Save, check in, preview, and publish a preset layout

To save and check in your preset layout, follow these steps.

1. Select **Save** at the top of the layout editor. Saved changes don't affect downstream pages until they are checked in.
2. Select **Finish editing**. Your changes are now discoverable for downstream workflows.

To preview your changes, either open an existing page that uses the preset layout or create a new page from the layout.

After you've previewed the changes to your preset layout, follow one of these steps to publish the layout to your live site:

- Go to **Layouts**, select the layout, and then select **Publish**.
- Select the layout name to open the layout editor, and then select **Publish**.
- Publish a page that references the unpublished layout. The layout will automatically be published.

### WARNING

Preset layouts can be referenced by multiple pages. When you publish a preset layout, be aware that you might affect the layout of multiple pages.

## Additional resources

[Templates and layouts overview](#)

[Work with templates](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Modify an existing site page

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to modify an existing site page in Microsoft Dynamics 365 Commerce.

## Overview

When you must modify a page, the first step is to open it in the page editor. Go to the site that contains your page, and then, in the list of pages, find the page that you want. If you can't find the page, you can use the authoring tool's rich search functionality. Either type the exact page name, or type the first few letters of it and then an asterisk (\*). A filtered list of pages appears. You can use this list to find the page that you want. After you find the correct page, select the page name to open the page in the page editor.

### TIP

If your page is visible in the page inspector, you can select **Edit** and check the page out before you open it in the page editor. In this way, you can check out multiple pages at the same time.

After the page is open in the page editor, you must make sure that it's checked out to you. The command bar in the authoring tool is dynamic, context-sensitive, and state-sensitive. Therefore, it shows only the actions that you can currently perform on the page. For example, if the page isn't checked out to you, the **Save** and **Finish editing** buttons don't appear on the command bar. The state of the page is also shown on the right side of the window.

If the page isn't already checked out to you, select **Edit** on the command bar. The command bar changes to reflect the new state of the page. You also receive a notification that states that the page was checked out to you.

The next step is to make your actual changes. Often, you will use the page outline tree on the left to find and select the module that you want to change, and then make changes in the properties pane on the right.

However, your change might sometimes involve adding or removing models or fragments. To add a fragment or module, use the page outline tree to find the slot that you want to add the module or fragment to, and then select the ellipsis button (...) for that slot. A menu appears that includes commands for adding a module or fragment. To remove a module or fragment, find and select it in the page outline tree, select the ellipsis button, and then select the command to delete the module or fragment.

### TIP

You can also view and edit the properties for any module that is visible in the visual page builder preview by selecting it directly.

After you've finished making your changes and previewing their effect, you should check in the page by selecting **Finish editing** on the command bar.

To publish your changes immediately, select **Publish** on the command bar. The latest checked-in version of the page that you modified is published and becomes available to external users who view your site.

## Example: Change the video on the home page

The following example shows how to modify the home page by changing the video that appears in the video

player module.

1. Under **Sites**, select **Fabrikam** (or the name of your site).
2. In the navigation pane on the left, select **Pages**.
3. Find and select the home page to open it in the page editor.
4. On the command bar, select **Edit**.
5. In the page outline, select the **Main** slot.
6. Under the **Main** slot, expand all the fluid container modules.
7. Find and select the video player module.
8. In the properties pane on the right, select the **video** property. The asset picker appears.
9. In the asset picker, select an available video asset, or select **Upload new asset** to upload a new video asset.
10. Select **OK**.
11. Select **Save**, and then select **Finish editing**.
12. In the **Comments** field, enter **Changed the video**, and then select **OK**.
13. Select **Preview** to preview the updated page. When you've finished, close the preview tab to return to the authoring tool.
14. Select **Publish**.

## Additional resources

[Add a new site page](#)

[Select page layouts](#)

[Manage SEO metadata](#)

[Save, preview, and publish a page](#)

[Enrich a product page](#)

[Enrich a category landing page](#)

[Verify page content accessibility](#)

[Create dynamic e-commerce pages based on URL parameters](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add a new site page

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add a new site page in Microsoft Dynamics 365 Commerce.

## Overview

After you've created templates and fragments for your site, the next step is to start to create pages that use them. To get started, you must select a template or layout, a page name, and a page URL.

## Template or layout

You can use either a template or a layout for your new page. For more information, see [Templates and layouts overview](#).

## Page name

The page name must be unique to your page. It should be descriptive, so that you can easily find it and other people know what the page is intended for. Choose the page name carefully, because it can't be changed later.

## Page URL

You can have the option to enter a URL for your new page. When you create a page, you can enter a string that will be used to form a complete URL. This string is known as a relative URL or a URL slug. A complete URL is then generated based on the URL slug, and the new page is assigned to it. You can change the URL slug later, before you publish the page. For more information, see [Create a page URL](#).

### NOTE

Dynamics 365 Commerce decouples URLs and content. In other words, a page can be created that isn't associated with an URL, and a URL can be created that isn't associated with a page. Therefore, content swapping can be done for a URL and doesn't require downtime, and redirects are easier to manage.

## Add a new page

To add a new site page to your site, follow these steps.

1. Under **Sites**, select **Fabrikam** (or the name of your site).
2. Select **New Page**.
3. In the **New Page** dialog box, select a template, and then select **OK**.
4. In the **Page Name** field, enter a page name (for example, **My New Page**).
5. In the **URL** field, enter a string (URL slug) to complete the URL (for example, **mynewpage**).
6. Select **OK**. The page editor appears. Notice that a header and a footer are automatically added to the page, based on the template that you selected.
7. In the page outline, select the **Main** slot, select the ellipsis button (...), and then select **Add Module**.
8. Select **Container**, and then select **OK**.
9. Select **Fluid Container**, select the ellipsis button, and then select **Add Module**.
10. Select **Content Rich block**, and then select **OK**.

11. Select **Content Rich Block**, select the ellipsis button, and then select **Add Module**.
12. Select **Content rich block item**, and then select **OK**.
13. In the properties pane on the right, select **Paragraph**, and then, in the field, enter **My test text**.
14. Select **Save**, and then select **Finish editing**.
15. In the **Comments** field, enter **Added new page**, and then select **OK**.
16. Select **Preview** to preview your page. When you've finished, close the preview tab to return to the authoring tool.
17. Select **Publish**.
18. In the navigation path (breadcrumbs), select **Fabrikam** (or the name of your site).
19. In the navigation pane on the left, select **URLs**.
20. Find and select your URL (**mynewpage**) in the list.
21. Select **Publish**.

## Additional resources

[Modify an existing site page](#)

[Select page layouts](#)

[Manage SEO metadata](#)

[Save, preview, and publish a page](#)

[Enrich a product page](#)

[Enrich a category landing page](#)

[Verify page content accessibility](#)

[Create dynamic e-commerce pages based on URL parameters](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Select page layouts

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to create and select page layouts in Microsoft Dynamics 365 Commerce.

## Create layouts for an existing page

### NOTE

You can create layouts for an existing page only if that page has at least two modules under the main slot.

To create layouts for an existing page, follow these steps.

1. Go to **Pages**, and find the existing page in the list. Use the search feature as you require.
2. Select the page, select **Edit** to check it out, and then select the page name to open it. Make a note of the module order.
3. Select **Save as New Layout**.
4. Enter a name for the layout, and then select **OK**.
5. Select **Convert to Embedded Layout**.
6. Change the order of the modules as you require, and make a note of the new order.
7. Select **Save as New Layout**.
8. Enter a name for the layout, and then select **OK**.
9. Select **Change Layout**, select the first layout that you created, and then select **OK**. Make a note of the module order. Change it so that it matches the module order that was saved with the layout.
10. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Select a different layout for an existing page

### NOTE

You can select a different layout for an existing page only if the template that was used to create that page has more than one layout.

To select a different layout for an existing page, follow these steps.

1. Go to **Pages**, and find the existing page in the list. Use the search feature as you require.
2. Select the page, select **Edit** to check it out, and then select the page name to open it.
3. Select **Change layout**.
4. Select the new layout for the page, and then select **OK**. The page editor is refreshed to show the new layout.
5. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Modify an existing site page](#)

[Add a new site page](#)

[Manage SEO metadata](#)

Save, preview, and publish a page

Enrich a product page

Enrich a category landing page

Verify page content accessibility

Create dynamic e-commerce pages based on URL parameters

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Manage SEO metadata

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to manage search engine optimization (SEO) metadata in Microsoft Dynamics 365 Commerce.

## Overview

SEO metadata for a site can be managed by using site maps and page metadata.

## Site maps

A site map is a machine-readable list, in XML format, of the pages on your website. It's intended to be consumed by search engines, so that they can provide better search results from your site. Site maps can be manually ingested by search engines or published in a robots.txt file.

Dynamics 365 Commerce supports automatic generation of site maps. Site maps are automatically updated when pages are published and unpublished.

### Turn on site map generation

1. Sign in to the authoring tool.
2. Under **Sites**, select **Fabrikam** (or the name of your site).
3. In the navigation pane on the left, select **Site Management**.
4. Make sure that the **Site maps enabled** option is turned on.

## Page metadata

Dynamics 365 Commerce lets you manage SEO metadata for individual pages. You can view and modify this information in the **SEO Properties** section of a page container. The following SEO metadata properties are supported:

- Title
- Description
- SEO keywords
- Aria labels
- noindex
- nofollow
- noarchive
- nocache
- noOpenDirectoryProject
- nosnippet
- noImageIndex
- unavailableAfter

### Modify page metadata

To modify page metadata, follow these steps.

1. Under **Sites**, select the **Fabrikam** (or the name of your site).
2. In the navigation pane on the left, select **Pages**.

3. Select the home page to open it in the page editor.
4. On the command bar, select **Edit**.
5. In the properties pane on the right, expand **Default metatags**.
6. To add a new metatag, select **Add**, and then enter the tag in the field. To remove an existing metatag, select the trash can symbol to the right of it.
7. Select **Save**, and then select **Finish editing**.
8. In the **Comments** field, enter **Updated metatags**, and then select **OK**.
9. Select **Preview** to preview your page. When you've finished, close the preview tab to return to the authoring tool.
10. Select **Publish**.

## Additional resources

[Modify an existing site page](#)

[Add a new site page](#)

[Select page layouts](#)

[Save, preview, and publish a page](#)

[Enrich a product page](#)

[Enrich a category landing page](#)

[Verify page content accessibility](#)

[Create dynamic e-commerce pages based on URL parameters](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Save, preview, and publish a page

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to save, preview, and publish a page in Microsoft Dynamics 365 Commerce.

## Save a page

To save a page, you must have it checked out to yourself and open in the page editor. To check out a page, select **Edit** on the command bar. After you've finished editing a page, you should immediately save it to ensure that your changes are stored.

When you save a page, the changes are visible only to you. The save operation is intended primarily to store changes while the page isn't yet ready to be checked in. When you've finished modifying the page, we recommend that you check it in, so that the changes become visible to others. At that point, the page can also be checked out by other users who must modify it.

## Preview a page

The authoring tool offers two kinds of preview features: visual page builder, which is a "what you see is what you get" (WYSIWYG) preview pane in the page editor, and a separate preview window.

While you're using the page editor, a "what you see is what you get" (WYSIWYG) preview appears in the center pane. This preview is automatically updated whenever you save the page. You can also manually update it by selecting **Refresh** on the command bar. The preview renders the page just as the site's users will see it, but authoring helpers are rendered on top of it.

When you've finished modifying the page, you might want to preview it to see what customers will see. To preview a page, select **Preview** on the command bar. The preview will appear in a separate browser window. The page in the preview window is rendered just as the site's user will see it. You can resize the window to make sure that responsive modules are correctly rendered in all view ports.

### NOTE

Authentication and correct permissions are required to preview unpublished content. Therefore, if you share the URL of the preview with someone, that person must have the correct permissions to access the content.

## Publish a page

When your page is ready, the next step is to publish it, so that external users can view the content. Before you can publish a page, you must check it in by selecting **Finish editing** on the command bar.

You can publish and unpublish pages from either the page inspector or the page editor. The page inspector shows a list of pages and allows for bulk operations. The page editor can be used to publish or unpublish only the single page that is open in it.

To publish one or more pages from the page inspector, select the pages, make sure that they are checked in, and then select **Publish** on the command bar. The pages are published, and you receive a notification about the operation in the authoring tool.

To publish a single page from the page editor, the procedure is similar. While the page is open in the page editor, make sure that it has been checked in, and then select **Publish** on the command bar. The page is published, and

you receive a notification about the operation.

When you publish a page, just the page content is published. You and other users can go to the page and view it only after a URL is associated with it. The URL must be published separately.

#### **IMPORTANT**

Before you can publish a page, any images or fragments that the page references must already be published.

## Save, preview, and publish a home page

To save, preview, and publish a home page, follow these steps.

1. Under **Sites**, select **Fabrikam** (or the name of your site).
2. In the navigation pane on the left, select **Pages**.
3. Find and select the home page to open it in the page editor.
4. Select **Edit**.
5. Modify the page as you require.
6. Select **Save**, and then select **Finish editing**.
7. In the **Comments** field, enter a note about the changes that you made, and then select **OK**.
8. Select **Preview** to preview your page. When you've finished, close the preview tab to return to the authoring tool.
9. Select **Publish**.

## Publish a URL

To publish a URL, follow these steps.

1. Under **Sites**, select **Fabrikam** (or the name of your site).
2. In the navigation pane on the left, select **URLs**.
3. Find and select the URL to publish.
4. On the command bar, select **Publish**.

## Additional resources

[Modify an existing site page](#)

[Add a new site page](#)

[Select page layouts](#)

[Manage SEO metadata](#)

[Enrich a product page](#)

[Enrich a category landing page](#)

[Verify page content accessibility](#)

[Create dynamic e-commerce pages based on URL parameters](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enrich a product page

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to enrich a product page in Microsoft Dynamics 365 Commerce.

## Overview

By default, your site uses a generic page to show product data. This page includes the basic information about the product and the controls that are required to sell it. However, you can supplement the information that comes from the Commerce Scale Unit with additional images or text for a specific product. This process is known as enriching the product page.

In many cases, you will want to use specific additional content for your products. When you go to **Retail and Commerce** in the authoring tool, you will see a list of products from the channel that is assigned to the site. In this list, the **Enriched** column indicates whether the product page for a product has been enriched. If a check mark appears in the column, an enriched product page exists for the product. If no check mark appears, the default product page and content are used for the product. You can preview both enriched and non-enriched product pages by selecting a product name in the list.

## Enrich a product page

To enrich a product page, follow these steps.

1. Under **Sites**, select **Fabrikam** (or the name of your site).
2. In the navigation pane on the left, select **Products**.
3. Select any product that doesn't have an enriched product page.
4. On the Action Pane, select **Enrich product page**.
5. Select **PDP-template**, and then select **OK**.
6. In the page outline tree on the left, expand the **Main** slot.
7. Select the ellipsis button (...) for the **Main** slot, and then select **Add Module**.
8. Select **Container 2**, and then select **OK**.
9. Select the ellipsis button for **Container 2**, and then select **Add Module**.
10. Select **Feature**, and then select **OK**.
11. In the properties pane on the right, in the **Rich Text** field, enter the updated description of the product.
12. In the **Heading** field, enter heading text, and then select **OK**.
13. Select **Save**, and then select **Finish editing**.
14. In the **Comments** field, enter **Enriched a product**, and then select **OK**.
15. Select **Preview** to preview the enriched product page. When you've finished, close the preview tab to return to the authoring tool.
16. Select **Publish**.

## Additional resources

[Modify an existing site page](#)

[Add a new site page](#)

[Select page layouts](#)

[Manage SEO metadata](#)

[Save, preview, and publish a page](#)

[Enrich a category landing page](#)

[Verify page content accessibility](#)

[Create dynamic e-commerce pages based on URL parameters](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enrich a category landing page

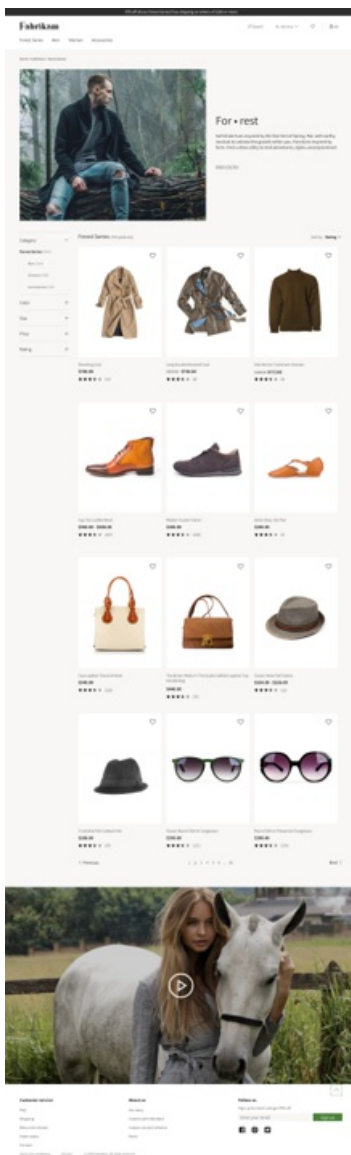
2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the enrichment of category pages in Dynamics 365 Commerce.

## Overview

Commerce provides a default category landing page that is used when category data is shown. A default category page contains required elements, such as refiners, categorized product placement, sorting options, a choice summary, and pagination controls.

However, instead of using the default category page, you might want to use an "enriched" category landing page that has more content and more specific elements. A typical enrichment might involve adding category-specific marketing content to the category page. This content might include cross-category product placement for cross-sell purposes, editorial lists, images, videos, and other text. You can either modify the default category page or define a different category page for a specific category.



In Commerce site builder, the **Products** page includes a list of categories from the channel that are assigned to the site. If the **Enriched** status is selected for a category page, that category page has been enriched. Otherwise, the default category page and content are used for the category. You can preview both the enriched and non-



enriched category pages for a category by selecting the category name.

To enrich a category page, do the following.

1. On the **Products** page, select the name of the category for which you want to enrich the category page. The default category page for the selected category is opened in the page editor.
2. Select **Enrich category page**.
3. Select a template for the enriched category page. If you're making only minor changes, you can select the default category page. Alternatively, you can select a specific category page template. When you select the template, the page editor is opened, and the selected template is used to create a new category page for the selected category. The page is checked out to you, and you can now make your changes.

#### **NOTE**

Modules that use category specification data use the data from your selected category. The settings of the template that you select determine the changes that you can make.

## Additional resources

[Modify an existing site page](#)

[Add a new site page](#)

[Select page layouts](#)

[Manage SEO metadata](#)

[Save, preview, and publish a page](#)

[Enrich a product page](#)

[Verify page content accessibility](#)

[Create dynamic e-commerce pages based on URL parameters](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Verify page content accessibility

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to verify the accessibility of page content in Microsoft Dynamics 365 Commerce.

## Overview

When you've finished changing a page, you should make sure that the content is accessible to everyone on the web. In the Commerce authoring tools, you can easily verify the accessibility of page content by using the integrated [Microsoft Accessibility Insights](#) service. This service verifies your page content against the latest [World Wide Web Consortium \(W3C\) accessibility](#) guidelines.

The [Microsoft Accessibility Insights](#) integration must be turned on at the tenant or site level before you can use it.

## Turn on Microsoft Accessibility Insights for all the sites in your tenant

To turn on the [Microsoft Accessibility Insights](#) integration for all the Commerce sites in your tenant, follow these steps.

### NOTE

To access tenant settings, you must be signed in to Commerce as a system admin.

1. Sign in to Commerce as a system admin.
2. In the left navigation pane, select **Tenant Settings** (next to the gear symbol) to expand it.
3. Under **Tenant Settings**, select **Features**.
4. Set the **Accessibility Check** option to **On**.

## Turn on Microsoft Accessibility Insights for a single site

To turn on the [Microsoft Accessibility Insights](#) integration for a single Commerce site, follow these steps.

1. Under **Sites**, select **Fabrikam** (or the name of your site).
2. In the left navigation pane, select **Site Settings** to expand it.
3. Under **Site Settings**, select **Features**.
4. Set the **Accessibility Check** option to **On**.

## Verify the accessibility of the content on the home page

To use the integrated [Microsoft Accessibility Insights](#) service to scan and verify the content of your home page in Commerce, follow these steps.

1. Under **Sites**, select **Fabrikam** (or the name of your site).
2. In the left navigation pane, select **Pages**.
3. Find and select the home page to open it in the page editor.
4. On the command bar, select **Check accessibility**. The **Check Accessibility** page appears.
5. After the scan is completed, review the contents of the report.
6. If any checks failed, select each failed check item to expand it so that you can view more details.

7. To close the report after you've finished reviewing it, scroll to the bottom of the **Check Accessibility** page, and select **OK**.

## Additional resources

[Modify an existing site page](#)

[Add a new site page](#)

[Select page layouts](#)

[Manage SEO metadata](#)

[Save, preview, and publish a page](#)

[Enrich a product page](#)

[Enrich a category landing page](#)

[Create dynamic e-commerce pages based on URL parameters](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create dynamic e-commerce pages based on URL parameters

2/18/2021 • 3 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic describes how to set up a Microsoft Dynamics 365 Commerce e-commerce page that can serve dynamic content, based on URL parameters.

An e-commerce page can be configured to serve different content, based on a segment in the URL path. Therefore, the page is known as a dynamic page. The segment is used as a parameter to retrieve the page content. For example, a page that is named **blog\_viewer** is created and associated with the URL `https://fabrikam.com/blog`. This page can then be used to show different content, based on the last segment in the URL path. For example, the last segment in the URL `https://fabrikam.com/blog/article-1` is **article-1**.

Separate custom pages that override the dynamic page can also be associated with segments in the URL path. For example, a page that is named **blog\_summary** is created and associated with the URL `https://fabrikam.com/blog/about-this-blog`. When this URL is requested, the **blog\_summary** page that is associated with the `/about-this-blog` parameter is returned instead of the **blog\_viewer** page.

## NOTE

The functionality for hosting, retrieving, and showing dynamic page content is implemented by using a custom module. For more information, see [Online channel extensibility](#).

## Set up a dynamic e-commerce page

To set up a dynamic e-commerce page, you must create the dynamic page, create the base URL, and configure the route to the dynamic page.

### Create the page that will serve dynamic content

To create a page that will serve dynamic content, follow the steps in [Add a new site page](#). The page that you create will require implementation of a module that uses the last segment in the URL path to retrieve content from an external data source. For more information about custom module development, see [Online channel extensibility](#).

### Create the base URL for the dynamic page

To create the base URL for the dynamic page in Commerce site builder, follow these steps.

1. Go to **URLs**, and select **New > New URL**.
2. In the **Create new URL** dialog box, select **Internal page**. Under **URL path**, enter the path that will serve as the root for the dynamic page (in this example, `/blog`). Then select **Next**.
3. In the **Select a page** dialog box, select the page that you created to serve as the dynamic page, and then select **Save**.
4. Select **Publish**.

## Configure the route to the dynamic page

To configure the route to the dynamic page in Commerce site builder, follow these steps.

1. Go to **Site Settings > Extensions**.
2. Under **Parameterized URL paths**, select **Add**, and then enter the URL path that you entered when you created the URL (in this example, **/blog**).
3. Select **Save and publish**.

After the route is configured, all requests to the parameterized URL path will return the page that is associated with that URL. If any requests contain an additional segment, the associated page will be returned, and the page content will be retrieved by using the segment as a parameter. For example,

`https://fabrikam.com/blog/article-1` will return the **blog\_summary** page, and the page content will be retrieved by using the **/article-1** parameter.

## Override a parameterized URL with a custom page

To override a parameterized URL with a custom page in Commerce site builder, follow these steps.

1. Go to **URLs**, and select **New > New URL**.
2. In the **Create new URL** dialog box, select **Internal page**. Under **URL path**, enter the path that includes the segment to override (in this example, **/blog/about-this-blog**). Then select **Next**.
3. In the **Select a page** dialog box, select the custom page, and then select **Save**.
4. Select **Publish**.
5. If the custom page hasn't yet been published, go to **Pages**, select the custom page, and then select **Publish**.

After the custom page is published, it will be served instead of the dynamic page that has parameterized content.

## Additional resources

[Modify an existing site page](#)

[Add a new site page](#)

[Select page layouts](#)

[Manage SEO metadata](#)

[Save, preview, and publish a page](#)

[Enrich a product page](#)

[Enrich a category landing page](#)

[Verify page content accessibility](#)

[Online channel extensibility](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customize site navigation

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to create a customized online navigation hierarchy to organize your products for browsing on your Microsoft Dynamics 365 Commerce site.

## Overview

Online storefronts typically let customers discover and browse products by navigating through product categories. This capability is usually provided by tabs at the top of the page or by a navigation bar on the left. In Dynamics 365 Commerce, you can create and manage the hierarchal structure of your category navigation and the products that are included in the various categories.

## Create a channel navigation hierarchy

To create a channel navigation hierarchy, follow these steps.

1. Go to **Retail and Commerce > Products and categories > Category and product management**.
2. Select **Category hierarchies**, and then select **New**.
3. Name the hierarchy.

### NOTE

The topmost category that you create is the root category node. It won't be shown on your site. To create a category hierarchy where a single top-level node is shown on your site, create and name the category as a child of the root category.

4. Select **New category node**, and name the category.
5. Continue to create sibling and child categories as you require.

You can now assign products to each category that you created under the top-level category.

## Customize the order of categories

By default, the categories that you define will appear in alphabetical order on your site. However, you can also customize the display order of categories.

## Assign a category hierarchy type

1. Go to **Retail and Commerce > Products and categories > Category and product management**.
2. Select **Category hierarchies**.
3. On the Action Pane, on the **Category hierarchy** tab, in the **Set up** group, select **Associate hierarchy type**.
4. Select **New**.
5. In the **Category hierarchy type** field, select **Channel navigation hierarchy**.
6. In the **Category hierarchy** field, select the channel navigation hierarchy that you created earlier.

# Publish new or updated navigation hierarchies

To make your navigation hierarchy available to your online storefront, follow these steps.

1. Go to **Retail and Commerce > Channel setup > Channel categories and product attributes**.
2. In the tree on the left, select your online store.
3. Select **Publish channel updates**.
4. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
5. In the list, find and select **Job 1040**.
6. Select **Run now**.
7. Repeat steps 5 and 6 for jobs 1070 and 1150.

## Show categories on your site

To show your category hierarchy on your online storefront, you must add the navigation menu module in the appropriate location in a template or fragment. The navigation menu module will then show your navigation hierarchy, provided that you've published your navigation hierarchy to the channel that your site is bound to.

### NOTE

The navigation menu module that is included in the module library lets users navigate only to categories that don't have subcategories. If your customers should be able to navigate to categories that have subcategories, you must customize the navigation menu module.

## Add custom navigation options

On your navigation menu, you can add navigation options that aren't part of your product category hierarchy. For example, at the end of the list of product categories, you can add a **Contact Us** item that points to a contact page that you've built for your site.

To add custom navigation options to your navigation menu, follow these steps.

1. In the template or fragment that you want to customize, select the navigation menu module.
2. In the property pane, on the **Data** tab, select **Add item** to create a new content management system (CMS) navigation item.
3. Enter link text and a URL.
4. Repeat steps 2 and 3 to add more custom navigation options.
5. When you've finished, select **Save** to save the template or fragment, and then select **Finish editing** to check it in.

## Additional resources

[Templates and layouts overview](#)

[Work with templates](#)

[Work with preset layouts](#)

[Work with fragments](#)

[Work with modules](#)

[Create a page URL](#)

[Work with publish groups](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Create a page URL

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic covers the basic concepts and procedures for creating a page URL on your site.

## Overview

The full, or absolute, URL that points to a page on your site consists of distinct parts. For example, the URL

`https://www.contoso.com/en-us/contactus` has the following parts:

- `https://www.contoso.com` – The HTTP protocol and the site's domain.
- `/en-us` – The site's language path.
- `/contactus` – The relative URL for the **Contact Us** page. A relative URL is also known as a URL *slug*.

You establish your site's domain and optional language path when you set up the site. You can add more domains and language paths to your site through the online stores page in the site's settings.

The URL slug for a page exists as a standalone entity in the site authoring environment. A page URL consists of two parts: a name that represents the URL slug, and a pointer to a page on either your site or an external site. A page URL can also be configured to act as a redirect to another page on either your site or an external site.

## Create a page URL

There are two ways to create page URLs:

- Automatically, when you create a page
- Manually, from the **URLs** page

### Create a page URL when you create a page

If you provide a name in the **URL** field when you create a new page, a page URL that points to that page is automatically created on the **URLs** page. After you publish the URL and the page that it points to, site users (your customers) can access the page that is associated with the URL.

#### NOTE

If you publish a URL without publishing the page that it points to, site users receive a 404 error when they try to access the page. If you publish a page without publishing the URL that points to it, the page can't be accessed by using a URL.

### Manually create a page URL

When you create new pages, you aren't required to specify a page URL. If you leave the URL field blank, the page is created in an unlinked state. In this case, customers won't be able to access the page, even if it's published. To make the page accessible, you must manually create the URL and link it to the page.

To manually create the page URL for a page, follow these steps.

1. On the **URLs** page, select **New**.
2. Select the site page to associate with the URL.
3. Enter the URL slug, and then select **OK**.

At this point, the URL is in a draft state. It must be published before site users can access the associated page.

# Update a page URL

To update the target page of a page URL, follow these steps.

1. On the **URLs** page, select the URL to update.
2. In the property pane on the right, select the ellipsis button (...) next to the target page field.
3. In the dialog box, select a different page, and then select **OK**.
4. Save and publish the URL.

# Redirect a page URL

Sometimes, you might want your customers to view a different page when they request a specific URL. In these cases, the best and easiest approach is often to change the page that the page URL points to. However, you might have legitimate reasons for using HTTP 301 or 3023 redirects to redirect requests for a URL to a different URL.

To redirect a URL to a different URL, follow these steps.

1. On the **URLs** page, select the URL to update.
2. In the property pane on the right, select **Redirect**.
3. Select a destination for the redirect:
  - To point to another page on your site, select **Internal URL**, select the ellipsis button (...), and then select the URL to redirect to.
  - To point to a page on an external site, select **External URL**, and then enter the full URL for that page. Be sure to include the protocol. For example, enter `https://domain.com/new/page`. If the URL already redirects to an internal URL, you must select **Clear selection** before you can enter an external URL.
4. Select a redirect type:
  - **Permanent redirect (301)** – Select this option when you know that your content is moving permanently and won't revert to its previous URL. Search engines will assign the search engine optimization (SEO) value of the redirecting URL to the URL that is being redirected to and update their record to show the new URL.
  - **Temporary redirect (302)** – Select this option to redirect traffic without updating search engines. This approach is typically used if the content will soon revert to its previous URL.
5. When you're ready to implement the redirect, save and publish the URL.

# Additional resources

[Customize site navigation](#)

[Add a new site page](#)

[Configure your domain name](#)

[Add languages to your site](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Digital asset management overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of digital asset management in Microsoft Dynamics 365 Commerce site builder.

## Overview

Dynamics 365 Commerce site builder's Media Library features rich digital asset management functionality, including support for the following:

- Image assets
- Video assets
- Other types of binary assets, such as Office documents and PDF files
- Localized images
- Folder uploads
- Cropping of images
- Customizing of image focal points
- Omni-channel image assets (product, product variants, catalog, category, worker, and customer images)

Digital assets are channel-specific, with the exception of omni-channel assets, which are environment-wide by default.

Digital assets and management features are located within site builder under **Media Library** on the left navigation pane within a site.

## Additional resources

[Upload images](#)

[Upload video](#)

[Upload files](#)

[Crop images](#)

[Customize image focal points](#)

[Upload and serve static files](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upload images

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to upload images in Microsoft Dynamics 365 Commerce site builder.

## Overview

The Commerce site builder Media Library allows you to upload images, either singly or in bulk using folders. You should always upload the version of the image with highest resolution and quality, because the image resizer component will automatically optimize the image for different viewports and their breakpoints.

### Image information specified during upload

When uploading an image, the following information can be specified.

- **Title, Alt Text, Description, Keywords:** Metadata of the image or images. Title and alt text are required values.
- **Select category:**
  - **None:** Used for an e-Commerce storytelling image or images.
  - **Product, Category, Customer, Employee, Catalog:** Used for Dynamics 365 Commerce omni-channel image or images.
- **Publish assets after upload:** When this check box is selected, the image or images are published immediately after upload.

#### NOTE

Image assets with a category assigned are also automatically tagged with the category as a keyword to aid searching for assets of a specific category.

### Naming conventions for omni-channel images

If you have configured the Media Library as the omni-channel image backend, you can use image categories to indicate which category the uploaded image belongs to. There is also a naming convention that should be followed to ensure that images are retrieved correctly by other channels, such as point of sale (POS).

The default naming convention varies based on the category:

- Catalog images should be named `"/Catalogs/{LanguageId}/{CatalogName}.jpg"`
- Category images should be named `"/Categories/{CategoryName}.png"`
- Customer images should be named `"/Customers/{CustomerNumber}.jpg"`
- Employee images should be named `"/Workers/{WorkerNumber}.jpg"`
- Product images should be named `"/Products/{ProductNumber}_000_001.png"`
  - 001 is the sequence of the image and it can be 001, 002, 003, 004 or 005
- Product variant images should be named `"/Products/{ProductNumber}_{Size}_{Color}_{Style}_000_001.png"`

## Upload an image

To upload an image in site builder, follow these steps.

1. In the left navigation pane, select **Media Library**.

2. On the command bar, select **Upload > Upload Media Items**.
3. In the File Explorer window, navigate to and select one or more image files to upload, and then select **Open**.
4. In the **Upload Media Item** dialog box, enter the required title and alt text.
5. Enter optional description and keywords and select a category if desired.
6. If you want to publish the image(s) immediately after upload, select the **Publish media items after upload** check box.
7. Select **OK**.

## Upload a folder of images

To bulk upload a folder of images in site builder, follow these steps.

1. In the left navigation pane, select **Media Library**.
2. On the command bar, select **Upload > Upload Folder**.
3. In the File Explorer window, navigate to and select a folder to upload, and then select **Open**.
4. In the **Upload Media Items** dialog box, enter optional keywords and select a category if desired.
5. If you want to publish the images in the folder immediately after upload, select the **Publish media items after upload** check box.
6. Select **OK**.

## Additional resources

[Digital asset management overview](#)

[Upload video](#)

[Upload files](#)

[Crop images](#)

[Customize image focal points](#)

[Upload and serve static files](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upload videos

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to upload videos in Microsoft Dynamics 365 Commerce site builder.

## Overview

The Commerce site builder Media Library allows you to upload videos. You should always upload the version of a video with the highest bitrate and resolution, because the video will be automatically converted to be suitable for different viewports and their breakpoints.

### Video information specified during upload

When uploading a video, the following information can be specified.

- **Title, Description, Keywords:** Metadata of the video.
- **Automatically generate closed captions:** Specifies whether closed captions should be automatically generated for the video.
- **Closed Caption:** Specifies the closed captions to be used.
- **Regular Audio:** Specifies the regular audio track to be used.
- **Thumbnail:** Specifies the thumbnail for the video. If not specified, it will be generated automatically.
- **Descriptive Audio:** Specifies the descriptive audio track to be used.

## Upload a video

To upload a video in site builder, follow these steps.

1. In the left navigation pane, select **Media Library**.
2. On the command bar, select **Upload > Upload Media Items**.
3. In the File Explorer window, navigate to and select one or more video files to upload, and then select **Open**.
4. In the **Upload Media Item** dialog box, enter the required title and alt text.
5. Enter optional description and keywords and select a category if desired.
6. If you want to publish the image(s) after immediately upload, select the **Publish media items after upload** check box
7. Select **OK**.

If you are uploading multiple types of assets simultaneously (for example, images and videos), in the **Upload Media Item** dialog box you will only be able to specify keywords, whether the files should be published immediately after upload, and whether closed captions should be automatically generated for video files. All the assets will share the same keywords.

## Additional resources

[Digital asset management overview](#)

[Upload images](#)

[Upload files](#)

[Crop images](#)

[Customize image focal points](#)

## Upload and serve static files

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upload files other than images and videos

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to upload files other than images and videos in Microsoft Dynamics 365 Commerce site builder.

## Overview

The Commerce site builder Media Library supports the uploading of binary assets other than images or videos. For example, you might want to upload Microsoft Excel, Microsoft Word, Microsoft PowerPoint, or PDF files.

The following document types are supported:

- 7Z
- AVI
- CS
- CSS
- DOC
- DOCX
- EPUB
- GIF
- INDD
- JAR
- JPG
- JPEG
- JS
- MP3
- MP4
- MPEG
- MPG
- ODP
- ODS
- ODT
- PDF
- PNG
- PPT
- PPTX
- PS
- QXP
- RAR
- RTF
- SVG
- TAR
- TGZ
- TXT
- WMV



- XLS
- XLSX
- XML
- ZIP

## Upload a file

To upload a file to Commerce site builder, follow these steps.

1. In the left navigation pane, select **Media Library**.
2. On the command bar, select **Upload > Upload Media Items**.
3. In File Explorer, select one or more files and then select **Open**.
4. In the **Upload Media Item** dialog box, enter title, description, and keyword metadata as needed.
5. To publish the file(s) immediately after upload, select the **Publish media items after upload** check box.
6. Select **OK**.

## Additional resources

[Digital asset management overview](#)

[Upload images](#)

[Upload video](#)

[Crop images](#)

[Customize image focal points](#)

[Upload and serve static files](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Crop images

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to crop images in Microsoft Dynamics 365 Commerce site builder.

## Overview

The Commerce site builder Media Library allows you to crop images to optimize them for different module types and viewports.

## Crop an image

To crop an image in site builder, follow these steps.

1. In the left navigation pane of Commerce site builder, select **Media Library**.
2. In the main window, select the image you want to modify.
3. On the command bar, select **Edit**.
4. Select the image to enter **Edit Mode**.
5. Under **Edit Mode**, select **Edit View by Module**.
6. From the **Module** drop-down menu, select the module type.
7. From the **View type** drop-down menu, select the view type.
8. From the **Placement** drop-down menu, select the image placement.
9. From the **Viewport** drop-down menu, select the viewport size.
10. The image is overlaid with the area representing the crop region. Move and resize the crop region as needed. The aspect ratio will be maintained automatically.
11. When you're done, on the command bar, select **Save**, and then select **Finish editing**.

After custom cropping is completed, image modifications will take effect almost immediately.

## Additional resources

[Digital asset management overview](#)

[Upload images](#)

[Upload video](#)

[Upload files](#)

[Customize image focal points](#)

[Upload and serve static files](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customize image focal points

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to customize image focal points in Microsoft Dynamics 365 Commerce site builder.

## Overview

When an image is uploaded to the Commerce site builder Media Library, the system attempts to determine the focal point of the image. For example, if the image has a person on it, the system will set the focal point to the face of the person by default. In most cases the automatically set focal point works well for all viewports, but sometimes you may want to adjust the focal point to ensure that a specific part of the image is always visible.

### Define a custom focal point for an image

To define a custom focal point for an image, follow these steps.

1. In the left navigation pane of Commerce site builder, select **Media Library**.
2. In the main window, select the image you want to modify.
3. On the command bar, select **Edit**.
4. Select the image to enter **Edit Mode**.
5. Under **Edit Mode**, select **Change Focal Point**. A circular focal point control appears over the image.
6. Select the focal point control to move it over the desired focal point.
7. When you're done, on the command bar select **Save**, and then select **Finish editing**.

## Additional resources

[Digital asset management overview](#)

[Upload images](#)

[Upload video](#)

[Upload files](#)

[Crop images](#)

[Upload and serve static files](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upload and serve static files

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to upload a static file into Microsoft Dynamics 365 Commerce site builder, and how to create a custom URL and file name that can be used to request that file.

Some third-party connectors require that a file be hosted and served from the e-commerce site. These connectors expect that the file will be returned by requests to a specific callback URL path and file name. Therefore, this topic explains how to upload and serve a static file that has a user-definable URL and file name on a Dynamics 365 Commerce e-commerce site.

## Create a site URL that returns a static file

To create a site URL that returns a static file in Commerce site builder, follow these steps.

1. Go to your site's Media library, and upload the file that should be served by requests to the URL that you will define. If you've already uploaded the file, you can skip this step.
2. Go to **URLs** for your site.
3. Select **New > New URL**.
4. In the **New URL** dialog box, select **Media library asset**.
5. In the **URL path** field, enter the URL path. Include the file name in the path.
6. Select **Next**. The Media library is opened and shows all media assets of the **document** type that have been uploaded.
7. Select the file that should be served for requests to the URL that you defined in step 5.
8. Select **Save**.

At this point, the URL that you created is in a draft state. The file that the URL points to won't be returned until you publish the URL. Before you publish the URL, you can validate that it returns the correct data.

## Validate and publish a URL

To validate an URL before you publish it, follow these steps.

1. Go to **URLs** for your site, and select the URL to preview.
2. In the properties pane on the right, below the **Edit** button, select the correct URL link. A new browser window is opened, and you should receive a 404 error.
3. Append the **?preview=inprogress** query string to the URL (for example, `https://yoursite.com/callback.html?preview=inprogress`), and reload the page. The file that you uploaded to the Media library should be returned in the response.

After you've validated the URL, you can publish it.

1. Go to **URLs** for your site, and select the URL.
2. Select **Publish** on the command bar.

## Update the file that a URL points to

After a URL is published, you can update it so that it points to a different file. Alternatively, you can update the URL so that it points to a different the type of resource, as described in the next section. For example, you can point the URL to an internal page or a redirect.

To update the file that a URL points to, follow these steps.

1. Go to **URLs** for your site, and select the URL to update.
2. In the properties pane on the right, select **Edit**.
3. Under **URL assignment**, select the **Step 2** box, and then select a new document from the Media library.
4. Select **Apply**.

## Update the asset type that a URL points to

You can also update a URL so that it points to a different type of asset (resource), such as an internal page or a redirect.

To update the asset type that a URL points to, follow these steps.

1. Go to **URLs** for your site, and select the URL to update.
2. In the properties pane on the right, select **Edit**.
3. Under **URL assignment**, under **Step 1**, select a different asset type.
4. Select the **Step 2** box, and then select the new asset.
5. Select **Apply**.

## Change the URL path

After a URL is created, its path can't be changed. If you must change the URL path that serves a file or any other type of resource, you have to create a new URL, map it to the existing file or other resource, and then unpublish and delete the old URL.

To change the URL path, follow these steps.

1. To create a new URL and map it to the existing file or another resource, follow the instructions in the [Create a site URL that returns a static file](#) section earlier in this topic.
2. Select the new URL, and select **Publish** on the command bar. The new URL is published.
3. To unpublish the old URL, select it, and then select **Unpublish** on the command bar. You can now delete the old URL if you want.

## Additional resources

[Digital asset management overview](#)

[Upload images](#)

[Upload videos](#)

[Upload files other than images and videos](#)

[Crop images](#)

[Customize image focal points](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Compliance overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic is an overview of the compliance guidance that is provided with Microsoft Dynamics 365 Commerce to help you make your company's sites compliant. Compliance is an important factor for all businesses, because it helps their user base connect with their site content. By staying compliant, you can also help protect your company from expensive legal action or hefty fines.

The compliance documentation includes a review of areas that you should consider when you use Dynamics 365 Commerce, to help you meet your company's compliance requirements.

You're responsible for reviewing your company's compliance requirements, and for authoring and building sites that meet the standards of those requirements. The following topics provide examples that show how you can take compliance into account as you use the Commerce authoring tools.

To learn more about the basic principles that Microsoft uses, visit the [Microsoft Trust Center](#). On that site, you can also get more information about areas of compliance.

## Additional resources

[Accessibility features and capabilities](#)

[Cookie compliance](#)

[Add a privacy policy page](#)

[Replace user IDs associated with tracked content changes](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Accessibility features and capabilities

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides information about the accessibility features and capabilities in Microsoft Dynamics 365 Commerce.

## Overview

Accessibility features and capabilities provide the functional means for all users to access and perform actions so that they can accomplish their goals. This broad range of users might require assistive tools for hearing, vision, mobility, or neurodiversity.

Various features in Dynamics 365 Commerce let you build your site so that it includes assistive functionality. When you design your site, you should consider the areas of accessibility functionality that are mentioned in the [Microsoft Accessibility Center](#).

This topic describes some additional areas of accessibility functionality that you should consider when you use Dynamics 365 Commerce.

## Image alt text

Dynamics 365 Commerce has a built-in digital asset management system to track image and video assets that are used on your site. Image captions, descriptions, and alt text can be added in the properties pane for an image when it's selected or uploaded.

## Video accessibility

The Dynamics 365 Commerce digital asset management system supports several accessibility features for video content. The following table lists some examples.

VIDEO FEATURE	DESCRIPTION
Closed captioning (CC)	Text that can be shown for the audio and audio descriptive elements of a video, to help users who are deaf or hard of hearing
Subtitles	Caption files that show the text of context clues or dialog on-screen
Audio transcripts	A textual transcript of spoken words that is generated from the audio of a video asset
Descriptive audio	A non-primary audio channel that describes the content or context that is occurring on-screen
Minimum age gate	An attribute that can store the minimum age that a viewer must be to view a video (metadata only)

### Configure video accessibility elements

In the Commerce **Media Library** section for your site, you can upload video assets that have separate files for closed captions, regular audio, and descriptive audio. Closed captions can also be generated automatically when

a video asset is uploaded.

#### **Generate or upload closed caption files during video asset upload**

To have a closed caption file automatically generated when you upload a video, follow this step.

- In the **Asset Upload** dialog box, select **Automatically generate closed captions**. If you're generating a closed caption file, the file selector for closed caption files will be unavailable in the dialog box.

To manually upload a closed caption file when you upload a video, follow this step.

- In the **Asset Upload** dialog box, clear **Automatically generate closed captions**.

To upload regular audio or descriptive audio files for the video, use the file selector in the **Asset Upload** dialog box.

#### **NOTE**

Closed caption, regular audio, and descriptive audio assets can also be added after a video asset is uploaded. Go to **Media Library**, select the video asset, and select **Edit** to check it out. Then, in the properties pane for the video asset, upload the additional assets.

#### **Edit CC and audio transcript files**

CC and audio transcript files can be edited directly in the authoring tool. Video playback is available during editing.

To edit CC and audio transcript files, follow these steps.

1. Go to **Media Library**, and select the file name of the video asset. The closed caption and transcript content editor appears.
2. Select **Edit**.
3. Edit the closed caption or transcript text.
4. When you've finished, select **Save**, and then select **Finish editing**.
5. When you're ready to publish, select **Publish**.

#### **Set the Minimum Age attribute**

A **Minimum Age** metadata attribute can be associated with video assets.

To set the **Minimum Age** attribute for a video asset, follow these steps.

1. Go to **Media Library**, and select the video asset.
2. Select **Edit**.
3. In the properties pane for the video asset, set the **Minimum Age** attribute.

#### **NOTE**

The properties pane is used only to set and store the metadata attribute value. Customized modules must be created to use this attribute for playback gating.

## Additional resources

[Accessibility in forms, products, and controls](#)

[Microsoft Accessibility Center](#)

[Dynamics 365 Accessibility Center](#)

[Compliance overview](#)



[Cookie compliance](#)

[Add a privacy policy page](#)

[Replace user IDs associated with tracked content changes](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cookie compliance

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes considerations for cookie compliance and the default policies that are included in Microsoft Dynamics 365 Commerce.

## Overview

Privacy is an important factor when tracking technologies that affect e-Commerce customers. Because of privacy compliance standards such as the General Data Protection Regulation (GDPR) in the European Union (EU), electronic privacy guidelines must be considered for any site that is active today. Because many e-Commerce sites are globally accessible by default, it's important that you review the compliance standards for your e-Commerce site.

To learn more about the basic principles that Microsoft uses for cookie compliance, visit the [Microsoft Trust Center](#). On that site, you can also get more information about areas of compliance and privacy.

The following table shows the current reference list of cookies placed by Dynamics 365 Commerce sites.

COOKIE NAME	USAGE
.AspNet.Cookies	Store Microsoft Azure Active Directory (Azure AD) authentication cookies for single sign-on (SSO). Stores encrypted user principal information (name, surname, email).
_msdyn365__cart_	Store cart ID used to obtain list of products added to cart instance.
_msdyn365__ucc_	Cookie compliance consent tracking.
ai_session	Detects how many sessions of user activity have included certain pages and features of the app.
ai_user	Detects how many people used the app and its features. Users are counted using anonymous IDs.
b2cru	Stores redirect URL dynamically.
JSESSIONID	Used by payment connector Adyen to store user session.
OpenIdConnect.nonce.*	Authentication
x-ms-cpim-cache:.*	Used for maintaining the request state.
x-ms-cpim-csrf	Cross-site request forgery (CSRF) token used for protection from CSRF.
x-ms-cpim-dc	Used to route requests to the appropriate production authentication server instance.

COOKIE NAME	USAGE
x-ms-cpim-rc.*	Used to route requests to the appropriate production authentication server instance.
x-ms-cpim-slice	Used to route requests to the appropriate production authentication server instance.
x-ms-cpim-ssorushmoreb2c.onmicrosoft.com_0	Used for maintaining the SSO session.
x-ms-cpim-trans	Used for tracking transactions (the number of open tabs authenticating against a business-to-consumer (B2C) site), including the current transaction.

## Site user cookie consent on an e-Commerce site

If an e-Commerce site feature or module uses a non-essential cookie, a site user's consent must be obtained before the cookie is tracked. To allow site users to provide cookie consent on the e-Commerce site, a site author must add and configure a cookie consent module in the page's header module to ensure that the consent is prompted for and received. Site user consent must be given before a feature or module using a non-essential cookie can be rendered on a site page.

## Additional resources

[Accessibility features and capabilities](#)

[Compliance overview](#)

[Add a privacy policy page](#)

[Replace user IDs associated with tracked content changes](#)

[Cookie consent module](#)

[Header module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add a privacy policy page

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to add a privacy policy page to your site in Microsoft Dynamics 365 Commerce.

## Overview

Privacy compliance includes organizational measures that inform site users about how their data is collected and handled. Users can then decide how they want their personal data to be handled and can take appropriate action.

## Review the Microsoft privacy statement in Dynamics 365 Commerce

To review the Microsoft privacy statement while you're signed in to the Dynamics 365 Commerce authoring tools, select the **Help** button (?) in the upper-right corner, and then select **Privacy and cookies**. A new tab is opened that has a link to the [Microsoft privacy statement](#).

## Build a privacy policy page for your site

In Dynamics 365 Commerce, there are several ways to give users of your site access to your privacy policy. This section shows how to build a privacy policy page and then reference the page by using a footer fragment.

The guidance that follows is an example that shows how to build a generic privacy policy page for a Commerce site. You're responsible for designing and implementing a privacy policy page solution that best meets your company's legal requirements.

To start, in the authoring tools, go to the site that you want to build a privacy policy page for.

### Create a template

#### NOTE

If a template that can be used for the privacy policy page has already been created, skip ahead to the [Build a privacy policy page](#) section.

To create a template, follow these steps.

1. Go to **Templates**, and then select **New** to create a page template.
2. In the **New Template** dialog box, under **Template Name**, enter **Promo banner template**, and then select **OK**.
3. In the template, add any required modules to the required page slots. For guidance, hover over the red exclamation marks. (For example, the **HTML Head** slot might require a **Default External Script** module.)
4. In the **Body** slot, add a **Default Page** module.
5. In the **Default Page** module, in the **Main** slot, add a **Content Rich Block** module.
6. In the **Content Rich Block** module, add a **Content rich block item** module.
7. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.

### Build a privacy policy page

To build a privacy policy page, follow these steps.

1. Go to **Pages**, and then select **New** to create a page.

2. In the **Choose a template** dialog box, select the template for the privacy policy page.
3. Enter a page name and page URL, and then select **OK**.
4. In the **Main** slot of the page, add a **Content Rich Block** module.
5. In the **Content Rich Block** module, add a **Content rich block item** module.
6. In the properties pane for the **Content Rich Block** module, select **Add Data Source**, and then select **Rich Text Content**.
7. In the rich text editor, enter the content for the privacy policy page. Expand the rich text editor to full-screen mode as you require.
8. When you've finished entering content, select **Preview** to preview the page in the web browser.
9. Complete any remaining additions to the page and module properties.
10. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

To publish the URL for the privacy policy page, follow these steps.

1. Go to **URLs**, and select the URL for the privacy policy page.
2. Select **Publish** to publish the selected URL.

### Create a link to the privacy policy page in a footer

You can add a link to the privacy policy page to a fragment. In this way, you can share the link and update it across multiple site pages by referencing the fragment. This example shows how to add a link to the privacy policy page to a footer fragment.

To add a link to a footer fragment, follow these steps.

1. Go to **Fragments**, and then select **New** to create a page fragment.
2. In the **New fragment** dialog box, select the **Footer** module.
3. Under **Fragment name**, enter a name for the fragment, and then select **OK**.
4. In the **Footer category** slot, add a **Footer item** module.
5. In the properties pane on the right, select **Link text**.
6. In the **Link text** dialog box, enter the link text and link target of the privacy policy page, and then click **OK**.
7. To get the URL of the privacy policy page, go to **Pages**, go to the privacy policy page, and copy the URL from the properties pane.
8. Select **Save**, select **Finish editing** to check in the fragment, and then select **Publish** to publish it.
9. Preview the fragment, and test the link to the privacy policy page.

The fragment can now be referenced in the template for other site pages. When this fragment is referenced in the **Footer** module of a template, the link reference will appear on any pages that are built by using that template.

## Additional resources

[Compliance overview](#)

[Accessibility features and capabilities](#)

[Cookie compliance](#)

[Replace user IDs associated with tracked content changes](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Replace user IDs associated with tracked content changes

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to replace user IDs that are associated with tracked content changes in Microsoft Dynamics 365 Commerce site builder.

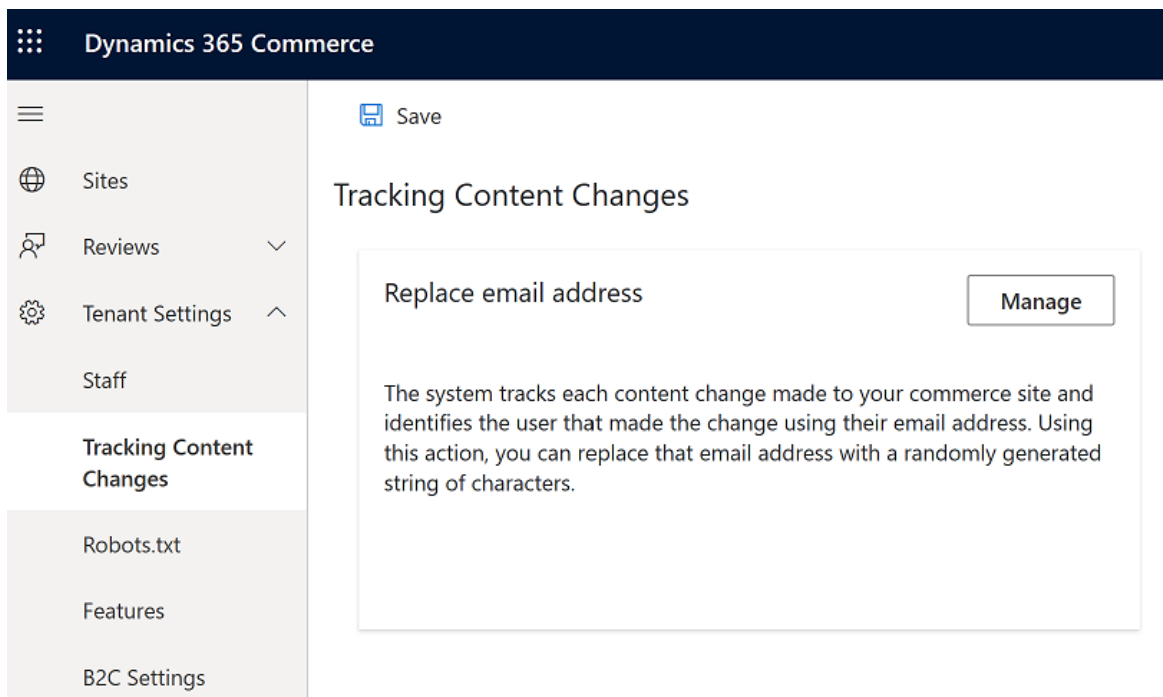
## Overview

In Dynamics 365 Commerce, the site builder authoring tool tracks changes that are made to items in the content management system (CMS). Therefore, a document change history can be shown to help teams track their efforts when they collaborate on content. To assign user identities to tracked changes, the system uses user IDs from the Azure Active Directory (Azure AD) identity management system. These user IDs are also the email addresses that are issued by Azure AD. Commerce system admins can replace user ID references in the change tracking history logs in site builder as they require.

## Replace a user ID in site builder

To replace a user ID in site builder, follow these steps.

1. Go to the **Home** page for your site.
2. In the left navigation pane, expand **Tenant Settings**, and then select **Tracking Content Changes**.



3. On the **Tracking Content Changes** page, select **Manage**.
4. In the **Replace email address** field, enter the user ID email address that should be removed from the change tracking logs, and then select **Replace**. (You can enter multiple email addresses before you select **Replace**.)



## Manage email address

### Replace email address

 user1@fabrikam.onmicrosoft.com

5. Select **OK**, and then select **Save**. A message box notifies you that the records for the user IDs that you entered have been updated.

#### **NOTE**

Site builder replaces every user ID email address with an anonymized, randomly generated string to remove all CMS references to the email address. This action affects only the history logs that are referenced in the specific e-Commerce environment (tenant) that is associated with the site builder instance.

## Additional resources

[Compliance overview](#)

[Accessibility features and capabilities](#)

[Cookie compliance](#)

[Add a privacy policy page](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module library overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic presents an overview of the Microsoft Dynamics 365 Commerce module library.

The Dynamics 365 Commerce module library is a collection of modules that can be used to build an e-Commerce website. Modules have both user interface (UI) aspects and functional behavior aspects.

Themes can be applied to the modules in the module library to change their look and feel. The themes use Cascading Style Sheets (CSS). A theme for a fictitious e-Commerce site that is named "Fabrikam" is provided as part of the module library and can be used as a reference.

## Module library modules

The following types of modules are provided in the module library:

- **Container module** – A container module is a simple module that acts as a host for other modules. It controls the layout of the modules that are inside it.
- **Marketing modules** – Marketing modules include content block, text block, video player, and carousel modules. All these modules can be used to showcase content. They can be put on any page and are driven by data from the content management system (CMS).
- **Header and footer modules** – Header and footer modules appear in the header and footer of all site pages. These modules can be configured as required through properties.
- **Search modules** – Products can be discovered by using the search module in the header. Search results appear on the search results page. Products can also be discovered on category pages, which are dedicated pages for each category that is supported in the channel navigation hierarchy. In addition, refiner modules can be used to further filter results on search results and category pages.
- **Product details page modules** – Product details pages use several modules to show product information. The buy box module lets customers view products and add them to the cart. Other modules, such as the tech specs module, show the product details. The ratings and reviews module can be used to view and provide reviews.
- **Buy online pick up in store module** – The buy online pick up in store module is integrated with Bing Maps. It can be used to find nearby stores where customers can pick up products that they have purchased.
- **Purchase modules** – Purchase modules include the cart module, which can be used to add items to the cart. The checkout module captures the shipping address, delivery options, and gift card, loyalty program, and credit card information, so that an order can be processed. After an order is placed, the order confirmation module can be used to show the confirmation details.
- **Account management modules** – The sign-in module lets customers sign in to an existing account, and the sign-up module lets them create a new account. After an account is created, the order history module can be used to view recent orders, and the order details module can be used to view order details.
- **Recommendations module** – Recommendations are shown by using the product placement module. This module supports algorithmic and editorial lists that can be showcased on any page.

## Additional resources

[Container module](#)

[Buy box module](#)

[Cart module](#)



Checkout module

Order confirmation module

Header module

Footer module

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Container module

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic covers container modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

A container module is a module that hosts other modules inside it. The primary purpose of a container module is to define, through the properties that are set for it, the layout of the modules that it contains. For example, those modules can appear side by side in a two-column, three-column, four-column, or six-column layout. They can also be limited to the width of the container, or they can fill the screen. A heading can also be added to every container module.

Three container modules are supported: container, container with 2-slots, and container with 3-slots. Modules of any type can be put inside these containers.

## NOTE

We recommend that you always put modules inside a container module, so that they can be limited to the width of the container.

## Examples of container modules in e-Commerce

- A site author wants a three-column layout, where three modules appear side by side. Therefore, the site author uses a container module of the container with 3-slots type.
- A site author wants a six-column layout, where six modules appear side by side. Therefore, the site author uses a container of the contain type that has six columns inside it.
- A site author wants to put a module on a page but doesn't want it to fill the screen. Therefore, the site author adds the module to a container module and sets the container's **Width** property to **Fit container**.

The following image shows an example of a container module that contains a carousel module in Commerce site builder. In this example, the **Width** property of the container module is set to **Fill Screen**.

The screenshot displays the Dynamics 365 Commerce site builder interface. The main workspace shows a 'Homepage' page with a 'Carousel container' module selected. The carousel displays a woman in a purple coat with the text 'NEW ARRIVALS' and 'Into the summer breeze with all new designer dresses'. The right-hand pane shows the properties for the 'Carousel container' module, with the 'Width' property set to 'Fill Screen' and the 'Container Layout' set to 'Stacked'. The left-hand pane shows the site structure, including 'Live site', 'Pages', 'Products', 'URLs', 'Layouts', 'Templates', 'Page Fragments', and 'Media Library'.

# Container module properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Heading text and heading tag ( <b>H1</b> , <b>H2</b> , <b>H3</b> , <b>H4</b> , <b>H5</b> , or <b>H6</b> )	An optional heading can be provided for the container. By default, the <b>H2</b> heading tag is used for the heading. However, the tag can be changed to meet accessibility requirements.
Width	<b>Fit container</b> or <b>Fill screen</b>	If the value is set to <b>Fit container</b> (the default value), the modules inside the container are limited to the width of the container. If the value is set to <b>Fill screen</b> , the modules aren't limited to the container width but can fill the screen.
Number of columns	<b>1</b> , <b>2</b> , <b>3</b> , <b>4</b> , <b>6</b> , or <b>12</b>	This property defines the number of columns in the container. A container can have up to 12 columns.

## Container with 2-slots

The container with 2-slots type is optimized for a two-column layout. This type of container has two slots to allow for a side-by-side view of the modules that are inside.

Additional properties can be used to optimize the layout for different view ports (mobile devices, tablets, computers, and so on). For every view port, the width of each column can be defined. The following column width settings are available:

- **75%/25%** – The first module has a column width of 75 percent, and the second module has a column width of 25 percent. A **25%/75%** option is also available.
- **50%/50%** – Both modules have equal column width.
- **67%/33%** – The first module has a column width of 67 percent, and the second module has a column width of 33 percent. A **33%/67%** option is also available.
- **100%** – Both modules have a full-column width. Therefore, the modules are vertically stacked in a single column. Although this single-column layout goes against intent of the container with 2-slots type, it might be preferable for some view ports (for example, extra-small view ports such as mobile devices).

### Container with 2-slots properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Heading text and heading tag	An optional can be provided for the container.
X-Small view port configuration	<b>25%/75%</b> , <b>75%/25%</b> , <b>50%/50%</b> , <b>67%/33%</b> , <b>33%/67%</b> , or <b>100%</b>	This property defines the layout for extra-small view ports.
Small view port configuration	<b>25%/75%</b> , <b>75%/25%</b> , <b>50%/50%</b> , <b>67%/33%</b> , <b>33%/67%</b> , or <b>100%</b>	This property defines the layout for small view ports, such as mobile devices.
Medium view port configuration	<b>25%/75%</b> , <b>75%/25%</b> , <b>50%/50%</b> , <b>67%/33%</b> , <b>33%/67%</b> , or <b>100%</b>	This property defines the layout for medium view ports, such as tablets.

PROPERTY NAME	VALUES	DESCRIPTION
Large view port configuration	25%/75%, 75%/25%, 50%/50%, 67%/33%, 33%/67%, or 100%	This property defines the layout for large view ports, such as computers.

## Container with 3-slots

The container with 3-slots modules type is optimized for a three-column layout.

Additional properties can be used to optimize the layout for different view ports. For every view port, the width of each column can be defined. The following column width settings are available:

- 33%/33%/33% – All three modules have equal column width.
- 50%/25%/25% – The first module has a column width of 50 percent, and each of the remaining two modules has a column width of 25 percent. 25%/50%/25% and 25%/25%/50% options are also available.
- 16%/16%/67% – Each of the first two modules has a column width of 16 percent, and the third module has a column width of 67 percent. 16%/67%/16% and 67%/16%/16% options are also available.

### Container with 3-slots properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Heading text and heading tag	An optional heading can be added to the container.
X-Small view port configuration	33%/33%/33%, 50%/25%/25%, 25%/50%/25%, 25%/25%/50%, 16%/16%/67%, 16%/67%/16%, or 67%/16%/16%	This property defines the layout for extra-small view ports.
Small view port configuration	33%/33%/33%, 50%/25%/25%, 25%/50%/25%, 25%/25%/50%, 16%/16%/67%, 16%/67%/16%, or 67%/16%/16%	This property defines the layout for small view ports, such as mobile devices.
Medium view port configuration	33%/33%/33%, 50%/25%/25%, 25%/50%/25%, 25%/25%/50%, 16%/16%/67%, 16%/67%/16%, or 67%/16%/16%	This property defines the layout for medium view ports, such as tablets.
Large view port configuration	33%/33%/33%, 50%/25%/25%, 25%/50%/25%, 25%/25%/50%, 16%/16%/67%, 16%/67%/16%, or 67%/16%/16%	This property defines the layout for large view ports, such as computers.

## Add a container module to a page

To add a container player module to a new page and set the required properties, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template name**, enter **Container template**, and then select **OK**.
3. In the **Body** slot, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Default Page** module, and then select **OK**.
5. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
6. Go to **Pages**, and select **New** to create a new page.

7. In the **Choose a template** dialog box, select the video player template that you created. Under **Page name**, enter **Container page**, and then select **OK**.
8. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add Module**.
9. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
10. In the property pane for the container module, set the **Number of columns** property to **1** and the **Width** property to **Fill container**.
11. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
12. In the **Add Module** dialog box, select the **Content block** module, and then select **OK**.
13. In the property pane for the content block module, configure the heading, image, and layout.
14. Select **Save**, and then select **Preview** to preview the page. You should see one feature module that fits within the width of the container module.
15. In the property pane for the container module, change the value of the **Number of columns** property to **3**.
16. Add two more content block modules to the container module, and configure them.
17. Select **Save**, and then select **Preview** to preview the page. You should now see three content block modules that appear side by side.
18. After you've achieved the layout that you want, select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Accordion module](#)

[Tab module](#)

[Carousel module](#)

[Text block module](#)

[Buy box module](#)

[Cart module](#)

[Checkout module](#)

[Header module](#)

[Footer module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Accordion module

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic covers accordion modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

Accordion modules are container-like modules that are used to organize the information or modules on a page by providing a collapsible drawer-like capability. An accordion module can be used on any page.

Inside every accordion module, one or more accordion item modules can be added. Each accordion item module represents a collapsible drawer. Inside every accordion item module, one or more modules can be added. There are no restrictions on the types of modules that can be added to an accordion item module.

The following image shows an example of an accordion module that is used to organize information on a store's frequently asked questions (FAQ) page.

The screenshot shows a 'Frequently asked questions' section with a heading and 'Open all | Close all' links. The accordion is expanded to show the first question: 'How do I make payments?'. The answer includes a list of accepted payment methods (Credit Cards: Visa, MasterCard; Gift Cards) and a note that cash, CODs, checks, money orders, and gift certificates are not accepted. Other questions listed include 'How can I track my order?', 'How do I know if my order online is secure?', 'What happens when my item is backordered?', 'What are the benefits of registering?', and 'Can I buy online and pick up in store?'.

## Accordion module properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Text	This property specifies an optional text heading for the accordion module.
Expand All	True or False	If the value is set to <b>True</b> , expand/collapse functionality is turned on, so that all items in the accordion module can be expanded and collapsed.

PROPERTY NAME	VALUES	DESCRIPTION
Interaction Style	<b>Independent</b> or <b>Expand one item only</b>	This property defines the style of interaction for accordion items. If the value is set to <b>Independent</b> , each accordion item can be expanded or collapsed independently. If the value is set to <b>Expand one item only</b> , only one item can be expanded at a time. As items are expanded, previously expanded items are collapsed.

## Accordion item module properties

PROPERTY NAME	VALUES	DESCRIPTION
Title	Text	This property specifies the title text for the accordion item module. By selecting the title region, users can expand or collapse the section.
Expand by default	<b>True</b> or <b>False</b>	If the value is set to <b>True</b> , the accordion item is expanded by default when the page is loaded.

## Add an accordion module to a FAQ page

To add an accordion module to a FAQ page and set its properties in site builder, follow these steps.

1. Go to **Pages**, and use the Fabrikam marketing template (or any template that has no restrictions) to create a new page that is named **Store FAQ**.
2. In the **Main** slot of the **Default page**, select the ellipsis (...), and then select **Add Module**.
3. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
4. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
5. In the **Add Module** dialog box, select the **Accordion** module, and then select **OK**.
6. In the property pane of the accordion module, select **Heading** next to the pencil symbol.
7. In the **Heading** dialog box, under **Heading Text**, enter **Frequently asked questions**. Then select **OK**.
8. In the property pane of the accordion module, select the **Show expand all** check box, and then, in the **Interaction style** field, select **Independent**.
9. In the **Accordion** slot, select the ellipsis (...), and then select **Add Module**.
10. In the **Add Module** dialog box, select the **Accordion item** module, and then select **OK**.
11. In the property pane of the accordion item module, under **Title**, enter title text (for example, **How do returns work?**).
12. In the **Accordion item** slot, select the ellipsis (...), and then select **Add Module**.
13. In the **Add Module** dialog box, select the **Text block** module, and then select **OK**.
14. In the property pane of the text block module, enter a paragraph of text (for example, **Returns must be processed via the call center. Contact 1-800-FABRIKAM for returns. Products have a 30-day return policy. Returns must be initiated within this time frame.**).
15. In the **Accordion** slot, add a few more accordion item modules. In each accordion item module, add a text block module that has content.
16. Select **Save**, and then select **Preview** to preview the page. The page will show an accordion module that has the content that you added.

17. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Container module](#)

[Tab module](#)

[Text block module](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Tab module

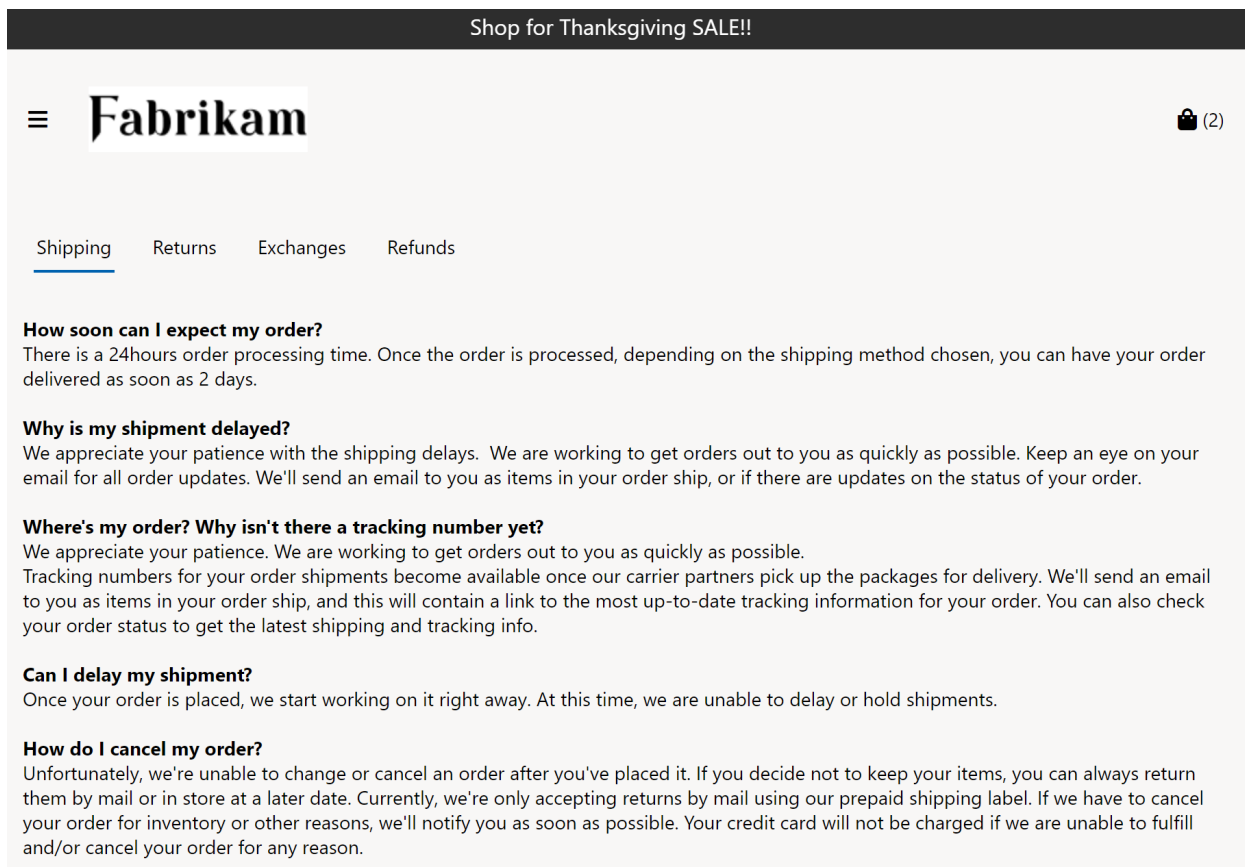
2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers tab modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

Tab modules are container-like modules that are used to organize the information on a site page on tabs. They can be used on any page where information must be presented on tabs.

In every tab module, one or more tab item modules can be added. Each tab item module represents a single tab. In each tab item module, one or more modules can be added. There are no restrictions on the types of modules that can be added to a tab item module.

The following image shows an example of a tab module on a site page. In this example, the **Shipping** tab is selected.



The screenshot shows a website header with a dark bar containing the text "Shop for Thanksgiving SALE!!". Below the header is the Fabrikam logo and a shopping cart icon with "(2)" items. A navigation bar contains four tabs: "Shipping" (which is underlined and selected), "Returns", "Exchanges", and "Refunds". The main content area displays several sections of text related to shipping, each with a bold heading:

- How soon can I expect my order?**  
There is a 24hours order processing time. Once the order is processed, depending on the shipping method chosen, you can have your order delivered as soon as 2 days.
- Why is my shipment delayed?**  
We appreciate your patience with the shipping delays. We are working to get orders out to you as quickly as possible. Keep an eye on your email for all order updates. We'll send an email to you as items in your order ship, or if there are updates on the status of your order.
- Where's my order? Why isn't there a tracking number yet?**  
We appreciate your patience. We are working to get orders out to you as quickly as possible. Tracking numbers for your order shipments become available once our carrier partners pick up the packages for delivery. We'll send an email to you as items in your order ship, and this will contain a link to the most up-to-date tracking information for your order. You can also check your order status to get the latest shipping and tracking info.
- Can I delay my shipment?**  
Once your order is placed, we start working on it right away. At this time, we are unable to delay or hold shipments.
- How do I cancel my order?**  
Unfortunately, we're unable to change or cancel an order after you've placed it. If you decide not to keep your items, you can always return them by mail or in store at a later date. Currently, we're only accepting returns by mail using our prepaid shipping label. If we have to cancel your order for inventory or other reasons, we'll notify you as soon as possible. Your credit card will not be charged if we are unable to fulfill and/or cancel your order for any reason.

## Tab module properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Text	This property specifies an optional text heading for the tab module.
Active Tab Index	Number	This property specifies the tab that should be active by default when a page is loaded. If no value is provided, the first tab item is active by default.

# Tab item module properties

PROPERTY NAME	VALUES	DESCRIPTION
Title	Text	This property specifies the title text for the tab item module.

## Add a tab module to a page

To add a tab module to a page and set the properties, follow these steps.

1. Use the Fabrikam marketing template (or any template that has no restrictions) to create a new page that is named **Store policies page**.
2. In the **Main** slot of the **Default page**, select the ellipsis (...), and then select **Add Module**.
3. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
4. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
5. In the **Add Module** dialog box, select the **Tab** module, and then select **OK**.
6. In the property pane of the tab module, select **Heading** next to the pencil symbol.
7. In the **Heading** dialog box, under **Heading Text**, enter heading text (for example, **Policies**). Then select **OK**.
8. In the **Tab** slot, select the ellipsis (...), and then select **Add Module**.
9. In the **Add Module** dialog box, select the **Tab item** module, and then select **OK**.
10. In the property pane of the tab item module, under **Title**, enter title text (for example, **Delivery**).
11. In the **Tab item** slot, select the ellipsis (...), and then select **Add Module**.
12. In the **Add Module** dialog box, select the **Text block** module, and then select **OK**.
13. In the property pane of the text block module, under **Rich text**, enter a paragraph of text.
14. In the **Tab** slot, add a few more tab item modules that have titles. In each tab item module, add a text block module that has content.
15. Select **Save**, and then select **Preview** to preview the page. The page will show a tab module that contains tab item modules have the content that you added.
16. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Container module](#)

[Accordion module](#)

[Text block module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Iframe module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the iframe module and describes how to add it to site pages in Microsoft Dynamics 365 Commerce.

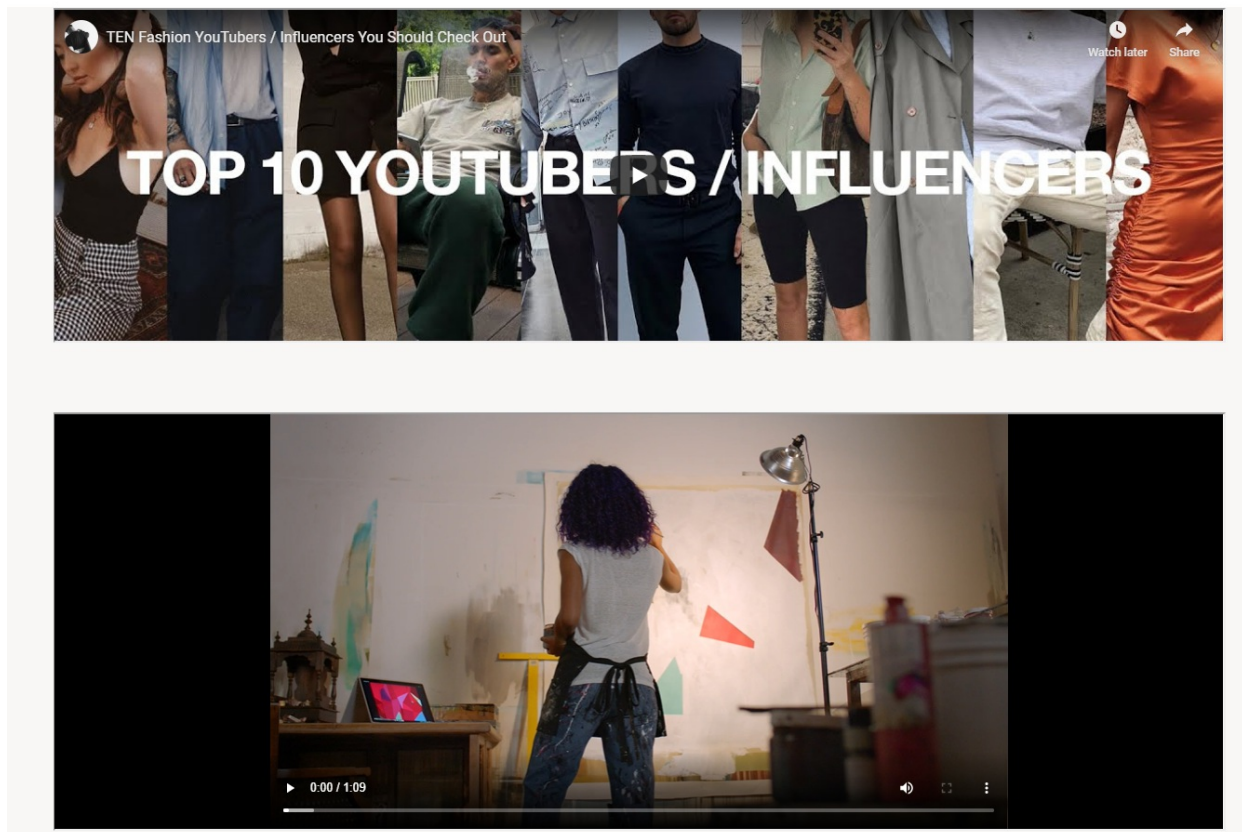
An iframe module provides an iframe (inline frame) that hosts external content on a site. For example, it can be used to host a YouTube video or a PDF file viewer on any site page.

An iframe module requires a target URL. It then hosts the content of the target page inside an HTML **iframe** element. External URLs must be on the allow list per the site's content security policy (CSP) directives. For iframe content, URLs should be allowed by using the **frame-ancestor** directive. For more information, see [Manage Content Security Policy \(CSP\)](#).

## NOTE

The iframe module is available in the Dynamics 365 Commerce 10.0.13 release.

The following image shows examples of iframe modules that showcase external videos on site pages.



## Iframe module properties

PROPERTY NAME	VALUE	DESCRIPTION
Heading	Text	The heading for the module.
Target URL	URL	The URL that is hosted in the module.

PROPERTY NAME	VALUE	DESCRIPTION
Height	Number or percentage	The height of the module, in pixels or as a percentage. For example, the value <b>100</b> will be treated as a number of pixels, and the value <b>100%</b> will be treated as a percentage.
Aria label	Text	An Accessible Rich Internet Applications (ARIA) label can be defined for accessibility purposes.

## Add an iframe module to a page

To add an iframe module to a page to show an external video, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template name**, enter **Marketing template**, and then select **OK**.
3. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
4. Go to **Pages**, and select **New** to create a new page.
5. In the **Choose a template** dialog box, select the **Marketing template** template. Under **Page name**, enter **Marketing page**, and then select **OK**.
6. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add Module**.
7. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
8. In the module's properties pane, set the **Width** value to **Fill Container**.
9. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
10. In the **Add Module** dialog box, select the **iframe** module, and then select **OK**.
11. In the module's properties pane, set the **Target URL** value to an external URL for a video.
12. Set other properties, such as **Heading** and **Height**, as you require.
13. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
14. Go to the marketing page on your site. You should see that the video is rendered in the iframe module.

## Additional resources

[Module library overview](#)

[Manage Content Security Policy \(CSP\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Promo banner module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers promo banner modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

Promo banner modules are used to show inline informational messages on a page. They can be used to show site-wide promotions that appear on all pages of an e-Commerce site.

Promo banner modules support a text message and a link. If multiple messages are added to a promo banner module, it becomes a rotating carousel banner that lets customers cycle through all the messages.

Promo banner modules are driven by data from the content management system (CMS) and can be put on any page.

## Usage examples of promo banners in e-Commerce

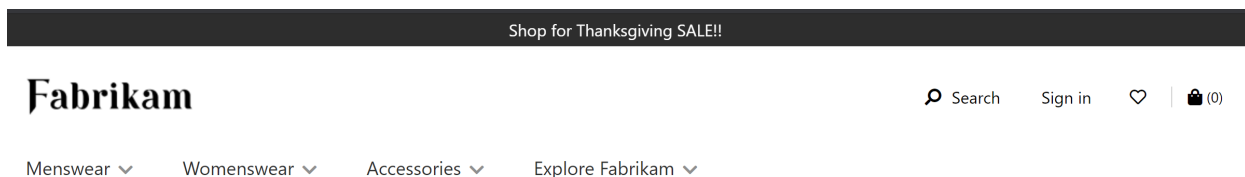
Promo banners can be used in the site header to show site-wide promotions or messages, as in the following examples.

"Annual sale ends in 10 days"

"Save big with back to school sale. Shop Now."

"Shop for Thanksgiving SALE!"

The following image shows an example of a promo banner.



## Promo banner module properties

PROPERTY NAME	VALUE	DESCRIPTION
Banner messages	Text and links	An array of text and links.
Autoplay	<b>True</b> or <b>False</b>	A value that indicates whether messages are automatically cycled through, if multiple messages are configured.
Slide transition interval	A number of milliseconds (ms)	The interval that is used to cycle through messages.
Allow dismiss	<b>True</b> or <b>False</b>	If the value is set to <b>True</b> , customers can dismiss the alert.

PROPERTY NAME	VALUE	DESCRIPTION
Show carousel flipper	<b>True or False</b>	A value that indicates whether the carousel flippers should be shown, so that customers can manually cycle through multiple banner items.
Text alignment	<b>Right, Left, or Center</b>	The text alignment in the promo banner module.
Link	A URL	The URL for an optional link.

## Add a promo banner module to a page

To add a promo banner module to a page and set the required properties, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template Name**, enter **Promo banner template**, and then select **OK**.
3. Under **Page Outline**, add a **Default page** module to the **Body** slot.
4. Select **Finish editing** to check in the template, and then select **Publish** to publish it.
5. Use the template that you just created to create a page that is named **Promo banner page**.
6. In the **Main** slot of the new page, add a container module.
7. In the pane on the right, set the **Width** value to **Fill Container**.
8. Under **Page Outline**, add a promo banner module to the container module.
9. In the settings for the banner module, add one or more banner messages. Each message can have text together with a link. You can edit the other properties to customize the module further.
10. Select **Save**, and then select **Preview** to preview the page. At the top of the page, you should see an alert that shows the text that you added.
11. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

### NOTE

A promo banner is typically used in the page header slot or a subheader slot.

## Additional resources

[Module library overview](#)

[Carousel module](#)

[Text block module](#)

[Content block module](#)

[Video player module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Carousel module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers carousel modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

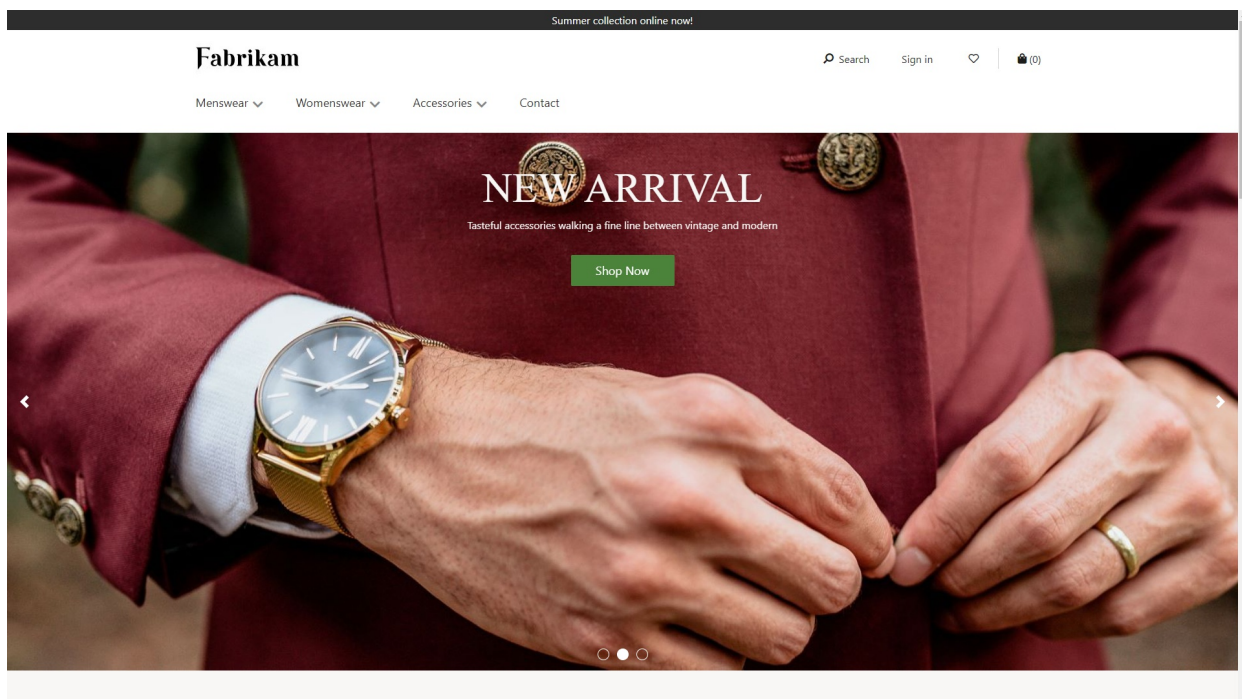
A carousel module is used to put multiple promotional items (including rich images) in a rotating carousel banner that customers can browse. For example, a retailer can use a carousel module on a home page to showcase multiple new products or promotions.

You can add content block modules inside a carousel module. The properties of the carousel module then define how those modules are rendered.

## Examples of carousel modules in e-Commerce

- A carousel that has multiple promotional modules inside it can be used on a home page.
- A carousel that has multiple promotional modules inside it can be used on a product details page.
- A carousel can be used on any marketing page to promote multiple promotions or products.

The following image shows an example of a carousel module that is used on a home page. This carousel module contains multiple content block items.



## Carousel module properties

PROPERTY NAME	VALUE	DESCRIPTION
---------------	-------	-------------

PROPERTY NAME	VALUE	DESCRIPTION
Autoplay	<b>True or False</b>	If the value is set to <b>True</b> , the transition between items inside the carousel occurs automatically. If the value is set to <b>False</b> , no transition occurs unless the customer uses the keyboard or mouse to move from one item to the next item.
Slide transition interval	A value in seconds	The interval for transitions between items.
Transition type	<b>Slide or Fade</b>	The transition effect between items.
Hide carousel flipper	<b>True or False</b>	If the value is set to <b>True</b> , the carousel flipper and sequence indicator are hidden.
Allow carousel dismiss	<b>True or False</b>	If the value is set to <b>True</b> , users can dismiss the carousel.

## Add a carousel module to a page

To add a carousel module to a new page and set the required properties, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template Name**, enter **Carousel template**, and then select **OK**.
3. In the **Body** slot, add a **Default page** module.
4. Select **Finish editing** to check in the template, and then select **Publish** to publish it.
5. Use the carousel template that you just created to create a page that is named **Carousel page**.
6. In the **Main** slot of the new page, add a container module.
7. In the pane on the right, set the **Width** value to **Fill Screen**.
8. Under **Page Outline**, add a carousel module to the container module.
9. Add a content block module to the carousel module. Set the properties of the content block module by providing **Heading**, **Link**, **Layout**, and other properties.
10. Add and configure another content block module.
11. Set additional properties for the carousel module as you require.
12. Select **Save**, and then select **Preview** to preview the page. The page should show a carousel that has two modules inside it (a hero module and a feature module). You can change additional properties for the carousel, hero, and feature modules to achieve the desired effect.
13. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Promo banner module](#)

[Text block module](#)

[Content block module](#)

[Video player module](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Text block module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers text block modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

A text block module is a module that is used to add textual content. This content can be informational or promotional.

Text block modules are driven by data from the content management system (CMS) and can be put on any page. They are stand-alone modules that don't depend on page context or any other modules.

## Examples of text block modules in e-Commerce

Text block modules can be used in the following ways:

- To showcase product features on a product details page.
- For informational purposes on any page. For example, they can explain the benefits of loyalty programs, describe shipping and return policies, answer frequently asked questions, or provide "about us" content.
- To add custom messages on a product details page. (for example, "Free shipping for orders over \$50").
- For disclaimers and contact details on product details pages, cart pages, checkout pages, and other pages (for example, "Shipping and returns are subject to store policies").

The following image shows an example of a text block module that is used on a home page.



## Text block module properties

PROPERTY NAME	VALUE	DESCRIPTION
Rich text	Rich text	Paragraph text. Some basic rich text capabilities are supported, such as bold, underlined, and italic text.
Custom class name	A Cascading Style Sheets (CSS) class name	The name of a custom CSS class that a developer provides to format this module. The class name should be defined in the theme pack.
Font size	<b>Small, Medium, Large, or X-Large</b>	The font size of the content.

## Add a text block module to a page

To add a text block module to a new page and set the required properties, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template name**, enter **Content template**.
3. In the **Body** slot, select the ellipsis (...), and then select **Add Module**.

4. In the **Add Module** dialog box, select the **Default page** module, and then select **OK**.
5. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
6. Go to **Pages**, and select **New** to create a new page.
7. In the **Choose a template** dialog box, select **Content template**. Under **Page name**, enter **Content page**, and then select **OK**.
8. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add Module**.
9. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
10. In the property pane for the container module, set the **Width** property to **Fill container**.
11. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
12. In the **Add Module** dialog box, select the **Text block** module, and then select **OK**.
13. In the property pane of the text block module, add text to the **Rich text** field.
14. Select **Save**, and then select **Preview** to preview the page.
15. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Promo banner module](#)

[Carousel module](#)

[Content block module](#)

[Video player module](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Content block module

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic covers content block modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

A content block module is used to market products or promotions through a combination of images and text. For example, a retailer can add a content block module to the home page of an e-Commerce site to promote a new product and attract the attention of customers.

A content block module is driven by data from the content management system (CMS). It's a stand-alone module that doesn't depend on any other modules on the page. A content block module can be put on any site page where a retailer wants to market or promote something (for example, products, sales, or features).

## Examples of content block module in e-Commerce

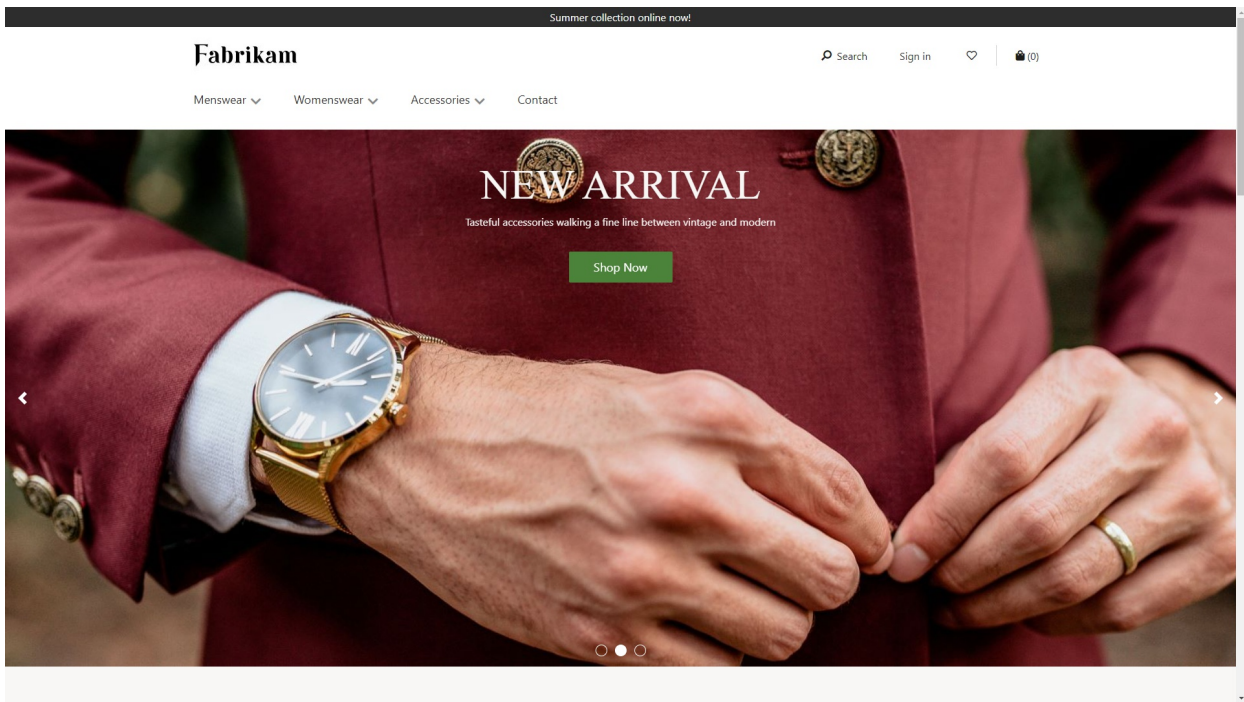
- A content block module can be used on the home page of an e-Commerce site to highlight promotions and new products.
- A content block module can be used on a product details page to showcase product information.
- Multiple content block modules can be put inside a carousel module to highlight multiple products or promotions.

## Content block modules and themes

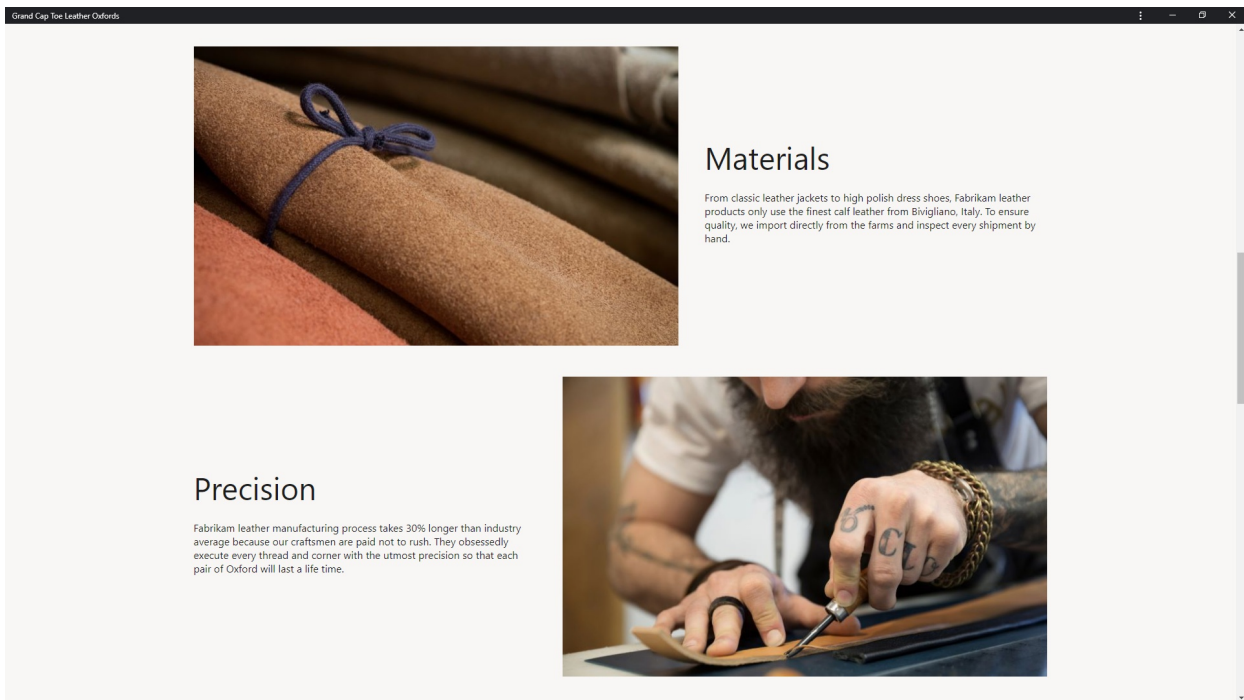
Content block modules can support various layouts and styles based on a theme. For example, the Fabrikam theme supports three layout variations of a content block module: hero, feature, and tile. The hero layout shows an image on the background with text overlay. The feature layout shows an image and text side by side. The tile layout allows multiple content blocks in a tile format.

In addition, the theme can expose different properties for each layout. A theme developer can build more layouts with more styles using the content block module.

The following image shows an example of a content block module with a hero layout.



The following image shows an example of a content block module with a feature layout.



## Content block module properties

PROPERTY NAME	VALUES	DESCRIPTION
Image	Image file	An image can be used to showcase a product or a promotion. An image can be uploaded to the image gallery, or an existing image can be used.
Heading	Heading text and heading tag (H1, H2, H3, H4, H5, or H6)	Every hero module can have a heading. By default, the H2 heading tag is used for the heading. However, the tag can be changed to meet accessibility requirements.

PROPERTY NAME	VALUES	DESCRIPTION
Paragraph	Paragraph text	Hero modules support paragraph text in rich text format. Some basic rich text capabilities are supported, such as bold, underlined, and italic text, and hyperlinks. Some of these capabilities can be overridden by the page theme that is applied to the module.
Link	Link text, link URL, Accessible Rich Internet Applications (ARIA) label, and <b>Open link in new tab</b>	Hero modules support one or more "call to action" links. If a link is added, link text, a URL, and an ARIA label are required. ARIA labels should be descriptive to meet accessibility requirements. Links can be configured so that they are opened on a new tab.

## Content block module properties exposed by the Fabrikam theme

PROPERTY NAME	VALUES	DESCRIPTION
Text placement	<b>Left, Right, Center</b>	This property defines the position of the text on the image. It only applies to the hero layout.
Text theme	<b>Light</b> or <b>Dark</b>	A color scheme can be defined for the text, based on the background image. For example, if the image has a dark background, a light theme can be applied to make the text more visible and to meet color contrast ratios for accessibility purposes. It only applies to the hero layout.
Image placement	<b>Left, Right</b>	This property specifies if the image should be to the left or right of the text. It only applies to the feature layout.

## Add a content block module to a new page

To add a hero module to a new page and set the required properties, follow these steps.

1. Go to **Templates**, and create a page template that is named **Content block template**.
2. In the **Main** slot of the default page, add a hero module.
3. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
4. Use the hero template that you just created to create a page that is named **Content block page**.
5. In the **Main** slot of the default page, select the ellipsis button (...), and then select **Add Module**.
6. In the **Add Module** dialog box, under **Select Modules**, select the hero module, and then select **OK**.
7. In the outline tree on the left, select the content block module.
8. In the properties pane on the right, select **Add an image**. Then either select an existing image or upload a new image.
9. Select **Heading**.
10. In the **Heading** dialog box, add the heading text, select the heading level, and then select **OK**.

11. Under **Rich Text**, add text as you require.
12. Select **Add Link**.
13. In the **Link** dialog box, add link text, a link URL, and an ARIA label for the link, and then select **OK**.
14. Select the **Hero** layout.
15. Select **Save**, and then select **Preview** to preview the page.
16. Select **Finish editing** to check in the template, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Promo banner module](#)

[Carousel module](#)

[Text block module](#)

[Video player module](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Video player module

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic covers video player modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

The video player module is used to support video playback. It can be added to any page, provided that video content is uploaded to and available in the content management system (CMS). The video player module supports the .mp4 media type.

## Video player module

The video player module can be used to showcase videos on an e-Commerce site. It supports all playback capabilities, such as play, pause, full-size mode, audio descriptions, and closed captions. The video player module also supports customization of closed captions to meet Microsoft accessibility standards. For example, you can customize the font size and background color.

The video player module also supports secondary audio tracks. When a video is uploaded to the CMS, a secondary audio track can also be uploaded. The video player module can then play the secondary audio track if a user selects it.

### Examples of video player modules in e-Commerce

- Instructional videos on product details pages or marketing pages
- Promotional videos or videos about policies on any marketing page
- Marketing videos that highlight product features on product details pages or marketing pages

The following image shows an example of a video player module on a home page.



### Video player module properties



PROPERTY NAME	VALUE	DESCRIPTION
Auto play	<b>True or False</b>	When the value is set to <b>True</b> , the video is automatically played.
Mute	<b>True or False</b>	When the value is set to <b>True</b> , the audio is muted. For this player, the default value is <b>False</b> . In the Chrome browser, autoplay videos are muted by default, and the audio is played only if the user manually plays the video.
Loop	<b>True or False</b>	When the value is set to <b>True</b> , the video is repeated in a loop.
Media	Video file path and name	The video file that is played in the video player.
Play fullscreen	<b>True or False</b>	When the value is set to <b>True</b> , the video is played in full-screen mode.
Play pause trigger	<b>True or False</b>	When the value is set to <b>True</b> , a play/pause button is shown on the video.
Video player controls	<b>True or False</b>	When the value is set to <b>True</b> , all video player controls are shown. These controls include play and pause buttons, a progress indicator, and closed caption options.
Hide poster image	<b>True or False</b>	A video can have a poster frame. When the value of this property is set to <b>True</b> , the poster frame is hidden.
Mask level	A number from <b>0</b> through <b>100</b>	The mask that is applied to the video for styling.

## Add a video player module to a page

### NOTE

Before you create a video player module, you must first upload a video to the Media Library.

To add a video player module to a new page and set the required properties, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template name**, enter **Video player template**, and then select **OK**.
3. In the **Body** slot, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Default Page** module, and then select **OK**.
5. In the **Main** slot of the **Default Page** module, select the ellipsis (...), and then select **Add Module**.
6. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
7. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.

8. In the **Add Module** dialog box, select the **Video player** module, and then select **OK**.
9. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
10. Go to **Pages**, and select **New** to create a new page.
11. In the **Choose a template** dialog box, select the video player template that you created. Under **Page name**, enter **Video player page**, and then select **OK**.
12. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add Module**.
13. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
14. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
15. In the **Add Module** dialog box, select the **Video player** module, and then select **OK**.
16. In the property pane of the video player module, select **Add a video**.
17. In the **Media Picker** dialog box, select a video, and then select **Upload new media item**.
18. In File Explorer, select a video file, and then select **Open**.
19. In the **Upload Media Item** dialog box, enter a title and other information as needed, and then select **OK**.
20. In the **Media Picker** dialog box, select **Close**.
21. Select **Save**, and then select **Preview** to preview the page. You should see the video module on the page. You can change additional settings to customize the behavior of the module.
22. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Promo banner module](#)

[Carousel module](#)

[Text block module](#)

[Content block module](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Header module

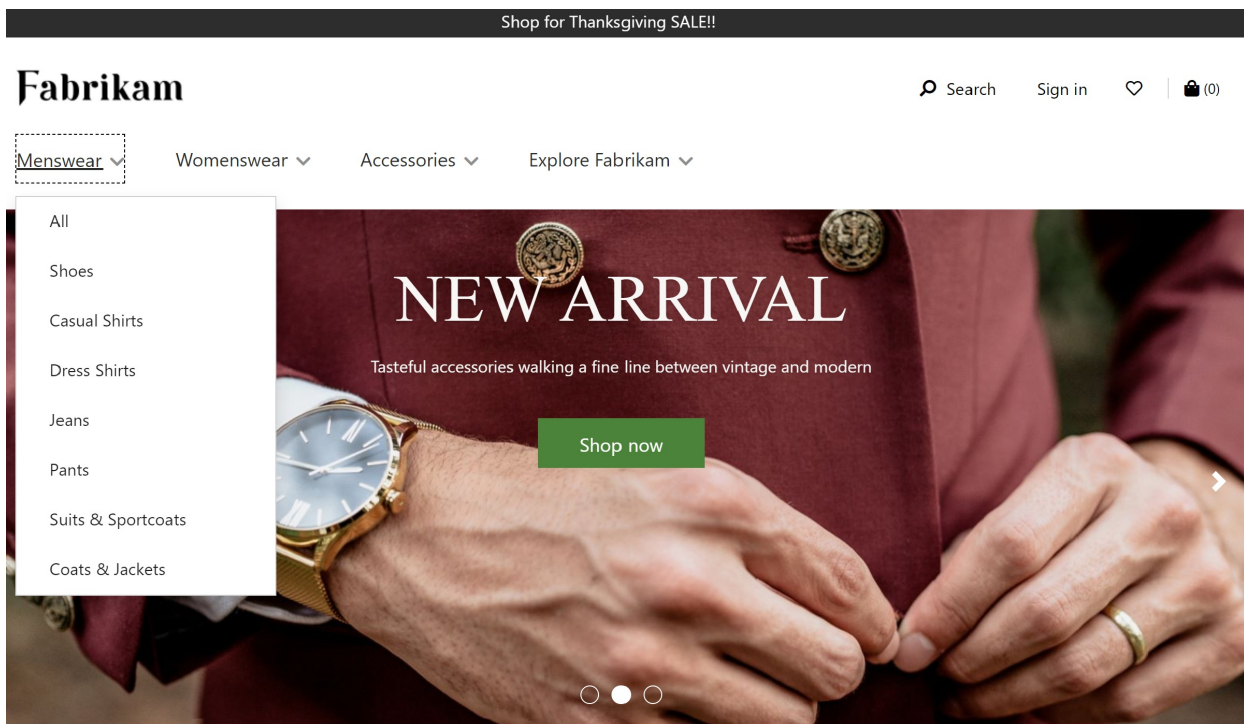
2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic covers header modules and describes how to create page headers in Microsoft Dynamics 365 Commerce.

In Dynamics 365 Commerce, a page header is configured as a page fragment that includes the header, promo banner, and cookie consent modules.

The header module includes a site logo, links to the navigation hierarchy, links to other pages on the site, a cart icon module, a wishlist symbol, sign-in options, and the search bar. A header module is automatically optimized for the device that the site is being viewed on (in other words, for a desktop device or a mobile device). For example, on a mobile device, the navigation bar is collapsed into a **Menu** button (which is sometimes referred to as a *hamburger menu*).

The following image shows an example of a header module on a home page.



## Properties of a header module

A header module supports **Logo image**, **Logo link**, and **My account links** properties.

The **Logo image** and **Logo link** properties are used to define a logo on the page. For more information, see [Add a logo](#).

The **My account links** property can be used to define account pages that the site owner wants to show quick links for in the header.

## Modules that are available within a header module

The following modules can be used in a header module:

- **Navigation menu** – The navigation menu represents the channel navigation hierarchy and other static navigation links. For more information, see [Navigation menu module](#).

- **Search** – The search module lets users enter search terms to search for products. The URL of the default search page and the search query parameters must be provided at **Site Settings > Extensions**. The search module has properties that let you suppress the search button or label as you require. The search module also supports auto-suggest options, such as product, keyword, and category search results.
- **Cart icon** - The cart icon module represents the cart icon, which shows the number of items in the cart at any given time. For more information, see [Cart icon module](#).
- **Site selector** - The site selector module lets users browse across different predefined sites, based on market, regions, and locales. For more information, see [Site selector module](#).
- **Store selector** - The store selector module can be included in a header module's store selector slot. It lets users browse and find nearby stores. Users can also specify a preferred store. That store will then be shown in the header. When the store selector module is included in the header module, its **Mode** property must be set to **Find stores**. For more information, see [Store selector module](#).

#### NOTE

- Support for using the cart icon module in header modules is available in the Dynamics 365 Commerce 10.0.11 release.
- Support for using the site selector module in header modules is available in the Dynamics 365 Commerce 10.0.14 release.
- Support for using the store selector module in header modules is available in the Dynamics 365 Commerce 10.0.15 release.

## Create a header fragment for a page

To create a header fragment, follow these steps.

1. Go to **Fragments**, and select **New** to create a new fragment.
2. In the **New fragment** dialog box, select the **Container** module, enter a name for the fragment, and then select **OK**.
3. Select the **Default container** slot, and then, in the properties pane on the right, set the **Width** property to **Fill Screen**.
4. In the **Default container** slot, select the ellipsis (...), and then select **Add Module**.
5. In the **Add Module** dialog box, select the **Cookie consent**, **Header**, and **Promo banner** modules, and then select **OK**.
6. In the properties pane of the **Promo banner** module, select **Add Message**, and then select **Message**.
7. In the **Message** dialog box, add text and links for the promotional content, and then select **OK**.
8. In the properties pane of the **Cookie consent** module, add and configure text and a link to the site privacy page.
9. In the **Navigation menu** slot of the header module, select the ellipsis (...), and then select **Add Module**.
10. In the **Add Module** dialog box, select the **Navigation menu** module, and then select **OK**.
11. In the property pane for the navigation menu module, under **Source for navigation menu**, select **Retail Server**.
12. In the property pane for the navigation menu module, under **Static menu items**, select **Add Menu item**, and then select **Menu item**.
13. In the **Menu item** dialog box, under **Menu Item Text** enter "Contact."
14. In the **Menu item** dialog box, under **Menu Item Link target** select **Add a link**.
15. In the **Add a link** dialog box, select the URL for the site's "Contact" page, and then select **OK**.
16. In the **Menu item** dialog box, select **OK**.
17. In the **Search** slot of the header module, select the ellipsis (...), and then select **Add Module**.

18. In the **Add Module** dialog box, select the **Search** module, and then select **OK**.
19. In the property pane for the search module, configure the properties as needed.
20. In the **Cart icon** slot of the header module, select the ellipsis (...), and then select **Add Module**.
21. In the **Add Module** dialog box, select the **Cart icon** module, and then select **OK**.
22. In the property pane for the cart icon module, configure the properties as needed. If you want the cart icon to display a cart summary (also known as a mini cart) when users hover over it, select **Show mini cart**.
23. Select **Save**, select **Finish editing** to check in the fragment, and then select **Publish** to publish it.

To help ensure that a header appears on every page, follow these steps on every page template that is created for the site.

1. In the **Header** slot of the **Default page** module, add the footer fragment that you created.
2. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Container module](#)

[Cart icon module](#)

[Promo banner module](#)

[Navigation Menu module](#)

[Cookie consent](#)

[Footer module](#)

[Site selector module](#)

[Store selector module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Footer module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers footer modules and describes how to create them in Microsoft Dynamics 365 Commerce.

The footer module is a special container that is used to host the modules that appear in the page footer. For example, it can include links to various pages across the site, such as **Contact Us** and **Store Policies** pages.

The following image shows an example of a footer module on a site page.

## Customer Service

FAQ  
Returns & refunds  
Terms and conditions  
Privacy Policy

## Fabrikam Store

Store locations  
Store hours  
Store events  
Fabrikam store support

## About us

Our story  
Careers with Fabrikam  
News

## Follow us



## Footer module properties

Like most containers, a footer module supports properties for the heading and the width. It also supports the addition of multiple footer category modules. Each footer category module that is added is rendered as a column in the footer module.

## Modules available in a footer module

**Footer items** – A footer items module can contain a heading, an image, and a link. The heading can be used either alone or in combination with an image and a link. Every link in the footer can be configured so that it has just text (for example, "Contact Us" and "Privacy" links), or so that it has both text and an image (for example, social media links).

**Back to top** – A back to top module provides a link for quick navigation to the top of the page. A destination is required. The default destination value is #, which takes the user to the top of the page.

## Create a footer module

1. Go to **Fragments**, and select **New** to create a new fragment.
2. In the **New fragment** dialog box, select the **Container** module, enter a name for the fragment, and then select **OK**.
3. In the **Default container** slot, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Footer category** module, and then select **OK**.
5. In the **Footer category** slot, select the ellipsis (...), and then select **Add Module**.
6. In the **Add Module** dialog box, select the **Footer item** module, and then select **OK**.
7. Select the **Footer item** slot, and then, in the properties pane on the right, configure the heading, link and link text, and image as needed.
8. To add more footer items, repeat steps 5 through 7 for each.
9. To add a "back to top" link to your footer, select the ellipsis (...) in the **Footer category** slot, and then select **Add Module**.

10. In the **Add Module** dialog box, select the **Back to top** module, and then select **OK**.
11. Select the **Back to top** slot, and then, in the properties pane on the right, configure the text and other module properties as needed.
12. Select **Finish editing** to check in the fragment, and then select **Publish** to publish it.

To help guarantee that a header appears on every page, follow these steps on every page template that is created for the site.

1. In the **Footer** slot of the **Default page** module, add the footer fragment that you created.
2. Select **Finish editing** to check in the template, and then select **Publish** to publish it.

By adding the fragment to page templates, you help guarantee that the footer is rendered on every page.

## Additional resources

[Module library overview](#)

[Container module](#)

[Buy box module](#)

[Cart module](#)

[Checkout module](#)

[Order confirmation module](#)

[Header module](#)

[Footer module](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Buy box module

2/18/2021 • 6 minutes to read • [Edit Online](#)

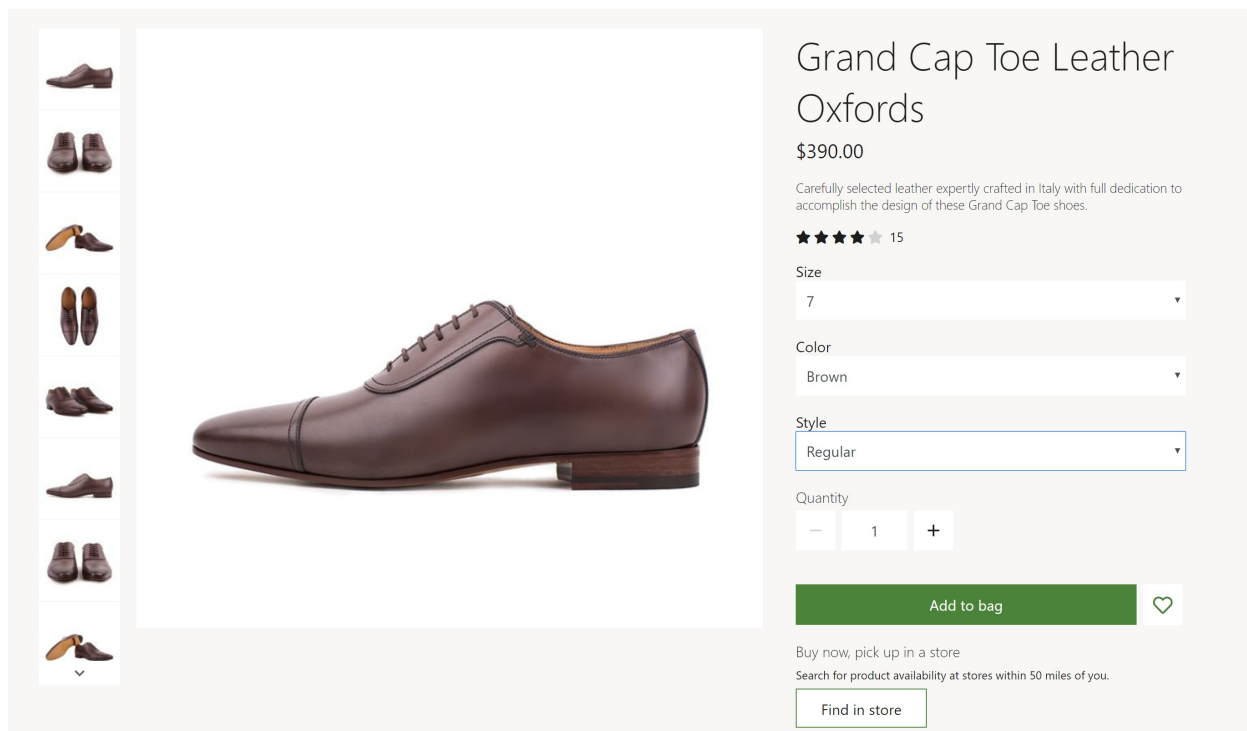
This topic covers buy box modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

The term *buy box* typically refers to the area of a product details page that is "above the fold," and that hosts all the most important information that is required to make a product purchase. (An area that is "above the fold" is visible when the page is first loaded, so that users don't have to scroll down to see it.)

A buy box module is special container that is used to host all the modules that are shown in the buy box area of a product details page.

The URL of a product details page includes the product ID. All the information that is required to render a buy box module is derived from this product ID. If a product ID isn't provided, the buy box module won't be rendered correctly on a page. Therefore, a buy box module can be used only on pages that have product context. To use it on a page that doesn't have product context (for example, a home page or a marketing page), you must do additional customizations.

The following image shows an example of a buy box module on a product details page.



## Buy box module properties and slots

On a product details page, a buy box is divided into two regions: a media region on the left and a content region on the right. By default, the ratio of the width of the media region column to the width of the content region column is 2:1. On mobile devices, the two regions are stacked so that one region appears below the other region. Themes can be used to customize the column widths and stacking rank.

A buy box module renders the title, description, price, and ratings of a product. It also lets customers select product variants that have different product attributes, such as size, style, and color. When a product variant is selected, other properties in the buy box (for example, the product description and images) are updated to reflect the variant information.



A quantity selector is provided, so that customers can specify the quantity of items to purchase. The maximum quantity that can be purchased can be defined in the site settings.

From the buy box, customers can also perform actions such as adding a product to the cart, adding a product to their wishlist, and selecting a pickup location. These actions can be performed on a product or a product variant. To add a product to a wishlist, the customer must be signed in.

Themes can be used to remove or change the order of buy box product properties and action controls.

## Module properties

- **Heading tag** – This property defines the heading tag for the product title. If the buy box is at the top of the page, this property should be set to **h1** to meet accessibility standards.
- **Enable "shop similar looks" recommendations** - This property allows the buy box to show links to products that look similar to the currently viewed item. This feature is available in Commerce release 10.0.13 and later.

## Modules that can be used in a buy box module

- **Media gallery** – This module is used to showcase images of a product on a product details page. For more information about this module, see [Media gallery module](#).
- **Store selector** – This module shows a list of nearby stores where an item is available for pickup. It lets users enter a location to find stores that are nearby. For more information about this module, see [Store selector module](#).
- **Social share** - This module can be added to the buy box to allow users to share product information on social media. For more information, see [Social share module](#).

## Buy box module settings

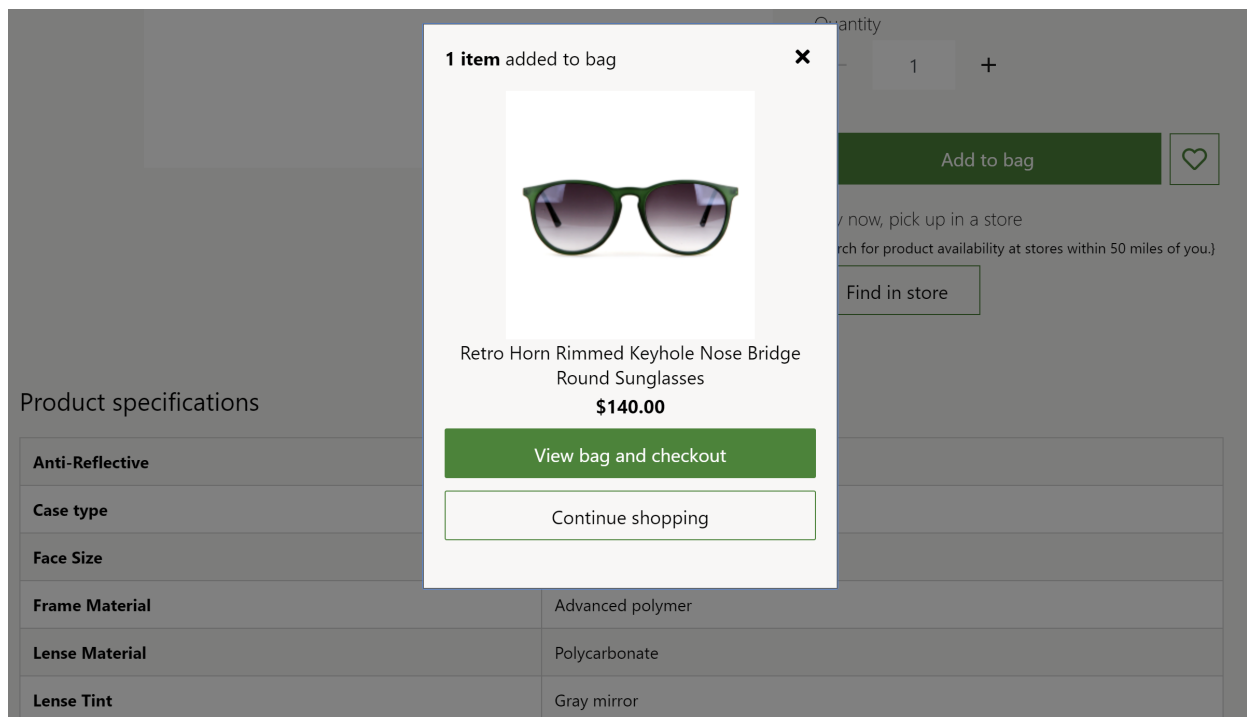
The following buy box module settings can be configured at **Site Settings > Extensions**:

- **Cart line quantity limit** – This property is used to specify the maximum number of each item that can be added to the cart. For example, a retailer might decide that only 10 of each product can be sold in a single transaction.
- **Inventory** – For information about how to apply inventory settings, see [Apply inventory settings](#).
- **Add product to cart** - This property is used to specify the behavior after an item is added to the cart. The possible values are **Navigate to cart page**, **Do not navigate to cart page**, and **Show notification**. When the value is set to **Navigate to cart page**, users are sent to the cart page after they add an item. When the value is set to **Do not navigate to cart page**, users aren't sent to the cart page after they add an item. When the value is set to **Show notification**, users are shown a confirmation notification and can continue to browse on the product details page.

### IMPORTANT

The **Add product to cart** site settings are available in the Dynamics 365 Commerce 10.0.11 release. If you are updating from an older version of Dynamics 365 Commerce, you must manually update the appsettings.json file. For instructions on updating the appsettings.json file, see [SDK and module library updates](#).

The following image shows an example of an "added to cart" confirmation notification on the Fabrikam site.



## Commerce Scale Unit interaction

The buy box module retrieves product information by using Commerce Scale Unit application programming interfaces (APIs). The product ID from the product details page is used to retrieve all information.

## Add a buy box module to a page

To add a buy box module to a new page and set the required properties, follow these steps.

1. Go to **Fragments**, and select **New** to create a new fragment.
2. In the **New fragment** dialog box, select the **Buy box** module.
3. Under **Fragment name**, enter the name **Buy box fragment**, and then select **OK**.
4. In the **Media Gallery** slot of the buy box module, select the ellipsis (...), and then select **Add Module**.
5. In the **Add Module** dialog box, select the **Media gallery** module, and then select **OK**.
6. In the **Store selector** slot of the buy box module, select the ellipsis (...), and then select **Add Module**.
7. In the **Add Module** dialog box, select the **Store selector** module, and then select **OK**.
8. Select **Save**, select **Finish editing** to check in the fragment, and then select **Publish** to publish it.
9. Go to **Templates**, and select **New** to create a new template.
10. In the **New Template** dialog box, under **Template name**, enter **PDP template**, and then select **OK**.
11. In the **Body** slot, select the ellipsis (...), and then select **Add Module**.
12. In the **Add Module** dialog box, select the **Default Page** module, and then select **OK**.
13. In the **Main** slot of the default page, select the ellipsis (...), and then select **Add fragment**.
14. In the **Select fragment** dialog box, select the **Buy box fragment** fragment that you created earlier, and then select **OK**.
15. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
16. Go to **Pages**, and select **New** to create a new page.
17. In the **Choose a template** dialog box, select the **PDP template** template. Under **Page name**, enter **PDP page**, and then select **OK**.
18. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add fragment**.
19. In the **Select fragment** dialog box, select the **Buy box fragment** fragment that you created earlier, and then select **OK**.

20. Save and preview the page. Add the **?productid= <product id>** query string parameter to the URL of the preview page. In that way, the product context is used to load and render the preview page.
21. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it. A buy box should appear on the product details page.

## Additional resources

[Module library overview](#)

[Store selector module](#)

[Media gallery module](#)

[Container module](#)

[Cart module](#)

[Checkout module](#)

[Order confirmation module](#)

[Header module](#)

[Footer module](#)

[Social share module](#)

[Calculate inventory availability for retail channels](#)

[SDK and module library updates](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Media gallery module

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic covers media gallery modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

Media gallery modules show one or more images in a gallery view. Media gallery modules support thumbnail images, which can be arranged either horizontally (as a row below the image) or vertically (as a column next to the image). Media gallery modules also provide capabilities that enable images to be zoomed (magnified) or viewed in full-screen mode. To be rendered in a media gallery module, an image must be available in the Commerce site builder Media Library. Currently, media gallery modules support only images.

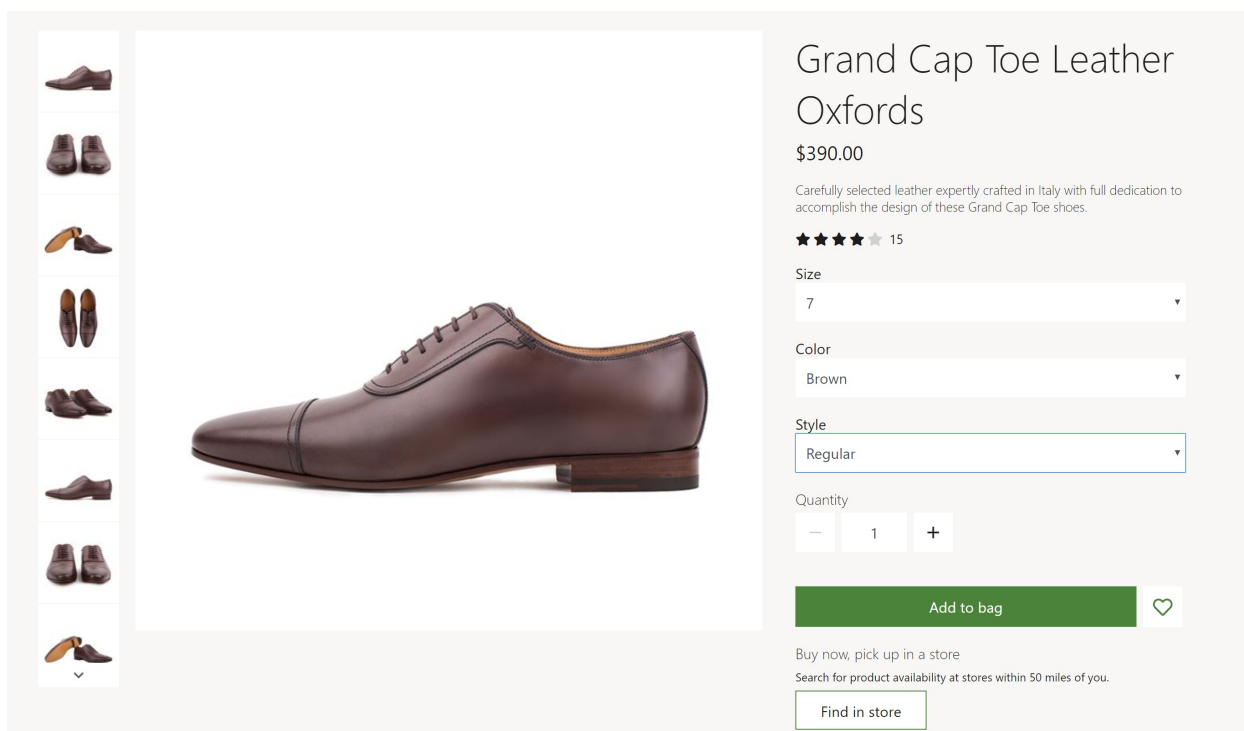
In the default mode, a media gallery module uses the product ID that is available from the page context of a product details page (PDP) to render the corresponding product images. In Commerce headquarters, a media file path must be defined for all products. Images should then be uploaded to the site builder Media Library according to the file path that was defined for the products in Commerce headquarters. These images include images for products and any product variants. For more information about how to upload images to site builder Media Library, see [Upload images](#).

Alternatively, a media gallery module can host a fully curated set of images on an image gallery page, where there are no dependencies on the product ID or page context. In this case, images must be uploaded to site builder Media Library and specified in site builder.

Here are some usage examples for media gallery modules:

- Rendering product images on a PDP
- Rendering product images on a product marketing page
- Showcasing a curated set of images on a marketing page, such as a gallery page

In the example in the following illustration, a buy box on a PDP hosts product images by using a media gallery module.



**Grand Cap Toe Leather Oxfords**

\$390.00

Carefully selected leather expertly crafted in Italy with full dedication to accomplish the design of these Grand Cap Toe shoes.

★★★★★ 15

Size: 7

Color: Brown

Style: Regular

Quantity: 1

[Add to bag](#) [♥](#)

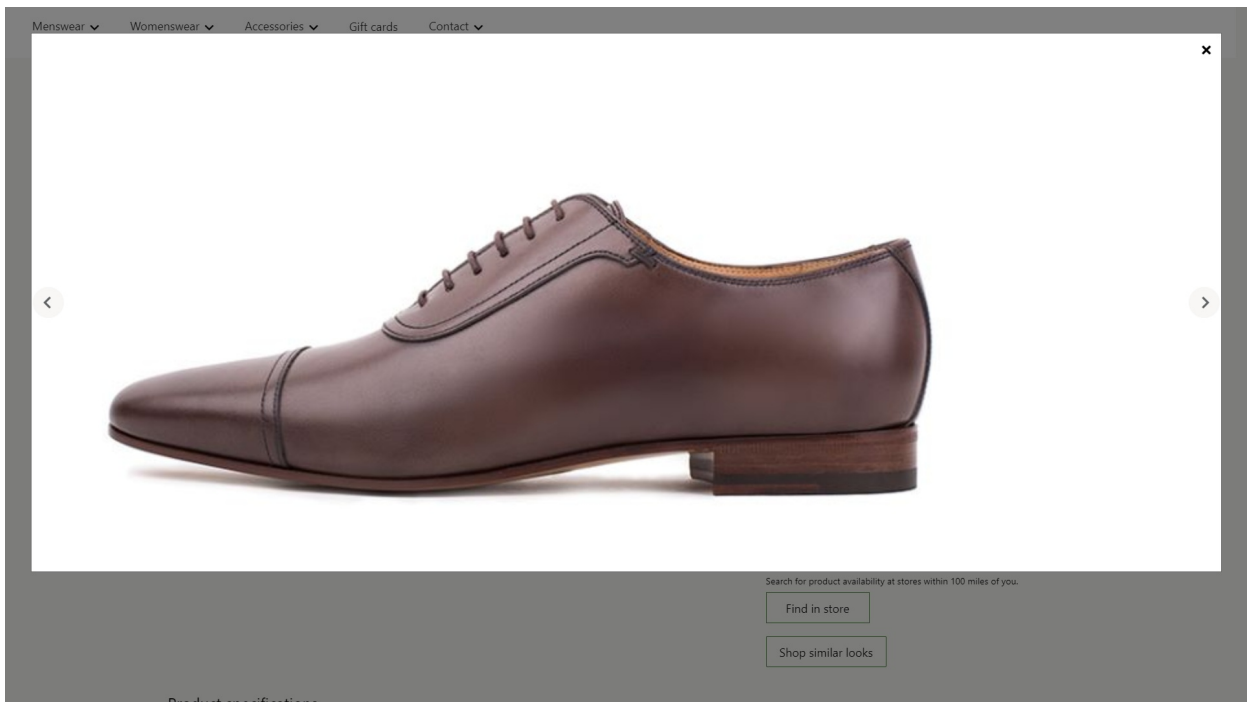
Buy now, pick up in a store  
Search for product availability at stores within 50 miles of you.

[Find in store](#)

## Media gallery properties

PROPERTY NAME	VALUES	DESCRIPTION
Image source	<b>Page context</b> or <b>Product ID</b>	The default value is <b>Page context</b> . If <b>Page context</b> is selected, the module expects the page to provide the product ID information. If <b>Product ID</b> is selected, the product ID for an image must be provided as the value of the <b>Product ID</b> property. This capability is available in Commerce version 10.0.12.
Product ID	A product ID	This property is applicable only if the value of the <b>Image source</b> property is <b>Product ID</b> .
Image zoom	<b>Inline</b> or <b>Container</b>	This property lets the user zoom images in the media gallery module. An image can be zoomed either inline or in a separate container next to the image. This capability is available in 10.0.12
Zoom scale	A decimal number	This property specifies the scale factor for zooming images. For example, if the value is set to 2.5, images are magnified 2.5 times.
Full screen	<b>True</b> or <b>False</b>	This property specifies whether images can be viewed in full-screen mode. In full-screen mode, images can be also be further magnified if the zoom capability is turned on. This capability is available in Commerce version 10.0.13.
Images	Images that are selected from site builder Media Library	In addition to being rendered from a product, images can be curated for a media gallery module. These images will be appended to any product images that are available. This capability is available in Commerce version 10.0.12.
Thumbnail orientation	<b>Vertical</b> or <b>Horizontal</b>	This property specifies whether thumbnail images should be shown in a vertical strip or a horizontal strip.

The following illustration shows an example of a media gallery module where the full-screen and zoom options are available.



The following illustration shows an example of a media gallery module that has curated images (that is, the specified images aren't dependent on the product ID or page context).



## Commerce Scale Unit interaction

When the image source is derived from the page context, the product ID from the PDP is used to retrieve the images. The media gallery module retrieves the image file path for products by using Commerce Scale Unit application programming interfaces (APIs). The images are then pulled from the Media Library so that they can be rendered in the module.

## Add a media gallery module to a page

To add a media gallery module to a marketing page, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template name**, enter **Marketing template**, and then select **OK**.
3. In the **Body** slot, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Default Page** module, and then select **OK**.
5. In the **Main** slot of the default page, select the ellipsis (...), and then select **Add Module**.
6. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
7. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
8. Go to **Pages**, and select **New** to create a new page.
9. In the **Choose a template** dialog box, select the **Marketing template** template. Under **Page name**, enter **Media gallery page**, and then select **OK**.
10. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add Module**.
11. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
12. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
13. In the **Add Module** dialog box, select the **Media gallery** module, and then select **OK**.
14. In the property pane for the media gallery module, under **Image source**, select **Productid**. Then, in the **Product id** field, enter a product ID.
15. Select **Save**, and then select **Preview** to preview the page. You should be able to see the images for the product in a gallery view.
16. To use only curated images, in the property pane, under **Image source**, select **Productid**. Then, under **Images**, select **Add an image** as many times as required to add images from the Media Library.
17. Set any additional properties that you want to set, such as **Image zoom**, **Zoom factor**, and **Thumbnails orientation**.
18. When you've finished, select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Buy box module](#)

[Container module](#)

[Upload images](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Ratings and reviews modules

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers ratings and reviews modules used on product details pages (PDPs) in Microsoft Dynamics 365 Commerce.

Ratings and reviews on e-Commerce websites help customers learn about products before they make a purchase decision, and are also a mechanism for collecting customer feedback about products.

Ratings are shown on product list pages, category list pages, search results pages, and other site pages.

Ratings histograms and product reviews are shown on PDPs. A **Write a review** button lets customers submit ratings and reviews for a product. These PDP features are controlled by ratings and review modules.

## Ratings and reviews modules on PDPs

Three modules show the ratings and reviews summary on PDPs:

- Write review module
- Product reviews list module
- Ratings histogram module

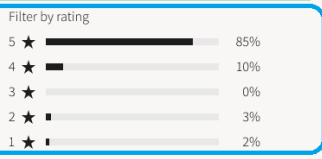
The following illustration shows what the ratings and reviews modules look like on a PDP.



Reviews

Write a review

Write review module



Rating histogram module

Reviews list module

★★★★★ 4/5 (126) Rating histogram - summary view Filter by: 5 stars Sort by: Popularity

★★★★★  
**You**  
 Verified purchase  
 July 13, 2018

**Review title**  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

**Fabrikam**  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Was this review helpful? (0) (0) Edit Delete

★★★★★  
**Customer Name**  
 Verified purchase  
 March 13, 2019

**Review title**  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Was this review helpful? (0) (0) Report

★★★★★  
**Customer Name**  
 Verified purchase  
 March 13, 2019

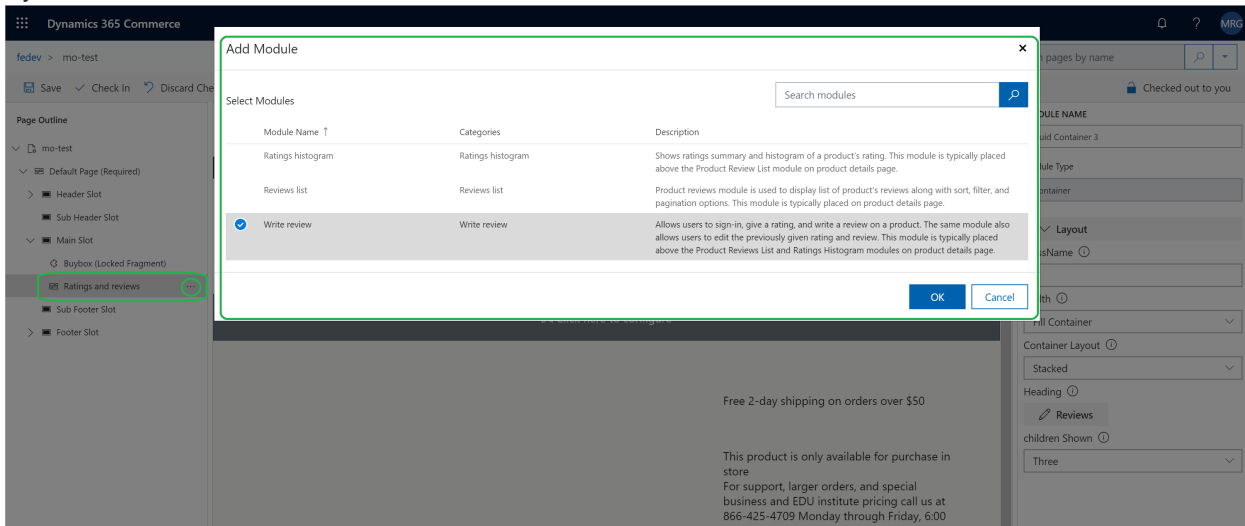
**Review title**  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Was this review helpful? (2) (0) Report

< Previous 1 2 3 4 5 6 ... 36 Next >

**TIP**  
 For information about how to optimize PDP templates and layouts so that you can share the configurations for ratings and reviews modules among multiple PDPs on your e-Commerce site, see [Templates and layouts overview](#).

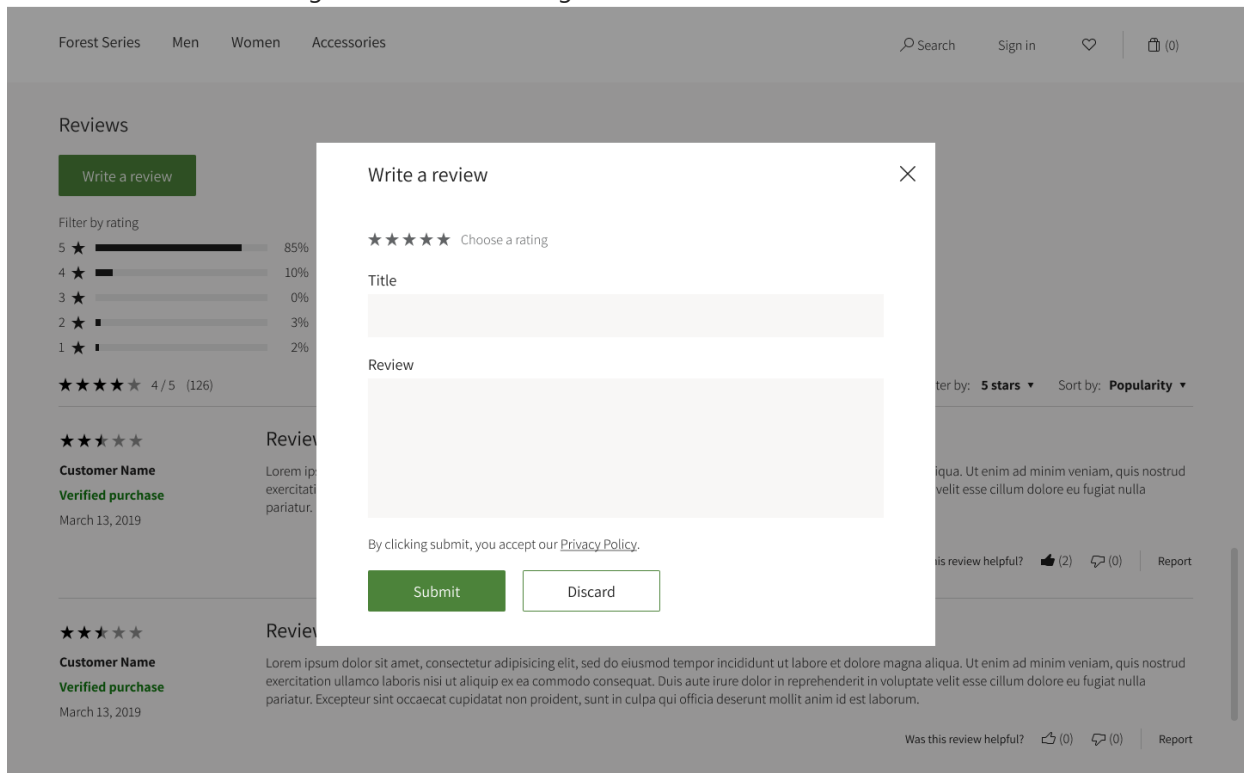
The following illustration shows how the **Add module** dialog box presents ratings and reviews modules in Dynamics 365 Commerce.



**Write review module**

The write review module includes a **Write a review** button that lets users sign in, assign a rating, and write a

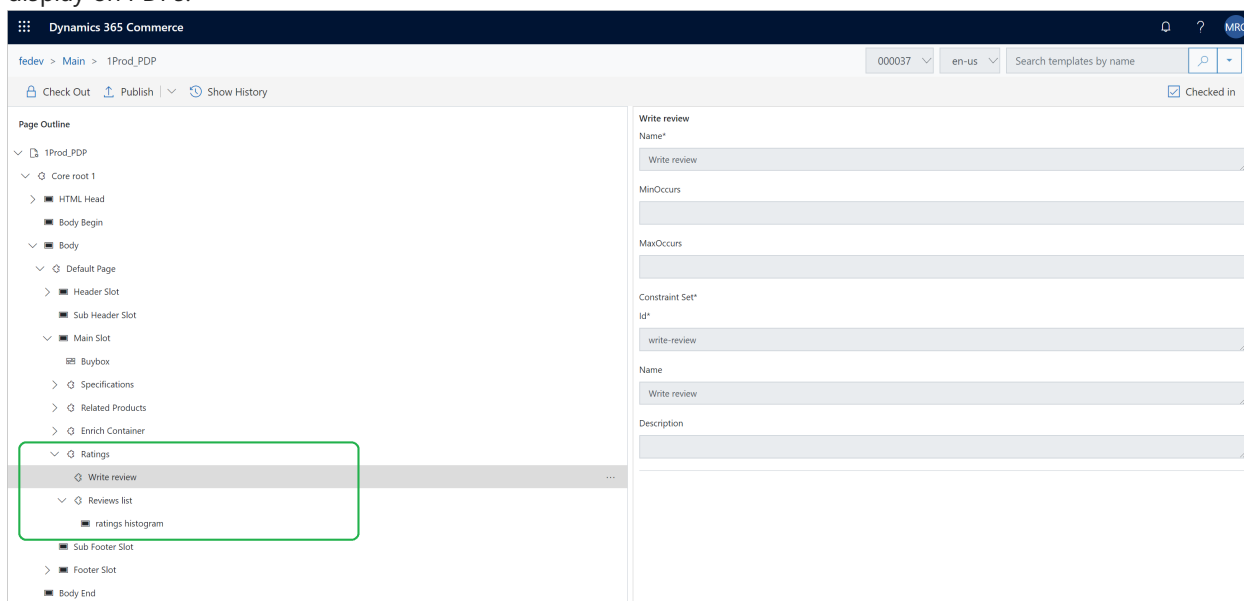
review of a product. This module also lets users edit a rating or review that they previously submitted. This module typically appears above the ratings histogram and product reviews list modules on a PDP. The following illustration shows the **Write a review** dialog box that appears when a customer selects **Write a review**. The customer can use this dialog box to submit a rating and a review.



The following table shows the write review module property that needs to be configured in the authoring tool. | Property name | Value | Property description | |-----|-----|-----| | Name | Write review | The name of the write review module. |

### Ratings histogram module

The ratings histogram module shows a ratings histogram. This module typically appears between the write review module and the product reviews list module on a PDP. The ratings histogram module requires no configuration. You just have to add the module in the PDP template. The following illustrations shows what a PDP template looks like in Dynamics 365 Commerce when ratings and reviews modules are configured for display on PDPs.



### Product reviews list module

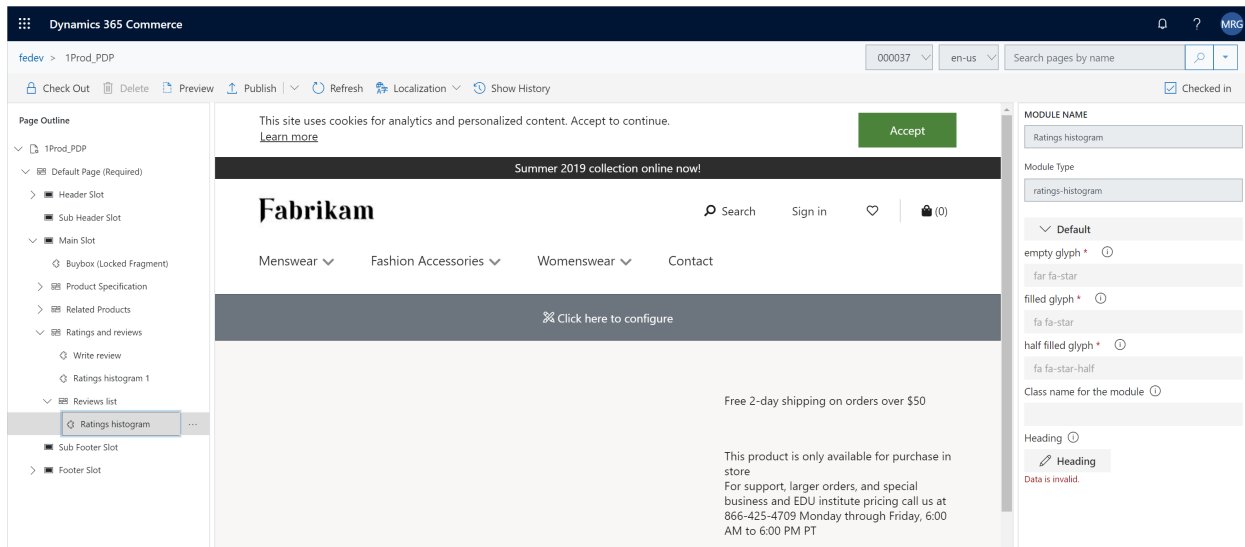
The product reviews list module shows a list of product reviews together with sort, filter, and pagination options. This module typically appears after the ratings histogram module on a PDP. The following table shows the

product reviews list module properties that need to be configured in the authoring tool.

PROPERTY NAME	VALUE	PROPERTY DESCRIPTION
Reviews shown on each page	10	The number of reviews that should be shown at a time on a PDP. <b>Next</b> and <b>Previous</b> buttons are included, so that users can move through the pages of reviews.

### Ratings histogram – Summary view

The product reviews list module includes a slot where you can add a ratings histogram module. The following illustration shows how you can add a ratings histogram module in the product reviews list module in Dynamics 365 Commerce.



## Additional resources

[Module library overview](#)

[Container module](#)

[Cart module](#)

[Checkout module](#)

[Order confirmation module](#)

[Header module](#)

[Footer module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Social share module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers social share modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

Social share modules allow users to share e-Commerce site page URLs on social media such as Facebook, Twitter, Pinterest, and LinkedIn. Site page URLs can also be shared via email. Social share modules are commonly used on product details pages (PDPs) to help users share product information.

Each social share module is a container for social share item modules. Each social share item module can be configured to point to a specific social media site. Integration with Facebook, Twitter, Pinterest, LinkedIn, and email is supported out of the box. When a site user selects a social media symbol, an HTML iframe is launched for the respective social media site. Within the iframe, the user can sign in and post the page content that they were viewing.

Each social media platform may track cookies, so this module requires site users to accept the cookie consent notification message. When cookie consent is not accepted, the module will be hidden on the page. For more information, see [Cookie compliance](#).

The following illustration highlights an example of a social share module used on a product details page.

Back to results Fabrikam Fashion / Womenswear / Coats / Camel Polka-dot Trench

**Camel Polka-dot Trench**  
\$170.00

This timeless Camel Polka-dot trench coat features a quilted liner, a tie waist, and a water resistant exterior.

★★★★★ 4

Size  
XS

Quantity  
- 1 +

Available

Add to bag

Buy now, pick up in a store  
Search for product availability at stores near you.

Find in store

Facebook, Twitter, LinkedIn

Shop similar looks

## Social share module properties

PROPERTY NAME	VALUE	DESCRIPTION
Caption	Text	This property specifies a caption for the module.
Orientation	Horizontal or Vertical	This property defines the layout orientation for the social media items.

# Social share item module properties

PROPERTY NAME	VALUE	DESCRIPTION
Social media	Facebook, Twitter, Pinterest, LinkedIn, Mail	A drop-down menu with a list of social media platforms.
Icon	Image	This will be the image that will be shown for the respective social media. As a best practice, refer to the social media platform's SDK for the recommended image to use for each platform.

## Add a social share module to a buy box module

To add a social share module to a buy box module, follow these steps.

1. In the Fabrikam site, select **Pages**, and then select the **DefaultPDP** page to open the product details page.
2. In the **Buybox (required)** slot, select the ellipsis (...), and then select **Add Module**.
3. In the **Add Module** dialog box, select the **Social Share** module, and then select **OK**.
4. In the **Social Share** slot, select the ellipsis (...), and then select **Add Module**.
5. In the **Add Module** dialog box, select the **SocialShare** module, and then select **OK**.
6. In the properties pane of the **SocialShare** module, under **Orientation**, select **Horizontal**. Add a caption as needed.
7. In the **SocialShare** slot, select the ellipsis (...), and then select **Add Module**.
8. In the **Add Module** dialog box, select the **SocialShareItem** module, and then select **OK**.
9. In the properties pane of the **SocialShareItem** module, under **Social Media**, select **Facebook**.
10. In the properties pane of the **SocialShareItem** module, under **Icon**, select **+ Add an image**.
11. In the **Media Picker** dialog box, select the Facebook logo image, and then select **OK**. If no Facebook logo image is present, select **Upload new media item** to upload one.
12. Add and configure additional **SocialShareItem** modules as needed.
13. Select **Save**, and then select **Preview** to preview the page. The page will show the social share module.
14. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Buy box module](#)

[Cookie compliance](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cart module

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic covers cart modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

A cart module shows the items that have been added to the cart before the customer proceeds to checkout. The module also shows an order summary and lets the customer apply or remove promotional codes.

The cart module supports signed-in checkout and guest checkout. It also supports a **Back to shopping** link. You can configure the route for this link at **Site Settings > Extensions > Routes**.

The cart module renders data based on the cart ID, which is a browser cookie available throughout the site.

The following image shows an example of a cart page on the Fabrikam site.


Summer collection online now!

## Fabrikam

Search Karen (1)

Menswear ▼ Womenswear ▼ Accessories ▼ Contact

### Shopping bag



**Grand Cap Toe Leather Oxfords**  
Size: 7.5  
Color: Brown  
Style: Regular

Quantity **\$390.00**

– 1 +

Ship to an address

Pick this up

[Move to Wishlist](#) [Remove](#)

#### Order summary

Promo Code

Enter promo code

---

Subtotal **\$390.00**

Estimated shipping **To be calculated**




Estimated tax **To be calculated**

Order total **\$390.00**

---

Amount due **\$390.00**  
\*Before applicable taxes

### Frequently bought together




Tailored Patchwork Dress Shoes  
**\$214.00**  
 ★★★★★ 49


Suede Dress Shoes  
**\$195.00**  
 ★★★★★ 57

Classic Leather Dress Shoes  
**\$145.00**


Trending < >



Small Belt Leather Satchel  
**\$240.00**




Retro Americana Watch  
**\$495.00**  
 ★★★★★ 1




Diamond Pattereded Crew Neck Sweater  
**\$120.00**

The following image shows an example of a cart page on the Fabrikam site. In this example, there is a handling fee for a line item.

Shopping bag



**Camel Polka-dot Trench**  
 Size: XS  
 Loyalty 5% off over \$100: Savings \$8.50 (5%)  
 Available  
 Handling: \$10.00

 Ship to an address

[Move to wishlist](#) [Remove](#)

Quantity

– 1 +

**\$161.50**  
 Savings \$8.50

**Order summary**

Promo code

Subtotal	<b>\$170.00</b>
Estimated Shipping	<b>To be calculated</b>
Other charges	<b>\$10.00</b>
Estimated Tax	<b>To be calculated</b>
Total discounts	<b>-\$8.50</b>
<b>TOTAL</b>	<b>\$171.50</b>

[Back to shopping](#)

## Cart module properties and slots

PROPERTY	VALUES	DESCRIPTION
Heading	Heading text and a heading tag (H1, H2, H3, H4, H5, or H6)	A heading for the cart, such as "Shopping bag" or "Items in your cart."
Show out of stock errors	<b>True</b> or <b>False</b>	If this property is set to <b>True</b> , the cart page will show stock-related errors. We recommend that you set this property to <b>True</b> if inventory checks are applied on the site.

PROPERTY	VALUES	DESCRIPTION
Show shipping charges for line items	<b>True or False</b>	If this property is set to <b>True</b> , cart line items will show the shipping charges, if this information is available. This feature isn't supported in the Fabrikam theme, because users select shipping only in the checkout flow. However, this feature can be turned on in other workflows if it's applicable.
Show available promotions	<b>True or False</b>	If this property is set to <b>True</b> , the cart shows available promotions, based on items in the cart. This capability is available in the Dynamics 365 Commerce 10.0.16 release.

## Modules that can be used in a cart module

- **Text block** – This module supports custom messaging in the cart module. The messages are driven by the content management system (CMS). Any message can be added, such as "For issues with your order, contact 1-800-Fabrikam."
- **Store selector** – This module shows a list of nearby stores where an item is available for pickup. It lets users enter a location to find stores that are nearby. For more information on this module, see [Store selector module](#).

## Module properties

The following cart module settings can be configured at **Site Settings > Extensions**:

- **Maximum quantity** – This property is used to specify the maximum number of each item that can be added to the cart. For example, a retailer might decide that only 10 of each product can be sold in a single transaction.
- **Inventory** – For information about how to apply inventory settings, see [Apply inventory settings](#).
- **Back to shopping** – This property is used to specify the route for the **Back to shopping** link. The route can be configured at the site level, allowing retailers to take the customer back to the home page or any other page on the site.

### IMPORTANT

In the Dynamics 365 Commerce 10.0.14 release and later, items in the cart are aggregated based on the settings that are defined in the online functionality profile for the online store in Commerce headquarters. For more information about how to create an online functionality profile and set the properties that are required for aggregation, see [Create an online functionality profile](#).

## Commerce Scale Unit interaction

The cart module retrieves product information by using Commerce Scale Unit APIs. The cart ID from the browser cookie is used to retrieve all the product information from Commerce Scale Unit.

## Add a cart module to a page

To add a cart module to a new page and set the required properties, follow these steps.



1. Go to **Fragments**, and select **New** to create a new fragment.
2. In the **New fragment** dialog box, select the **Cart** module.
3. Under **Fragment name**, enter the name **Cart fragment**, and then select **OK**.
4. Select the **Cart** slot.
5. In the properties pane on the right, select the pencil symbol, enter heading text in the field, and then select the check mark symbol.
6. In the **Cart** slot, select the ellipsis (...), and then select **Add Module**.
7. In the **Add Module** dialog box, select the **Store selector** module, and then select **OK**.
8. Select **Save**, select **Finish editing** to check in the fragment, and then select **Publish** to publish it.
9. Go to **Templates**, and select **New** to create a new template.
10. In the **New Template** dialog box, under **Template name**, enter a name for the template.
11. In the outline tree, select the **Body** slot, select the ellipsis (...), and then select **Add fragment**.
12. In the **Select fragment** dialog box, select the **Cart fragment** fragment, and then select **OK**.
13. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
14. Go to **Pages**, and select **New** to create a new page.
15. In the **Choose a template** dialog box, select the template that you created, enter a page name, and then select **OK**.
16. Select **Save**, and then select **Preview** to preview the page.
17. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Cart icon module](#)

[Checkout module](#)

[Payment module](#)

[Shipping address module](#)

[Delivery options module](#)

[Pickup information module](#)

[Order details module](#)

[Gift card module](#)

[Calculate inventory availability for retail channels](#)

[Create an online functionality profile](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cart icon module

2/18/2021 • 2 minutes to read • [Edit Online](#)

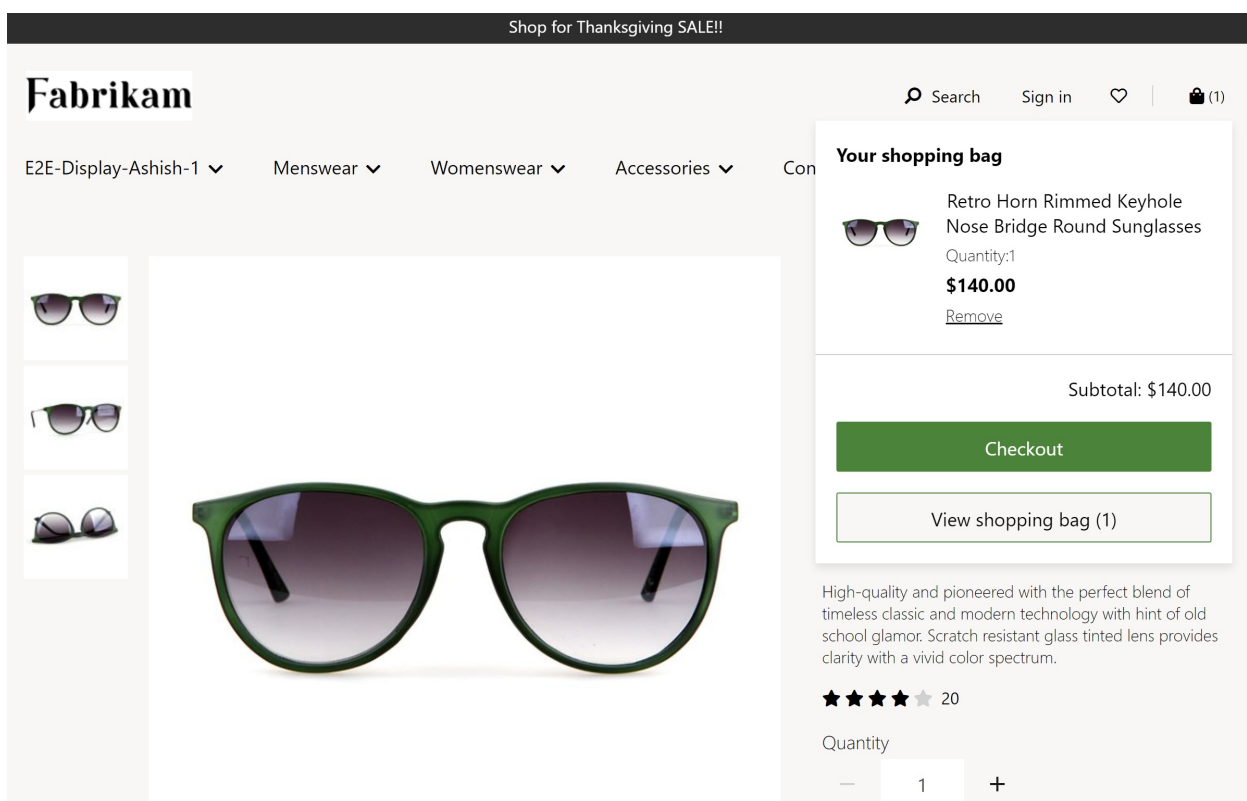
This topic covers the cart icon module and describes how to add it to site pages in Microsoft Dynamics 365 Commerce.

The cart icon module represents the cart in the header module of the page, and shows the number of items in the cart. The cart icon module also displays a cart summary (also known as a mini cart) when the cart icon is hovered over. The mini cart provides the user with a summary of the items in the cart without having to navigate to the cart page. In addition, it also allows the user to directly go to checkout page if they are happy with the summary. This reduces the number of page navigations and makes checkout faster.

## NOTE

Support for the cart icon module is available in the Dynamics 365 Commerce 10.0.11 release.

The following image shows an example of a cart icon module that displays a mini cart in the Fabrikam header.



## Module properties

- **Show mini cart** – When true, this property enables a cart summary (mini cart) to be displayed when the cart icon is hovered over. This functionality is only supported for desktop view ports.

## Add a cart icon module to a page

To add a cart icon module, see [Header module](#).

## Additional resources

Cart module

Checkout module

Payment module

Shipping address module

Delivery options module

Pickup information module

Order details module

Gift card module

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Checkout module

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to add a checkout module to a page and set the required properties.

A checkout module is a special container that hosts all modules that are required to create an order. It presents a step-by-step flow that a customer uses to enter all the relevant information to make a purchase. It captures the shipping address, shipping method, and billing information. It also provides an order summary and other information that is related to a customer order.

A checkout module renders data based on the cart ID. This cart ID is saved as a browser cookie. A cart ID is required to render information in the checkout module, such as the items in the order, the total amount, and discounts.

The following image shows an example of a Fabrikam checkout module on a checkout page.

Summer collection online now!

# Fabrikam

Search Karen (1)

Menswear Womenswear Accessories Contact

## Checkout

### 1. SHIPPING ADDRESS [Change](#)

Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
 4257058000

### 2. DELIVERY OPTION [Change](#)

Standard - Free

### 3. PAYMENT METHOD [Change](#)

*Notice: Payment information is for demonstration purposes only, and no items will be delivered.*

#### Card information

Karen Berg  
Card ending in 1111  
Expires 10/2020

#### Billing address

Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
 4257058000

### 4. CONTACT INFORMATION [Change](#)

Karen.berg@microsoft.com

[Place order](#) [Back to shopping](#)

### Order summary

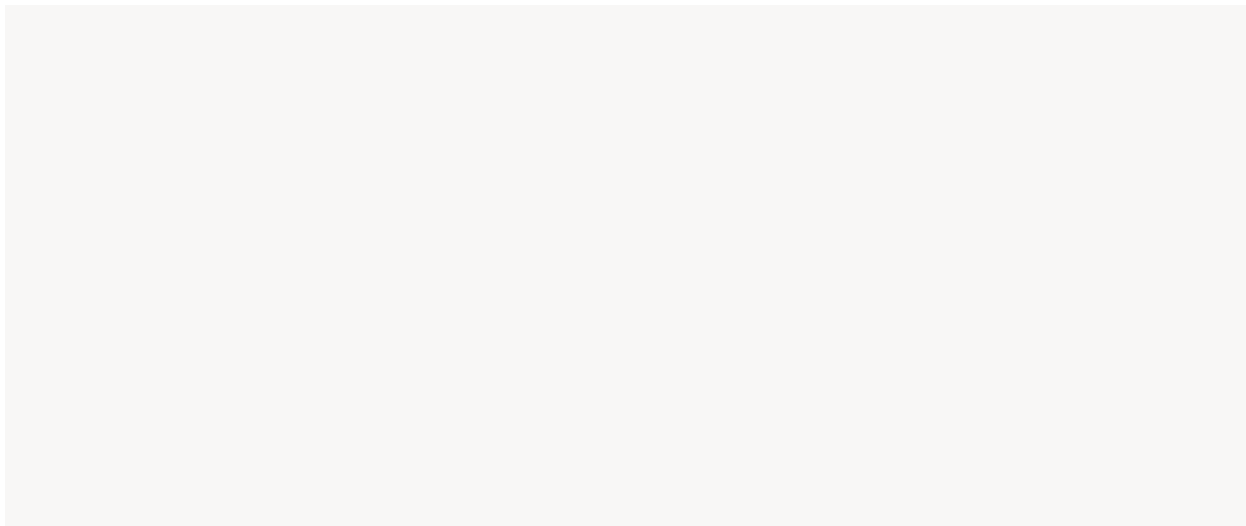
Subtotal	<b>\$390.00</b>
Estimated shipping	<b>Free</b>
Estimated tax	<b>\$25.35</b>
Order total	<b>\$415.35</b>
-----	
Amount due	<b>\$415.35</b>

[Place order](#)

[Back to shopping](#)

### Shipping (1 item) [Edit Cart](#)

**Grand Cap Toe Leather Oxfords**  
Size: 7.5  
Color: Brown  
Style: Regular  
Quantity: 1  
\$390.00



#### Customer Service

FAQ  
Returns & refunds  
Terms & conditions  
Privacy Policy

#### Fabrikam Store

Store locations  
Store hours  
Store events  
Fabrikam store support

#### About us

Our story  
Careers with Fabrikam  
News

#### Follow us



## Checkout module properties

A checkout module shows an order summary and provides the functionality for placing an order. To gather all the customer information that is required before an order can be placed, additional modules must be added to the checkout module. Therefore, retailers have the flexibility to add custom modules to the checkout flow, or to exclude modules, based on their requirements.

PROPERTY NAME	VALUES	DESCRIPTION
Checkout heading	Heading text and a heading tag ( <b>H1</b> , <b>H2</b> , <b>H3</b> , <b>H4</b> , <b>H5</b> , or <b>H6</b> )	A heading for the checkout module.
Order summary heading	Heading text	A heading for the order summary section of the module.
Cart line items heading	Heading text	A heading for cart line items that are shown in the checkout module.
Show shipping charges on line item	<b>True</b> or <b>False</b>	If this property is set to <b>True</b> , the shipping charges that are applicable for line items will be shown on cart lines. If the <b>Header charge with no proration</b> feature is turned on in Commerce headquarters, the shipping charge will be applied at the header level, not the line level. This feature was added in Commerce version 10.0.13.

## Modules that can be used in the checkout module

- **Shipping address** – This module lets a customer add or select the shipping address for an order. For more information about this module, see [Shipping address module](#).

The following image shows an example of a shipping address module on a checkout page.

The screenshot shows a checkout page with a shipping address form on the left and an order summary on the right. The shipping address form includes fields for Name, Street, City, State (a dropdown menu), Zip code, Country (a dropdown menu), and Phone. There are 'Save' and 'Cancel' buttons at the bottom of the form. The order summary on the right shows a subtotal of \$610.00, shipping and tax marked as 'To be calculated', and an amount due of \$610.00. Below the summary are 'Place order' and 'Back to shopping' buttons. A shopping bag section on the right lists two items: 'Belted Dress Boots' (Size: 6, Style: Regular, \$220.00, Quantity: 1) and 'Grand Cap Toe Leather Oxfords' (Size: 7).

- **Delivery options** – This module lets a customer select a mode of delivery for an order. For more information about this module, see [Delivery options module](#).

The following image shows an example of a delivery options module on a checkout page.

The screenshot shows a checkout page with a delivery options module on the left and an order summary on the right. The shipping address is pre-filled with 'Karen Berg, One Microsoft way, Redmond WA 98055, USA, 4257058000'. A 'Change' button is next to the address. The delivery options section has three radio buttons: 'Standard' (selected), 'Standard overnight', and 'Overnight', all with a 'Free' price. There is a 'Save & continue' button. Below the delivery options are sections for '3. PAYMENT METHOD' and '4. CONTACT INFORMATION'. At the bottom are 'Back to shopping' and 'Place order' buttons. The order summary on the right is identical to the previous screenshot, showing a subtotal of \$610.00, shipping and tax marked as 'To be calculated', and an amount due of \$610.00. A shopping bag section on the right lists two items: 'Belted Dress Boots' (Size: 6, Style: Regular, \$220.00, Quantity: 1) and 'Grand Cap Toe Leather Oxfords' (Size: 7).

- **Checkout section container** – This module is a container that you can put multiple modules inside to create a section within the checkout flow. For example, you can put all payment-related modules inside this container to make them appear as one section. This module affects only the layout of the flow.
- **Gift card** – This module lets a customer pay for an order by using a gift card. For more information about this module, see [Gift card module](#).
- **Loyalty points** – This module lets a customer pay for an order by using loyalty points. It provides a summary of available points and expiring points, and lets the customer select the number of points to

redeem. If the customer isn't signed in or isn't a loyalty member, or if the total amount in the cart is 0 (zero), this module is automatically hidden.

- **Payment** – This module lets a customer pay for an order by using a credit or debit card. Customers can also provide a billing address for the payment option that they select. For more information about this module, see [Payment module](#).

The following image shows an example of gift card, loyalty points, and payment modules on a checkout page.

The screenshot displays a checkout page with a '3. PAYMENT METHOD' section on the left and a 'Shopping bag' section on the right. The payment section includes a notice, a loyalty card field (55131), an amount covered by loyalty (\$0.00), and an amount to use field (0). It also features gift card fields for PIN and expiration date, and a credit card section with fields for cardholder name, card number, expiry date, and CVC. A billing address section is also present. The shopping bag section shows a list of items with their prices and discounts.

**3. PAYMENT METHOD**

*Notice: Payment info is for demo purposes only and does not process actual transactions; however, your payment info will be sent to your designated third-party payment provider.*

Loyalty card  
55131

Amount covered by loyalty: **\$0.00** ▼

Amount to use  
0

Gift card  PIN  Expiration date   
PIN  MM/YYYY

Your gift card may or may not include these fields

Credit Card

Cardholder name:

Card Number:

Expiry Date:  CVC / CWV:

**Billing address**

Same as shipping address

Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
4257058000

[Back to shopping](#)

Shopping bag [Edit Cart](#)

Shipping (5 items)

**Belted Dress Boots**  
Size: 6.5  
Style: Regular  
~~\$220.00~~ \$209.00  
Loyalty 5% off over \$100:  
Savings \$11.00 (5%)  
Quantity: 1

**Round Oversized Sunglasses**  
~~\$190.00~~ \$180.50  
Loyalty 5% off over \$100:  
Savings \$9.50 (5%)  
Quantity: 1

**Round Oversized Sunglasses**  
~~\$190.00~~ \$180.50  
Loyalty 5% off over \$100:  
Savings \$9.50 (5%)  
Quantity: 1

**Tailored Patchwork Dress Shoes**  
Size: 7  
Style: Regular  
~~\$214.00~~ \$203.30  
Loyalty 5% off over \$100:  
Savings \$10.70 (5%)  
Quantity: 1

**Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses**  
~~\$140.00~~ \$133.00  
Loyalty 5% off over \$100:  
Savings \$7.00 (5%)  
Quantity: 1

- **Contact information** – This module lets a customer add or change the contact information (email address) for an order.
- **Text block** – This module contains any messaging that is driven by the content management system (CMS). For example, it might contain a message that states, "For issues with your order, contact 1-800-Fabrikam."
- **Checkout terms and conditions** – This module shows rich text that contains the terms and conditions and a check box for the customer input. The check box is optional and configurable. The input is captured by the module and can be used as a check before order placement is triggered, but it isn't included in the order summary information. This module can be added to the checkout container, checkout section container, or terms and conditions slot, according to business needs. If it's added to the checkout container or checkout section container slot, it will appear as a step in the checkout process. If it's added to the terms and conditions slot, it will appear near the order placement button.

The following image shows an example of terms and conditions on a checkout page.

### 3. PAYMENT METHOD Change

**Important Notice.** Payment services are provided by designated third-party payment providers. Payment information is for demonstration purposes only and does not process actual transactions; however, any payment information that you input will be sent to the designated third-party payment provider, and will be handled in accordance with the terms and conditions and privacy statement of the payment provider.

**Card information**  
 Karen Berg Card ending in 1111 Expires 10/2020

**Billing address**  
 Karen Berg  
 One Microsoft way  
 Redmond WA 98055  
 USA  
 4257058000  
 Other

---

### 4. CONTACT INFORMATION Change

karen.berg@outlook.com

---

### 5. Terms and conditions

I have reviewed and accept [Fabrikam Terms and conditions](#)


[Save & continue](#)

[Place order](#)


[Back to shopping](#)

Shopping bag [Edit Cart](#)

**Shipping (3 items)**



**Camel Polka-dot Trench**  
 Size: XXS  
~~\$340.00~~ \$323.00  
 Loyalty 5% off over \$100:  
 Savings \$17.00 (5%)  
 Shipping charges: \$6.46  
 Handling: \$10.00  
 Quantity: 2



**Grand Cap Toe Leather Oxfords**  
 Size: 7.5  
 Color: Brown  
 Style: Regular  
~~\$390.00~~ \$370.50  
 Loyalty 5% off over \$100:  
 Savings \$19.50 (5%)  
 Shipping charges: \$7.41  
 Quantity: 1

[Back to shopping](#)

[Place order](#)

## Commerce Scale Unit interaction

Most of the checkout information, such as the shipping address and shipping method, is stored in the cart and processed as part of the order. The only exception is the credit card information. This information is processed directly by using the Adyen payment connector. The payment is authorized, but it isn't charged until the order is fulfilled.

## Add a checkout module to a page and set the required properties

To add a checkout module to a new page and set the required properties, follow these steps.

1. Go to **Fragments**, and select **New** to create a new fragment.
2. In the **New fragment** dialog box, select the **Checkout** module.
3. Under **Fragment name**, enter the name **Checkout fragment**, and then select **OK**.
4. Select the **Checkout module** slot.
5. In the properties pane on the right, select the pencil symbol, enter heading text in the field, and then select the check mark symbol.
6. In the **Checkout Information** slot, select the ellipsis (...), and then select **Add Module**.
7. In the **Add Module** dialog box, select the **Shipping address**, **Delivery options**, **Checkout section container**, and **Contact information** modules, and then select **OK**.
8. In the **Checkout section container** module, select the ellipsis (...), and then select **Add Module**.
9. In the **Add Module** dialog box, select the **Gift card**, **Loyalty**, and **Payment** modules, and then select **OK**. In this way, you make sure that all the payment methods appear together in a section.
10. In the **Terms and conditions** slot, add a **Checkout terms and conditions** module if it's required. In the module's properties pane, configure the terms and condition text as appropriate.
11. Select **Save**, and then select **Preview** to preview the fragment. Some modules that don't have a cart context might not be rendered in the preview.
12. Select **Finish editing** to check in the fragment, and then select **Publish** to publish it.
13. Create a template that uses the new checkout fragment.
14. Create a checkout page that uses the new template.

## Additional resources



Cart module

Cart icon module

Payment module

Shipping address module

Delivery options module

Pickup information module

Order details module

Gift card module

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Payment module

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic covers the payment module and explains how to configure it in Microsoft Dynamics 365 Commerce.

The payment module lets customers pay for orders by using credit or debit cards. Payment integration for this module is provided by the Dynamics 365 Payment Connector for Adyen. For more information about how to set up and configure the payment connector, see [Dynamics 365 Payment Connector for Adyen](#).

As of Commerce release 10.0.14, the payment module has also been integrated with the Dynamics 365 Payment Connector for PayPal to allow customers to pay for orders using PayPal. For more information on how to set up and configure the Dynamics 365 Payment Connector for PayPal, see [Dynamics 365 Payment Connector for PayPal](#).

## Dynamics 365 Payment Connector for Adyen

The payment module hosts the payment information that is served via Adyen in an HTML inline frame (iframe). The payment module interacts with the Commerce Scale Unit to retrieve the Adyen payment information. As part of the Commerce Scale Unit interaction, the payment module can allow billing address information to be served either in the iframe via Adyen or as a separate module. In the Fabrikam theme, the billing address is enabled as a separate module. This approach allows for more formatting flexibility, because the address lines can be rendered so that they resemble the lines of the shipping address.

The payment module also lets signed-in customers save their payment information. The payment information and billing address are saved and managed via the Adyen payment connector.

The payment module covers any order charges that aren't already covered by loyalty points or a gift card. If the total for an order is fully covered by loyalty points or gift card credits, the payment module will be hidden, and the customer will be able to place the order without it.

The Adyen payment connector also supports strong customer authentication (SCA). Part of the European Union (EU) Revised Payment Services Directive (PSD2) requires that online shoppers be authenticated outside of their online shopping experience when they use an electronic payment method. During the checkout flow, customers are redirected to their banking site, and then after authentication they are redirected back to the Commerce checkout flow. During this redirection, the information that a customer has entered during the checkout flow (for example, shipping address, delivery options, gift card information, and loyalty information) will persist. Before you can turn on the Adyen payment connector feature, the payment connector must be configured for SCA in Commerce headquarters. For more information, see [Strong Customer Authentication using Adyen](#). This feature was enabled in Commerce release 10.0.12.

### NOTE

For the Adyen payment connector, the iframe module in the payment module can be rendered only if you add the Adyen URL to your site's allow list. To complete this step, add `*.adyen.com` to the `child-src`, `connect-src`, `img-src`, `script-src`, and `style-src` directives of your site's content security policy. For more information, see [Manage Content Security Policy](#).

The following illustration shows an example of gift card, loyalty, and Adyen payment modules on a checkout page.

### 3. PAYMENT METHOD

*Notice: Payment info is for demo purposes only and does not process actual transactions; however, your payment info will be sent to your designated third-party payment provider.*

Loyalty card  
55131

Amount covered by loyalty: **\$0.00** ▾

Amount to use


---

Gift card  PIN  Expiration date

Your gift card may or may not include these fields

Credit Card

Cardholder name:

Card Number:

Expiry Date:  CVC / CVV:

**Billing address**

Same as shipping address

Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
📞 4257058000

Place order

[Back to shopping](#)

Shopping bag [Edit Cart](#)

Shipping (5 items)

**Belted Dress Boots**  
Size: 6.5  
Style: Regular  
~~\$229.00~~ \$209.00  
Loyalty 5% off over \$100:  
Savings \$11.00 (5%)  
Quantity: 1

**Round Oversized Sunglasses** ~~\$190.00~~ \$180.50  
Loyalty 5% off over \$100:  
Savings \$9.50 (5%)  
Quantity: 1

**Round Oversized Sunglasses** ~~\$190.00~~ \$180.50  
Loyalty 5% off over \$100:  
Savings \$9.50 (5%)  
Quantity: 1

**Tailored Patchwork Dress Shoes**  
Size: 7  
Style: Regular  
~~\$214.00~~ \$203.30  
Loyalty 5% off over \$100:  
Savings \$10.70 (5%)  
Quantity: 1

**Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses** ~~\$140.00~~ \$133.00  
Loyalty 5% off over \$100:  
Savings \$7.00 (5%)  
Quantity: 1

## Dynamics 365 Payment Connector for PayPal

As of Commerce release 10.0.14, the payment module is also integrated with the Dynamics 365 Payment Connector for PayPal. For more information about how to set up and configure this payment connector, see [Dynamics 365 Payment Connector for PayPal](#).

On the checkout page, you can have both Adyen and the PayPal connectors configured. The payment module has been enhanced with additional properties to help identify which connector it should work with. For details, see the **Supported tender types** and **Is primary payment** module properties in the following table.

When the payment module is configured to use the PayPal payment connector, a PayPal button appears on the checkout page. When invoked by the customer, the payment module renders an iframe containing PayPal information. The customer can sign in and provide their PayPal information within this iframe to complete their transaction. When a customer chooses to pay with PayPal, the remaining balance on the order will be charged via PayPal.

The PayPal payment connector does not require a billing address module because all billing-related information is handled by PayPal within its iframe. However, the shipping address and delivery options modules are required.

The following illustration shows an example of two payment modules on a checkout page, one configured with the Adyen payment connector and the other with the PayPal payment connector.

### 3. PAYMENT METHOD

**Important Notice.** Payment services are provided by designated third-party payment providers. Payment information is for demonstration purposes only and does not process actual transactions; however, any payment information that you input will be sent to the designated third-party payment provider, and will be handled in accordance with the terms and conditions and privacy statement of the payment provider.

Gift card	PIN	Expiration date
<input type="text"/>	<input type="text" value="PIN"/>	<input type="text" value="MM/YYYY"/>

Your gift card may or may not include these fields



Credit Card



Card number

Expiry date	CVC / CVV
<input type="text" value="MM/YY"/>	<input type="text" value="123"/>

Cardholder name

Save for my next payment

#### Billing address

Same as shipping address

Home  
Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
📞 4257058000

[Place order](#)

[Back to shopping](#)

Shopping bag

[Edit Cart](#)

#### In-store pickup (1 item)



#### Distressed 3" Shorts

Size: 4  
~~\$85.00~~ \$76.50  
Student discount: Savings  
\$8.50 (10%)  
Quantity: 1

Pick up at a store  
**San Francisco**

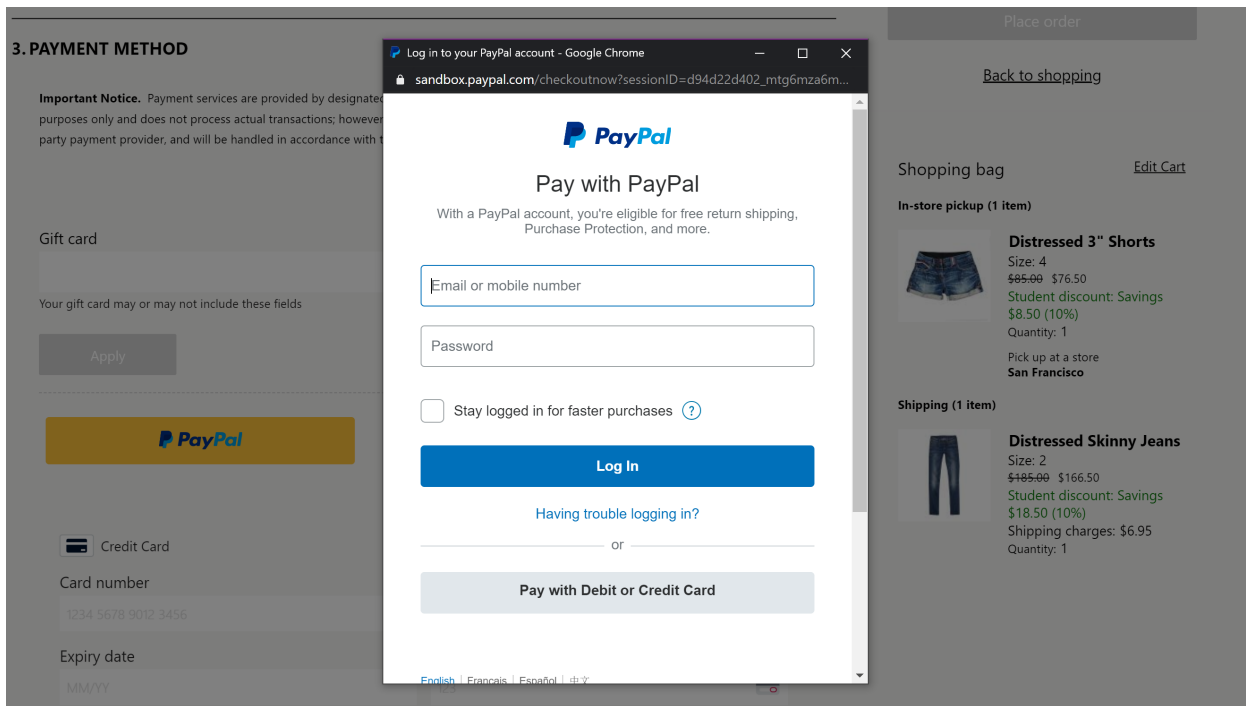
#### Shipping (1 item)



#### Distressed Skinny Jeans

Size: 2  
~~\$185.00~~ \$166.50  
Student discount: Savings  
\$18.50 (10%)  
Shipping charges: \$6.95  
Quantity: 1

The following illustration shows an example of the PayPal iframe invoked using the PayPal button.



## Payment module properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Heading text	An optional heading for the payment module.
Height of the iframe	Pixels	The iframe height, in pixels. The height can be adjusted as required.
Show billing address	True or False	If this property is set to <b>True</b> , the billing address will be served by Adyen inside the payment module iframe. If set to <b>False</b> , the billing address won't be served by Adyen, and a Commerce user will have to configure a module to show the billing address on the checkout page. For the PayPal payment connector, this field has no impact, as the billing address is fully handled within PayPal.
Payment style override	Cascading Style Sheets (CSS) code	Because the payment module is hosted in an iframe, there is limited styling capability. You can achieve some styling by using this property. To override site styles, you must paste the CSS code as the value of this property. Site builder CSS overrides and styles don't apply to this module.

PROPERTY NAME	VALUES	DESCRIPTION
Supported tender types	String	If multiple payment connectors are configured, you should provide the supported tender type string as defined in the Commerce headquarters payment connector configuration (see the following image). If blank, it defaults to the Adyen payment connector. Added in Commerce release 10.0.14.
Is primary payment	True or False	If <b>True</b> , any error messages will be generated from the primary payment connector on the checkout page. If both Adyen and PayPal payment connectors are configured, set Adyen to <b>True</b> , which was added in Commerce release 10.0.14.

The following illustration shows an example of the **Supported Tender Types** value set to "PayPal" in the payment connector configuration in Commerce headquarters.

[Online store](#)

## 128: Fabrikam extended online store

The screenshot shows the configuration page for a payment connector. The 'Supported Tender Types' field is highlighted with a red box and contains the value 'PayPal'. Other fields include 'Connector' (Dynamics 365 Payment Connector), 'Assembly Name' (Microsoft.Dynamics.C...), 'Service Account Id' (ee919080-281c-43a3-...), 'Merchant Client Id' (ARnZk4Lyd1d6OUni8...), 'Merchant API key' (masked with dots), 'Supported Currencies' (USD), and 'Supported Payment Method Variants' (PayPal).

## Billing address

A billing address module can be used on the checkout page if the Adyen payment connector billing address lines do not sufficiently match the appearance of the rest of the site.

To use a billing address module on the checkout page when the payment module is integrated with the Adyen payment connector, set the **Show billing address** property to **False** so that a dedicated billing address module can be used instead of the default Adyen billing address. In this case, the site author should include a billing address module on the checkout page. The Adyen payment connector also allows the ability to use the shipping address as the billing address to minimize the number of steps for the site user.

Similar to payment modules, a **Supported tender types** property has been added to the billing address

module in Commerce release 10.0.14. The value of this property should be identical to the value provided in the payment module to ensure that they work together. For the Adyen payment connector, both the payment module and the billing address module should leave this value blank (the default state). For the PayPal connector, a dedicated billing address module is not required. For other types of payment connectors, the string should be provided as configured in Commerce headquarters.

## Add a payment module to a checkout page and set the required properties

A payment module can be added only to a checkout module. For more information about how to configure a payment module for a checkout page, see [Checkout module](#).

If both Adyen and PayPal payment connectors are needed, add both modules to the payment section. Ensure that the **Supported tender types** property value is configured for PayPal, and leave it blank for Adyen. Also, set the **Is primary payment** property to **True** for Adyen.

## Additional resources

[Cart module](#)

[Cart icon module](#)

[Checkout module](#)

[Shipping address module](#)

[Delivery options module](#)

[Pickup information module](#)

[Order details module](#)

[Gift card module](#)

[Dynamics 365 Payment Connector for Adyen](#)

[Dynamics 365 Payment Connector for PayPal](#)

[Strong Customer Authentication using Adyen](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Shipping address module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes covers the shipping address module and explains how to configure it in Microsoft Dynamics 365 Commerce.

The shipping address module lets customers add or select the shipping address for an order during the checkout flow. If a customer is signed in, any addresses that were previously saved for that customer are shown, and the customer can select among them. The customer can also add a new address. The shipping address module is used for all items on an order that require shipping.

Shipping address formats can be defined in Commerce headquarters for each country or region, and the shipping address module then enforces country/region-specific rules.

When customers enter a shipping address during the checkout flow, they have the option to save the address as a primary address. This option is shown only if a customer is signed in.

Although the shipping address module doesn't provide address validation, this functionality can be implemented through customization.

The following illustration shows an example of a new shipping address module on a checkout page.

The screenshot displays a checkout page with a shipping address form on the left and an order summary on the right. The form is titled "1. SHIPPING ADDRESS" and includes fields for Name, Street, City, State (a dropdown menu), Zip code, Country (a dropdown menu), and Phone. At the bottom of the form are "Save" and "Cancel" buttons. The order summary on the right shows a subtotal of \$610.00, shipping and tax marked as "To be calculated", and a total amount due of \$610.00. Below the summary is a "Place order" button and a "Back to shopping" link. A shopping bag section on the right shows two items: "Belted Dress Boots" (Size: 6, Style: Regular, \$220.00, Quantity: 1) and "Grand Cap Toe Leather Oxfords" (Size: 7).

## Module properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Heading text and a heading tag (H1, H2, H3, H4, H5, or H6)	An optional heading for the shipping address module.



PROPERTY NAME	VALUES	DESCRIPTION
Show address type	True or False	If this optional property is set to <b>True</b> , an address type, such as <b>Home</b> or <b>Business</b> , will be shown. If no address type is specified, the address will automatically be saved as <b>Type=Other</b> .
Enable auto suggestion	True or False	If this optional property is set to <b>True</b> , automatic address suggestions will be provided. These suggestions are powered by Bing Maps. For information about how to set up Bing Maps integration for your site, see <a href="#">Store selector module</a> . This feature is available as of the Commerce version 10.0.15 release.
Auto suggest options	A number	If automatic address suggestions are enabled, you can specify additional options, such as the maximum number of suggestions that should be provided.

## Add a shipping address module to a checkout page and set the required properties

A shipping address module can be added only to a checkout module. For more information about how to configure the shipping address module and add it to a checkout page, see [Checkout module](#).

## Additional resources

[Cart module](#)

[Cart icon module](#)

[Checkout module](#)

[Payment module](#)

[Delivery options module](#)

[Pickup information module](#)

[Order details module](#)

[Gift card module](#)

[Store selector module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Delivery options module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers delivery options modules and explains how to configure them in Microsoft Dynamics 365 Commerce.

Delivery options modules let customers select a mode of delivery such as shipping or pickup for their online order. A shipping address is required to determine the mode of delivery. If the shipping address changes, the delivery options must be retrieved again. If an order includes only items that will be picked up in a store, this module is automatically hidden.

For information about how to configure modes of delivery, see [Online channel setup](#) and [Set up modes of delivery](#).

Each delivery mode can have an associated charge. For more information about how to configure charges for an online store, see [Omni-channel Advanced auto-charges](#).

In Commerce version 10.0.13, the delivery options module has been updated to support the **Header charges without proration** and **Shipping as a line charge** features. If proration is turned off, the expectation is that the e-Commerce workflow won't allow a mixed mode of delivery for the items in the cart (that is, some items are selected for shipment, but others are selected for pickup). The **Header charges without proration** feature requires that the **Enable consistent delivery mode handling in channel** flag be turned on in Commerce headquarters. When that flag is turned on, shipping charges will be applied at either the header level or the line level, depending on the configuration in Commerce headquarters.

The Fabrikam theme supports a mixed mode of delivery, where some items are selected for shipment but others are selected for pickup. In this mode, shipping charges will be prorated for all items that are selected for the shipping mode of delivery. For a mixed mode of delivery to work, you must first configure the **Header charges with proration** feature in Commerce headquarters. For more information about this configuration, see [Prorate header charges to match sales lines](#).

If shipping charges apply to line items, they can be shown on the cart line for each item. This functionality requires that the **Show shipping charges on line item** property be turned on for both the cart module and the checkout module. For more information, see [Cart module](#) and [Checkout module](#).

The following illustration shows an example of a delivery options module on a checkout page.

Checkout

### 1. SHIPPING ADDRESS Change

Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
📞 4257058000

---

### 2. DELIVERY OPTION

Standard Free  
 Standard overnight Free  
 Overnight Free

Save & continue

---

### 3. PAYMENT METHOD

---

### 4. CONTACT INFORMATION

[Back to shopping](#)
Place order

### Order summary

Subtotal	<b>\$610.00</b>
Shipping	<b>To be calculated</b>
Tax	<b>To be calculated</b>
<b>AMOUNT DUE</b>	<b>\$610.00</b>

Place order

[Back to shopping](#)

---

Shopping bag [Edit Cart](#)

**Shipping (2 items)**

**Belted Dress Boots**  
Size: 6  
Style: Regular  
\$220.00  
Quantity: 1

**Grand Cap Toe Leather Oxfords**  
Size: 7

## Delivery options module properties

PROPERTY	VALUES	DESCRIPTION
Heading	Heading text and a heading tag (H1, H2, H3, H4, H5, or H6)	An optional heading for the delivery options module.
Custom CSS class name	Text	A custom Cascading Style Sheets (CSS) class name that will be used to render this module, if applicable.
Filter Delivery Mode Option	<b>Do not filter</b> or <b>Non-shipping modes</b>	A value that specifies whether the delivery options module should filter out all non-shipping delivery modes.

## Add a delivery options module to a checkout page and set the required properties

A delivery options module can be added only to a checkout module. For more information about how to configure the delivery options module and add it to a checkout page, see [Checkout module](#).

## Additional resources

[Cart module](#)

[Checkout module](#)

[Payment module](#)

[Shipping address module](#)

[Pickup information module](#)

[Order details module](#)

Gift card module

Online channel setup

Omni-channel Advanced auto-charges

Prorate header charges to match sales lines

Set up modes of delivery

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Pickup information module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the pickup information module and describes how to add it to checkout pages in Microsoft Dynamics 365 Commerce.

The pickup information module can be used in a checkout module to show order pickup information. Customers can view available pickup dates and time slots, and then select a suitable time to pick up their order. For example, a customer can choose to pick up an order at 3 PM on March 21 from the San Francisco store.

Pickup time slots for the appropriate stores must be configured in Commerce headquarters. For more information, see [Create and update time slots for customer pickup](#).

If a pickup information module is created on a checkout page, but no time slots are defined for the store that is selected for pickup, the module will show information, but the user won't be able to select any time slots. Time slots are optional and aren't required to place an order.

If multiple items are selected for pickup across multiple stores, the pickup information module will let the user select a time slot for each store, provided that time slots are available for it.

## NOTE

Support for time slots and the checkout pickup information module is available in Dynamics 365 Commerce version 10.0.15 and later.

The following illustration shows an example of time slot selection through the pickup information module on a checkout page.

The screenshot displays a checkout page with the following sections:

- Checkout**
- 1. PICKUP INFORMATION**
  - Pick up, San Francisco** (1 items)
  - Pickup date:
  - Pickup time slot:
  -
- 2. PAYMENT METHOD**
- 3. CONTACT INFORMATION**
- 4. TERMS AND CONDITIONS**

At the bottom of the page, there are two buttons:  and .

## Module properties

- **Heading** – Enter a heading for the module.

## Show time slot information after an order is placed

After an order is placed, information about the selected time slot can be viewed in the [order confirmation module](#) and the [order details module](#). Both these modules have a **Show timeslot information** property. Before they can show the selected time slot during the order process, this property must be set to **True**.

## Add a checkout pickup information module to a page

For instructions about how to add a pickup information module to a checkout page and set the required properties, see [Checkout module](#).

The following illustration shows an example of an e-Commerce checkout page that includes time slots for pickup line items.

## Checkout

## 1. PICKUP INFORMATION

[Change](#)**Pick up, San Francisco** (1 items)**Pickup location**

San Francisco  
555 California St.  
San Francisco, CA, SAN FRANCI 94104  
USA

**Pickup date and time slot**

9/24/2020 8:00 AM - 9:00 AM

## 2. PAYMENT METHOD

[Change](#)

**Important Notice.** Payment services are provided by designated third-party payment providers. Payment information is for demonstration purposes only and does not process actual transactions; however, any payment information that you input will be sent to the designated third-party payment provider, and will be handled in accordance with the terms and conditions and privacy statement of the payment provider.

**Payment information**

Karen Berg Visa Card ending in 1111 Expires 10/2020

**Billing address**

Home  
Karen Berg  
One Microsoft way  
Redmond WA 98055  
USA  
📞 4257058000

## 3. CONTACT INFORMATION

[Change](#)

karenb@microsoft.com

## 4. TERMS AND CONDITIONS

[Change](#)

I have reviewed Fabrikam Terms and conditions.

[Back to shopping](#)[Place order](#)

## Order summary

Subtotal	<b>\$158.51</b>
Tax	<b>\$12.74</b>
<b>AMOUNT DUE</b>	<b>\$188.50</b>

[Place order](#)[Back to shopping](#)

## Shopping bag

[Edit Cart](#)

## In-store pickup (1 item)

**Camel Polka-dot Trench**

Size: XS  
\$158.51  
Quantity: 1

Pick up at a store  
**San Francisco**

## Additional resources

[Create and update time slots for customer pickup](#)[Checkout module](#)[Order confirmation module](#)[Order details module](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Order confirmation module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers order confirmation modules and describes how to use them in Microsoft Dynamics 365 Commerce.

The order confirmation module is used to show order confirmation details after an order has been placed. It shows the order confirmation ID, order contact information, and other order details, such as the items that were purchased, payment information, pickup options, and the shipping method.

## Order confirmation module properties

PROPERTY NAME	VALUES	DESCRIPTION
Heading	Heading text and heading tag (H1, H2, H3, H4, H5, or H6)	The order confirmation module can have a heading. By default, the H2 heading tag is used for the heading. However, the tag can be changed to meet accessibility requirements.
Contact number	Text	A contact number can be provided for order-related questions.
Show pickup timeslot information	True or False	This property is available in Dynamics 365 Commerce 10.0.15 and higher. When true, it displays the pickup timeslot information if provided for a pickup item.

## Modules that can be used on an order confirmation page

When you create an order confirmation page, you can add other relevant modules in addition to the order confirmation module. Here are some examples:

- **Recommendations module** – The recommendations module can be added to the order confirmation page to suggest other products to the customer.
- **Marketing modules** – Any marketing module can be added to the order confirmation page to show marketing content.

## Add an order confirmation module to a page

To add an order confirmation module to a new page and set the required properties, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template name**, enter the name **Order confirmation template**, and then select **OK**.
3. In the **Body** slot, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Default page** module, and then select **OK**.
5. In the **Main** slot of the **Default Page** module, select the ellipsis (...), and then select **Add Module**.
6. In the **Add Module** dialog box, select the **Order confirmation** module, and then select **OK**.
7. Select **Save**, and then select **Preview** to preview the template. The order confirmation module won't be

rendered, because it requires the context of the order confirmation number.

8. Select **Finish editing** to check in the template, and then select **Publish** to publish it.
9. Go to **Pages**, and select **New** to create a new page.
10. In the **Choose a template** dialog box, select **Order confirmation template**. Under **Page name**, enter **Order confirmation page**, and then select **OK**.
11. In the **Main** slot of the **Default Page** module, select the ellipsis (...), and then select **Add Module**.
12. In the **Add Module** dialog box, select the **Order confirmation** module, and then select **OK**.
13. In the properties pane for the order confirmation module, select **Heading** next to the pencil symbol.
14. In the **Heading Text** field of the **Heading** dialog box, enter the heading text **Order confirmation**, and then select **OK**.
15. Select **Save**, and then select **Preview** to preview the page.
16. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Cart module](#)

[Cart icon module](#)

[Checkout module](#)

[Payment module](#)

[Shipping address module](#)

[Delivery options module](#)

[Pickup information module](#)

[Gift card module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Gift card module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers gift card modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

Gift card modules can be used in checkout modules to accept gift cards, a common form of payment used for e-Commerce transactions. The gift card module supports Dynamics 365, SVS, and Givex gift cards. SVS and Givex gift cards are redeemed via the Adyen payment provider. For more information about support for external gift cards such as SVS and Givex, see [Support for external gift cards](#).

## NOTE

Support for redeeming SVS and Givex gift cards during checkout flow is available in the Dynamics 365 Commerce 10.0.11 release.

There are two gift card modules available:

- **Gift card** - This module can be used on a checkout page to redeem a gift card as tender.
- **Gift card balance check** - This module can be used on any page to check the balance on a gift card. This module is available in Commerce release 10.0.14 and later.

## NOTE

Support for the gift card balance check module is available in the Dynamics 365 Commerce 10.0.14 release.

The following image shows an example of a gift card module on a checkout page.

Gift card	PIN	Expiration date
<input type="text"/>	<input type="text" value="PIN"/>	<input type="text" value="MM/YYYY"/>
Your gift card may or may not include these fields		
<input type="button" value="Apply"/>		
Gift card ending in 5000 (Available balance \$3,415.13) <a href="#">Remove</a>		

## Module properties

- **Show additional fields** - This property defines what fields should be displayed for gift cards in addition to the gift card number, which is always displayed by default. For example, some gift cards support displaying a personal identification number (PIN), and others support displaying a PIN and expiration date. Alternatively, this property could be set to "None", which would only display the gift card number and no additional fields.

Supported values:

- PIN
- Expiration date
- PIN and expiration date

- None

## Site settings for gift card modules

In Commerce site builder under **Site Settings > Extensions**, there is a gift card module setting called **Supported gift card type**. This setting supports three values:

- **Dynamics 365 gift card** - When this setting is applied, the gift card module only allows the redemption of Dynamics 365 gift cards. This setting is only supported for signed-in users on the e-Commerce site.
- **SVS and Givex gift cards** - When this setting is applied, the gift card module only allows the redemption of Givex and SVS gift cards. This setting is supported for signed-in and anonymous users on the e-Commerce site.
- **Dynamics 365, SVS, and Givex gift cards** - When this setting is applied, the gift card module allows the redemption of Dynamics 365, Givex, and SVS gift cards. This setting is only supported for signed-in users on the e-Commerce site.

### IMPORTANT

These settings are available in the Dynamics 365 Commerce 10.0.11 release and are required only if you need support for SVS or Givex gift cards. If you are updating from an older version of Dynamics 365 Commerce, you must manually update the appsettings.json file. For instructions on updating the appsettings.json file, see [SDK and module library updates](#).

## Add a gift card module to a page

For instructions on how to add a gift card module to a checkout page and set the required properties, see [Checkout module](#).

## Additional resources

[Cart module](#)

[Cart icon module](#)

[Checkout module](#)

[Payment module](#)

[Shipping address module](#)

[Delivery options module](#)

[Pickup information module](#)

[Order details module](#)

[Support for external gift cards](#)

[SDK and module library updates](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Map module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers map modules and describes how to configure them in Microsoft Dynamics 365 Commerce.

A map module shows the locations of stores on an interactive map that is rendered by using the [Bing Maps V8 Web Control](#). A Bing Maps API key is required and must be added to the shared parameters page in Commerce headquarters. Map modules provide different views, such as Road, Aerial, and Streetside, that users can select to view map locations. They also allow for interactions such as zooming and using the user's location.

A map module works in conjunction with the store selector module to determine the geographic locations of stores that must be rendered on a map. Store selector and map modules interact when a user selects a store in one of those modules on a site page. Map modules can be extended for other scenarios, beyond interaction with store selector modules. However, module customization is required.

## NOTE

The map module is available in the Dynamics 365 Commerce 10.0.13 release.

The following image shows an example of a map module that is used on a store locations page.

**Fabrikam store locator and directory**  
We found 2 stores within 100 miles of you.

Enter ZIP code or city, state

See all stores

**1. Bellevue** (8 mi)  
**Contact**  
Bellevue Square  
Bellevue, WA 98004  
USA  
**Store hours**  
9:00 am-9:00 pm  
Set as preferred store

**2. Seattle** (9 mi)  
**Contact**  
2624 University Village  
Plaza NE  
Seattle, WA 98105  
USA  
**Store hours**  
9:00 am-9:00 pm  
 **Preferred store**

Microsoft Bing Maps Terms

## Module properties

PROPERTY NAME	VALUE	DESCRIPTION
Heading	Text	The heading for the module.
Pushpin options: Default icon	Image	The pushpin symbol image to use for stores that are shown on a map.
Pushpin options: Active icon	Image	The pushpin symbol image to use for a store that is selected on a map.

PROPERTY NAME	VALUE	DESCRIPTION
Pushpin options: Default icon color	Character string	The text or hexadecimal value for the color of pushpin symbols on a map.
Pushpin options: Active icon color	Character string	The text or hexadecimal value for the color of selected pushpin symbols on a map.
Show index	<b>True or False</b>	If this property is set to <b>True</b> , every pushpin symbol that indicates a store will show an index. This index will match the index in the list view that the store selector module shows.

## Add allowed mapping URLs to a site's content security policy directives

For the maps module to interact with Bing Maps, you must ensure that the following mapping URLs are allowed per your site's content security policy (CSP). This setup is done in Commerce site builder, by adding allowed URLs to various site CSP directives (for example, `img-src`). For more information, see [Content security policy](#).

- To the `connect-src` directive, add `*.bing.com`.
- To the `img-src` directive, add `*.virtualearth.net`.
- To the `script-src` directive, add `*.bing.com`, `*.virtualearth.net`.
- To the `script style-src` directive, add `*.bing.com`.

## Add a map module to a page

For detailed information about how to configure a map module on a page, see [Store selector module](#).

## Additional resources

[Module library overview](#)

[Buy box module](#)

[Cart module](#)

[Store selector module](#)

[Manage Bing Maps for your organization](#)

[Bing Maps V8 Web Control](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Store selector module

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic covers the store selector module and describes how to add it to site pages in Microsoft Dynamics 365 Commerce.

Customers can use the store selector module to pick up a product in a selected store after an online purchase. In Commerce version 10.0.13, the store selector module also includes additional capabilities that can showcase a **Find a Store** page that shows nearby stores.

The store selector module lets users enter a location (city, state, address, and so on) to search for stores within a search radius. When the module is first opened, it uses the customer's browser location to find stores (if consent is provided).

## Store selector module usage in e-Commerce

- A store selector module can be used on a product details page (PDP) to select a store for pickup.
- A store selector module can be used on a cart page to select a store for pickup.
- A store selector module can be used on a standalone page that shows all available stores.

## Bing Maps integration

The store selector module is integrated with the [Bing Maps REST application programming interfaces \(APIs\)](#) to use Bing's Geocoding and Autosuggest features. A Bing Maps API key is required and must be added to the shared parameters page in Commerce headquarters. The Geocoding API is used to convert a location to latitude and longitude values. The integration with the Autosuggest API is used to show search suggestions when users enter locations in the search field.

For the Autosuggest REST API, you must ensure that the following URLs are allowed per your site's content security policy (CSP). This setup is done in Commerce site builder, by adding allowed URLs to various CSP directives for the site (for example, `img-src`). For more information, see [Content security policy](#).

- To the `connect-src` directive, add `*.bing.com`.
- To the `img-src` directive, add `*.virtualearth.net`.
- To the `script-src` directive, add `*.bing.com`, `*.virtualearth.net`.
- To the `script style-src` directive, add `*.bing.com`.

## Pickup in store mode

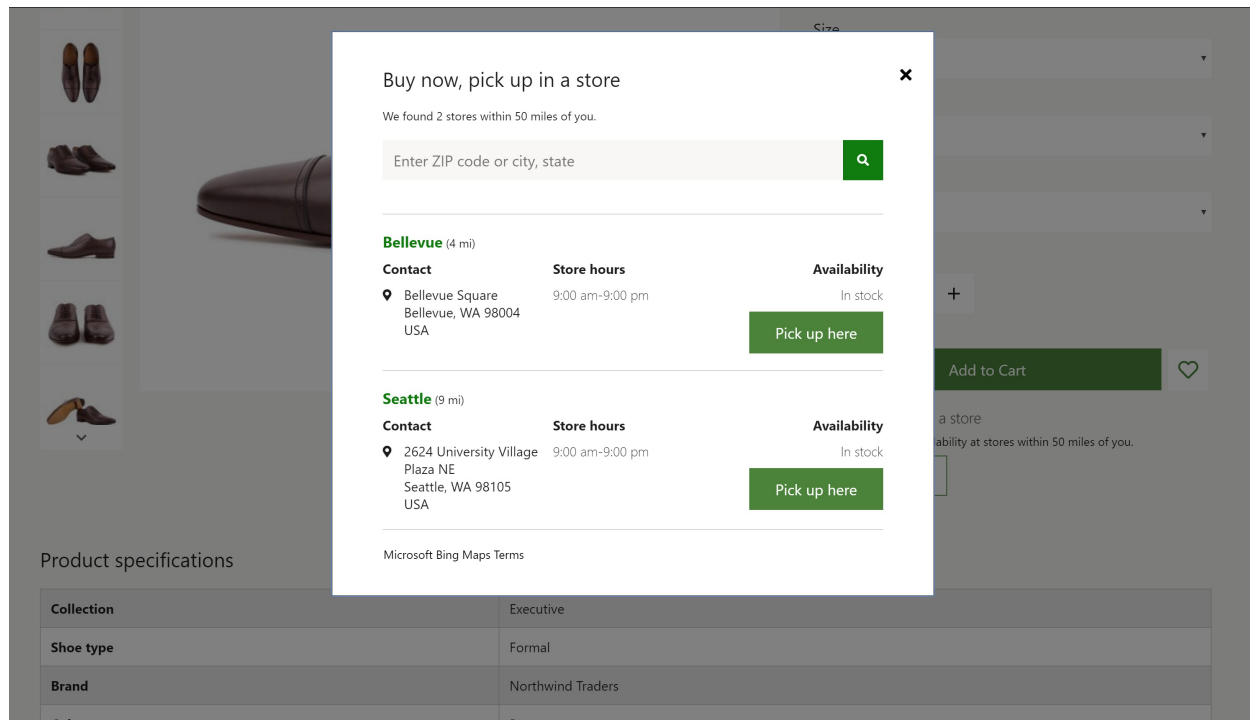
The store selector module supports a **Pick up in store** mode that shows a list of stores where a product is available for pickup. It also shows store hours and product inventory for each store in the list. The store selector module requires the context of a product to render product availability and to let the user add the product to the cart, if the product's delivery mode is set to **pickup** at the selected store. For more information, see [Inventory settings](#).

The store selector module can be added to a buy box module on a PDP to show stores where a product is available for pickup. It can also be added to a cart module. In this case, the store selector module shows pickup options for each line item in the cart. The store selector module can also be added to other pages or modules via extensions and customizations.

For this scenario to work, products should be configured so that the **pickup** delivery mode is used. Otherwise,

the module won't be shown on the product pages. For more information about how to configure the delivery mode, see [Set up modes of delivery](#).

The following image shows an example of a store selector module used on a PDP.



#### NOTE

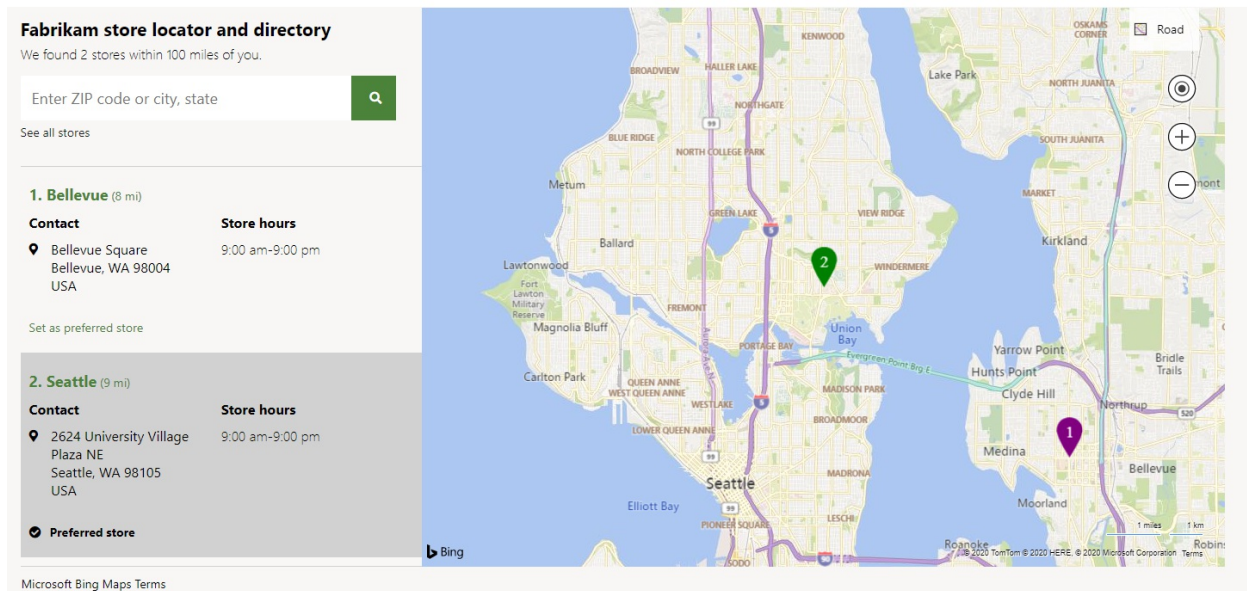
In version 10.0.16 and later, a new feature can be enabled which allows an organization to define multiple pick up modes of delivery options for customers. If this feature is enabled, the store selector and other modules of e-Commerce will be enhanced to allow the shopper to choose from potentially multiple pick up delivery options if configured. To learn more about this feature, refer to [this documentation](#).

## Find stores mode

The store selector module also supports a **Find stores** mode. This mode can be used to create a store locations page that shows available stores and their information. In this mode, the store selector module works without product context and can be used as a standalone module on any site page. In addition, if the relevant settings are turned on for the module, users can select a store as their preferred store. When a store is selected as a user's preferred store, the store ID is maintained in the browser cookie. Therefore, the user must accept a cookie consent message.

The following illustration shows an example of a store selector module that is used together with a map module on a store locations page.





## Render a map

The store selector module can be used together with the map module to show the store locations on a map. For more information about the map module, see [Map module](#)

## Store selector module properties

PROPERTY NAME	VALUE	DESCRIPTION
Heading	Text	The heading for the module.
Mode	<b>Find stores</b> or <b>Pick up in store</b>	<b>Find stores</b> mode shows available stores. <b>Pick up in store</b> mode lets users select a store for pickup.
Style	<b>Dialog</b> or <b>Inline</b>	The module can be rendered either inline or in a dialog box.
Set as preferred store	<b>True</b> or <b>False</b>	When this property is set to <b>True</b> , users can set a preferred store. This feature requires that users accept a cookie consent message.
Show all stores	<b>True</b> or <b>False</b>	When this property is set to <b>True</b> , users can bypass the <b>Search radius</b> property and view all stores.
Autosuggest options: Max results	Number	This property defines the maximum number of autosuggest results that can be shown via the Bing Autosuggest API.
Search radius	Number	This property defines the search radius for stores, in miles. If no value is specified, the default search radius of 50 miles is used.

PROPERTY NAME	VALUE	DESCRIPTION
Terms of service	URL	This property specifies the terms of service URL that is required to use the Bing Maps service.

## Add a store selector module to a page

For **Pickup in store** mode, the module can be used only on PDPs and cart pages. You must set the mode to **Pickup in store** in the module's property pane.

- For information on how to add a store selector module to a buy box module, see [Buy box module](#).
- For information on how to add a store selector module to a cart module, see [Cart module](#)

To configure the store selector module to show available stores for a store locations page, as in the illustration that appears earlier in this topic, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New Template** dialog box, under **Template name**, enter **Marketing template**, and then select **OK**.
3. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
4. Go to **Pages**, and select **New** to create a new page.
5. In the **Choose a template** dialog box, select the **Marketing template** template. Under **Page name**, enter **Store locations**, and then select **OK**.
6. In the **Main** slot of the new page, select the ellipsis (...), and then select **Add Module**.
7. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
8. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
9. In the **Add Module** dialog box, select the **Container with 2 columns** module, and then select **OK**.
10. In the module's properties pane, set the **Width** value to **Fill Container**.
11. Set the **X-Small view port column configuration** value to **100%**.
12. Set the **Small view port column configuration** value to **100%**.
13. Set the **Medium view port column configuration** value to **33% 67%**.
14. Set the **Large view port column configuration** value to **33% 67%**.
15. In the **Container with 2 columns** slot, select the ellipsis (...), and then select **Add Module**.
16. In the **Add Module** dialog box, select the **Store selector** module, and then select **OK**.
17. In the module's properties pane, set the **Mode** value to **Find stores**.
18. Set the **Search radius** value in miles.
19. Set other properties, such as **Set as preferred store**, **Show all stores**, and **Enable auto suggestion**, as you require.
20. In the **Container with 2 columns** slot, select the ellipsis (...), and then select **Add Module**.
21. In the **Add Module** dialog box, select the **Map** module, and then select **OK**.
22. In the module's properties pane, set any additional properties as you require.
23. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Buy box module](#)

[Cart module](#)

[Quick tour of PDP](#)

[Quick tour of cart and checkout](#)

[Set up modes of delivery](#)

[Manage Bing Maps for your organization](#)

[Bing Maps REST APIs](#)

[Maps module](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Account management pages and modules

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic covers account management pages and modules in Microsoft Dynamics 365 Commerce.

Account management refers to a group of pages that is used to manage user account-related information in Dynamics 365 Commerce. Account management pages include the account management landing page, user profile page, user address page, order history page, order details page, loyalty page, and wish list page.

## Account management landing page

The account management landing page uses the following modules:

- **Container** - All account management landing page modules should be placed within a container.
- **Account welcome tile** – This module is used to provide a welcome message on the account management page. It includes properties for the heading.
- **Account generic tile** - This module can be used to provide headings and links to account management pages, such as the "Order history" or "My profile" pages. The generic tile module can be used to configure a tile for any page. In Fabrikam, this module is used for "Order history" and "My profile" page links on the account management landing page.
- **Account wishlist tile** – This module is used to provide a summary of the items on the customer's wish list. For example, it might state, "You have 10 items in your wish list." It includes properties for the heading and the "View details" link. The "View details" link should be configured to redirect to the wish list page.
- **Account address tile** – This module is used to provide a summary of the user's addresses. For example, it might state, "You have 2 addresses added to your account." It includes properties for the heading and the "View details" link. The "View details" link should be configured to redirect to the user address page.
- **Account loyalty tile** – This module is used to display and link to loyalty program information. This tile has two states: one state shows links to join a loyalty program if the user is not a member already. The other state shows links to view the loyalty details page when the user is already a member. Properties include the heading, the "Sign-up" link, and the "View loyalty" link. The "View loyalty" link should be configured to redirect to the loyalty page. The "Sign-up" link should be configured to redirect to a page where users can join the loyalty program.

## Order history page

The order history page uses the order history module to show all the recent orders that the user has placed.

## Order details page

The order details page provides detailed information for each order and is accessed from the order history page. It uses the order details module, which requires the sales ID or transaction ID to retrieve the order details.

## User profile page

The user profile page shows user account details, such as a user's name and email address. It uses the user profile details and user profile edit modules. Although the email address can't be removed, it can be edited. The user profile page also shows user preferences that enable a user to opt in or opt out from certain features, such as personalization of recommendation lists.

## User address page

The user address page shows the list of addresses that are associated with the user account. The user either provided these addresses during checkout or added them directly on this page. The user address module is used to add and edit addresses, set the primary address, and render existing addresses on the page.

### **Wish list page**

The wish list page shows the items that have been added to the customer's wish list. It uses the wish list module to render wish list items.

### **Loyalty page**

The loyalty page lets customers view their loyalty details if they are already loyalty program members. They can also view the points that they have earned and redeemed in recent transactions. The page uses the loyalty details module to showcase the loyalty details.

To join loyalty program, a marketing page can be created with loyalty sign up and loyalty terms modules. If the user is not a member of a loyalty program, these modules will enable the user to sign up.

## **Additional resources**

[Module library overview](#)

[Container module](#)

[Buy box module](#)

[Cart module](#)

[Checkout module](#)

[Order confirmation module](#)

[Header module](#)

[Footer module](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Product collection modules

2/18/2021 • 6 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic provides an overview of product collection modules in Microsoft Dynamics 365 Commerce.

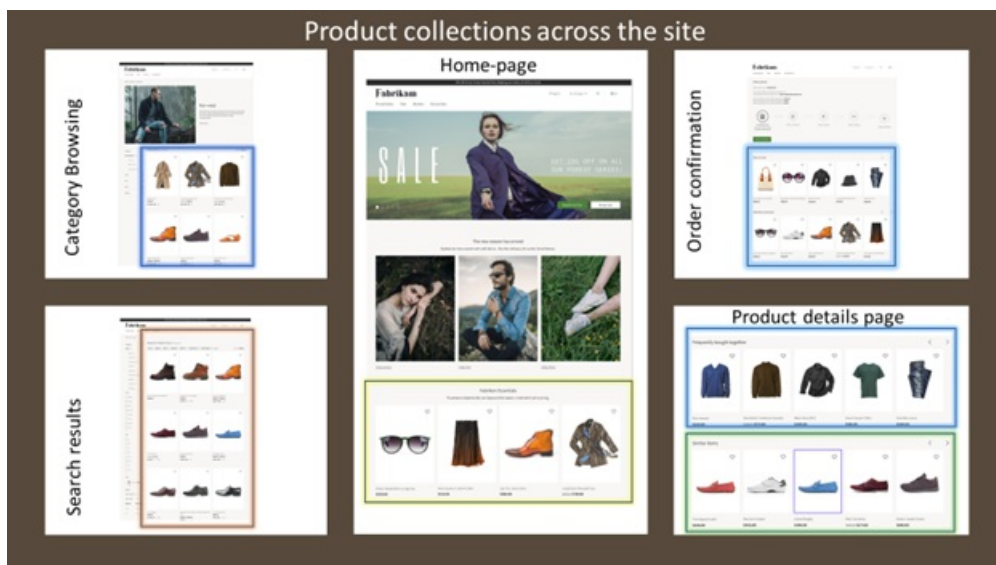
Product discovery is a primary tool that retailers use to engage with their customers on an e-Commerce website. Product collection modules help retailers build compelling shopping experiences by providing an intuitive visual interface that can be used to quickly author product collections.

Product collection modules represent physical products and services on the website. A product collection module is typically linked to a details page where customers can purchase a product or service, or learn more about it.

The sources for product collections can be lists of the following four types:

- Editorial lists of products that are manually defined in Dynamics 365 Commerce as related products for a product, or product lists
- Algorithmic lists, such as lists of new, best-selling, or trending products
- Recommendation lists that are based on machine learning
- Personalization lists that support personalized results for a customer. Customers must be signed in to the e-Commerce site to see personalized results. Guest users don't see personalized results. Customers can opt out of personalization from the [account management page](#).

The following illustration shows the different types of product collections being used on an e-Commerce site.



## NOTE

Always use product collection modules to show a group of products of a similar type.

# Product collection modules and types

The following table describes various types of product collection modules in Dynamics 365 Commerce.

PRODUCT COLLECTION MODULE	TYPE	DESCRIPTION
Category	Category	This module shows a list of products in a category, as defined by the navigation category hierarchy that the retailer created for a channel.
Related products	Editorial	This module shows a list of products that a merchandising manager has configured as related products in Commerce, for the relation type that the author has selected.
Search results	Search query	This type of product collection module shows a list of products that best match the search query that the customer entered.
Curated product lists	Editorial	This module shows custom lists that merchandisers and editors have created in Commerce.
New	Algorithmic	This module shows a list of the newest products that have been assorted to channels and catalogs. This list can show personalized results for a signed-in user if the site author chooses that option.
Best selling	Algorithmic	This module shows a list of products that are ranked by the highest number of sales. This list can show personalized results for a signed-in user if the site author chooses that option.
Trending	Algorithmic	This module shows a list of the highest-performing products for a given period. This list can show personalized results for a signed-in user if the site author chooses that option.
Frequently bought together	Artificial intelligence/Machine learning	This module uses machine learning to analyze consumer purchase patterns and recommend related items that are frequently bought together with a given product. This list can show personalized results for a signed-in user if the site author chooses that option.

PRODUCT COLLECTION MODULE	TYPE	DESCRIPTION
People also like	Artificial intelligence/Machine learning	This module uses machine learning to analyze consumer purchase patterns and recommend items that are related to a given product. This list can show personalized results for a signed-in user if the site author chooses that option.
Picks for you	Artificial intelligence/Machine learning	This module uses machine learning to analyze the purchase patterns of the signed-in user and provide personalized recommendations that are based on those purchase patterns. For a guest user, this list will be collapsed.

## Supported modules

The product collection module supports the [quick view module](#), which lets users view product information and add items to the cart from a product collection page.

## Add a product collection module to a category page

To add a product collection module to a category page, follow these steps.

1. Go to **Pages**, and select **New** to create a new page.
2. In the **Choose a template** dialog box, select the same template as that used by your default category page. Under **Page name**, enter an appropriate name, and then select **OK**.
3. In the **Sub footer** slot, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
5. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
6. In the **Add Module** dialog box, select the **Product collection** module, and then select **OK**.
7. In the properties pane for the product collection module, select **Add a product list**.
8. In the **Select product list configuration** dialog box, select the type of list, the list source, and enter the number of items. Configure any other options that are available for the list type. For more information about list types, see the table that follows.
9. Select **OK**.
10. Select **Save**, and then select **Preview** to preview the page.
11. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

The following table shows the list types that are available for selection in the **Select product list configuration** dialog box.

TYPE	DESCRIPTION	USAGE	PAGE CONTEXT	SPECIFIC CONTEXT	PERSONALIZATION
------	-------------	-------	--------------	------------------	-----------------



TYPE	DESCRIPTION	USAGE	PAGE CONTEXT	SPECIFIC CONTEXT	PERSONALIZATION
Products by category	A list of products that belong to a given category. This category is determined from either the page context or the context that the author provides.	This type of list can be used on any page (for example, a home page, category page, marketing page, or product details page [PDP]) to promote a specific category of products.	Category from the page context, where available (for example, a category page)	The author can provide a specific category as context for the list.	Not applicable
Related products	A list of products that a merchandising manager has configured as related products for the relation type in Commerce.	This type of list is used primarily on PDPs, but it can be used on any page if a parent product is provided.	Product from the page, relation type (mandatory)	The product can be selected in the picker, and the relation type is used.	Not applicable
Curated	A custom list that merchandisers and editors have created in Commerce.	Enrich category page, home page, checkout and cart pages, and product pages	Not applicable	Not applicable	Not applicable

TYPE	DESCRIPTION	USAGE	PAGE CONTEXT	SPECIFIC CONTEXT	PERSONALIZATION
Algorithmic	<ul style="list-style-type: none"> <li>• <b>New</b> – A list of the newest products that have been assorted to channels and catalogs.</li> <li>• <b>Best-selling</b> – A list of products that are ranked by the highest number of sales.</li> <li>• <b>Trending</b> – A list of the highest-performing products for a given period.</li> </ul>	Home page, enrich category page, and checkout and cart pages	Category from the page context (for example, a category page)	The category that is determined by the site author	Supported
Frequently bought together	A list that uses machine learning to analyze consumer purchase patterns and recommend related items that are frequently bought together with a given product.	This type of list is applicable only to the cart page.	Cart	Not applicable	Supported
People also like	A list that uses machine learning to analyze consumer purchase patterns and recommend items that are related to a given product.	This type of list is used on PDPs to show products that other customers have bought.	Product context from the page	The product that is provided by the site author	Supported

TYPE	DESCRIPTION	USAGE	PAGE CONTEXT	SPECIFIC CONTEXT	PERSONALIZATION
Picks for you	A list that uses machine learning to determine customer preferences.	This type of list can be used on any page.	Not applicable	Not applicable	Supported

## Additional resources

[Module library overview](#)

[Carousel module](#)

[Content rich block module](#)

[Container module](#)

[Buy box module](#)

[Product recommendations overview](#)

[Quick view module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Quick view module

2/18/2021 • 2 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic covers quick view modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

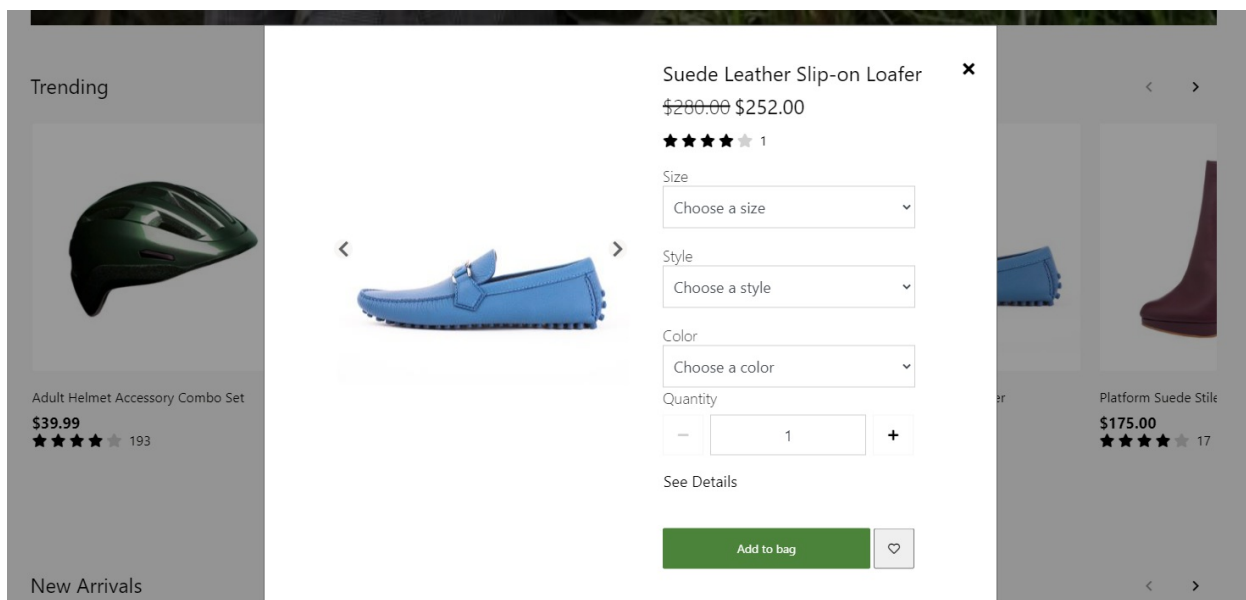
The quick view module lets users quickly view product information when they browse products on a list page, and add one or more products to the cart from the list page, without having to go to the product details page (PDP). The quick view module provides an overview of the product information that users require to make an "add to cart" decision. It also provides a link to the PDP, so that users can view additional product details and purchase options.

The quick view module is supported by the [product collection](#) and [search results](#) modules.

## IMPORTANT

The quick view module is available as of the Commerce version 10.0.17 release.

The following illustration shows an example of a quick view module on a product list page.



## Module properties

The quick view module supports some of the same functions as the buy box module. Therefore, the properties of a quick view module resemble the properties of a buy box module.

PROPERTY	VALUES	DESCRIPTION
----------	--------	-------------

PROPERTY	VALUES	DESCRIPTION
Heading tag	H1, H2, H3, H4, H5, or H6	This property defines the heading tag for the product title. If the quick view module is at the top of the page, this property should be set to H1 to meet accessibility standards.
Allow custom price	True or False	If this property is set to True, the user can enter a custom price.
Minimum price	Integer	This property is applicable only if the <b>Allow custom price</b> property is set to True. It defines the minimum price that the user can enter (for example, \$1).
Maximum price	Integer	This property is applicable only if the <b>Allow custom price</b> property is set to True. It defines the maximum price that the user can enter (for example, \$1,000).

## Commerce site builder settings

Like the buy box module, the quick view module respects the settings at **Site Settings > Extensions > Add product to cart**. However, the **Navigate to cart page** setting is ignored, because it's inconsistent with the purpose of the quick view module, which is to enable users to browse multiple products on a list page and add them to the cart without moving away from the list page.

## Add a quick view module to a product collection module

A quick view module can be added to the product collection and search results modules.

To add a quick view module to a product collection module in Commerce site builder, follow these steps.

1. Go to **Pages**, and then select the home page for the Fabrikam site.
2. Go to any **Product Collection** module on the homepage, select the ellipsis (...), and then select **Add Module**.
3. In the **Add Module** dialog box, select the **Quick View** module, and then select **OK**.
4. In the properties pane of the **Quick View** module, select **Heading**. In the **Heading** dialog box, set the **Heading Level** field to H2, and then select **OK**.
5. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Buy box module](#)

[Product collection module](#)

[Search results module](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Breadcrumb module

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic covers breadcrumb modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

Breadcrumb modules are used to provide secondary navigation on site pages. They are typically shown at the top of a page, below the header. Although breadcrumb modules can be added to any page, they are most often used on product details pages (PDPs), to show the product category hierarchy and provide a quick way to move around a site. A breadcrumb module can also be used to show a "Back to results" link when users open a PDP from a search or list page. In this way, users can quickly return to their filtered list page to continue shopping.

On pages that have product category context, such as PDPs and category pages, breadcrumb modules show the category hierarchy. On pages that don't have category context, breadcrumb modules show **<Site root> / <Current page>** by default. Breadcrumb modules can also be manually configured on other types of site pages to show links to specific pages on the site.

## NOTE

The breadcrumb module is available in the Dynamics 365 Commerce 10.0.12 release.

The following image shows an example of a breadcrumb module that shows the category hierarchy on a PDP.

The screenshot shows a product page for a brown leather pull-on workboot. At the top left is the Fabrikam logo with a hamburger menu icon. At the top right is a shopping bag icon with a '(2)' next to it. Below the logo is a breadcrumb trail: [Fabrikam Fashion](#) / [Menswear](#) / [Shoes](#) / [Leather Pull-on Workboot](#). The main image shows a side view of the brown leather boot with a pull-on tab at the back. To the right of the image, the product title 'Leather Pull-on Workboot' is displayed in a large font, followed by the price '\$170.00'. Below the price is a short description: 'Functional pull-on workboot made of authentic leather.' There is a 'Size' dropdown menu with 'Choose a size' selected. Below that is a 'Quantity' section with minus, plus, and '1' buttons. At the bottom of the product information is a green 'Add to bag' button and a heart icon for wishlists. At the very bottom of the page, there is a navigation bar with a back arrow, four circular indicators (the first is filled), and a forward arrow. Below the navigation bar, there is a promotional message: 'Buy now, pick up in a store Search for product availability at stores within 100 miles of you.'

# Breadcrumb module settings

The breadcrumb module relies on the **Breadcrumb display type on PDP** setting, which is defined at **Site Settings > Extensions** in site builder. This setting has three possible values:

- **Show category hierarchy** – When this value is selected, the breadcrumb module will show the full category hierarchy of the product that is viewed on the PDP.
- **Show back to results** – When this value is selected, the breadcrumb module will show a "Back to results" link on a PDP if the user opened the PDP from a module that allows for a "Back to results" link. This functionality is available when users navigate from category, search, list, and recommendation lists pages. To support this functionality, product collection and search results modules have a property that is named **Allow back to results on PDP**. This property gives you the flexibility to define which modules should support the "Back to results" link functionality on the PDP. For example, when **Show back to results** is selected for the **Breadcrumb display type on PDP** setting of the breadcrumb module, and **Allow back to results on PDP** is selected for the search page search results module, a "Back to results" link will be shown when users navigate from the search page to a PDP.
- **Show category hierarchy and back to results** – This value is a combination of the previous two. When this value is selected, the breadcrumb module will show both the full category hierarchy and a "Back to results" link (if it's configured) on a PDP.

## IMPORTANT

These settings are available in the Dynamics 365 Commerce 10.0.12 release. If you are updating from an older version of Dynamics 365 Commerce, you must manually update the appsettings.json file. For instructions on updating the appsettings.json file, see [SDK and module library updates](#).

# Breadcrumb module properties

PROPERTY NAME	VALUES	DESCRIPTION
Root	Text or link	This optional property specifies link text and a link target for the breadcrumb site root. If this property isn't configured, no root will be defined.
Breadcrumb link	Link	This optional property specifies links for a manually configured breadcrumb, if these links are required. Links appear in the order that they are listed in.

# Add a breadcrumb module to a new page

To add a breadcrumb module to a PDP and set the required properties, follow these steps.

1. Go to **Site Settings > Extensions**, and then, for the **Breadcrumb display type on PDP** setting, select **Show category hierarchy**.
2. Go to **Templates**, and select the PDP template.
3. In the **Container** slot that contains the buy box module, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Breadcrumb** module, and then select **OK**.
5. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.
6. Go to **Pages**, and open a PDP that uses the PDP template. If a PDP doesn't yet exist, create one.
7. In the **Container** slot that contains the buy box module, select the ellipsis (...), and then select **Add Module**.



8. In the **Add Module** dialog box, select the **Breadcrumb** module, and then select **OK**.
9. In the properties pane of the **Breadcrumb** slot, under **Root**, select **Link text**.
10. In the **Link text** dialog box, enter **Home**, and then, under **Link target**, select **Add a link**.
11. In the **Add a link** dialog box, select a link for the breadcrumb root, and then select **OK**.
12. Select **Save**, and then select **Preview** to preview the page.
13. Select **Finish editing** to check in the template, and then select **Publish** to publish it.

## Additional resources

[Module library overview](#)

[Navigation menu module](#)

[Site selector module](#)

[Overview of default category landing page and search results page](#)

[Product collection modules](#)

[Buy box module](#)

[SDK and module library updates](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Navigation menu module

2/18/2021 • 3 minutes to read • [Edit Online](#)

## IMPORTANT

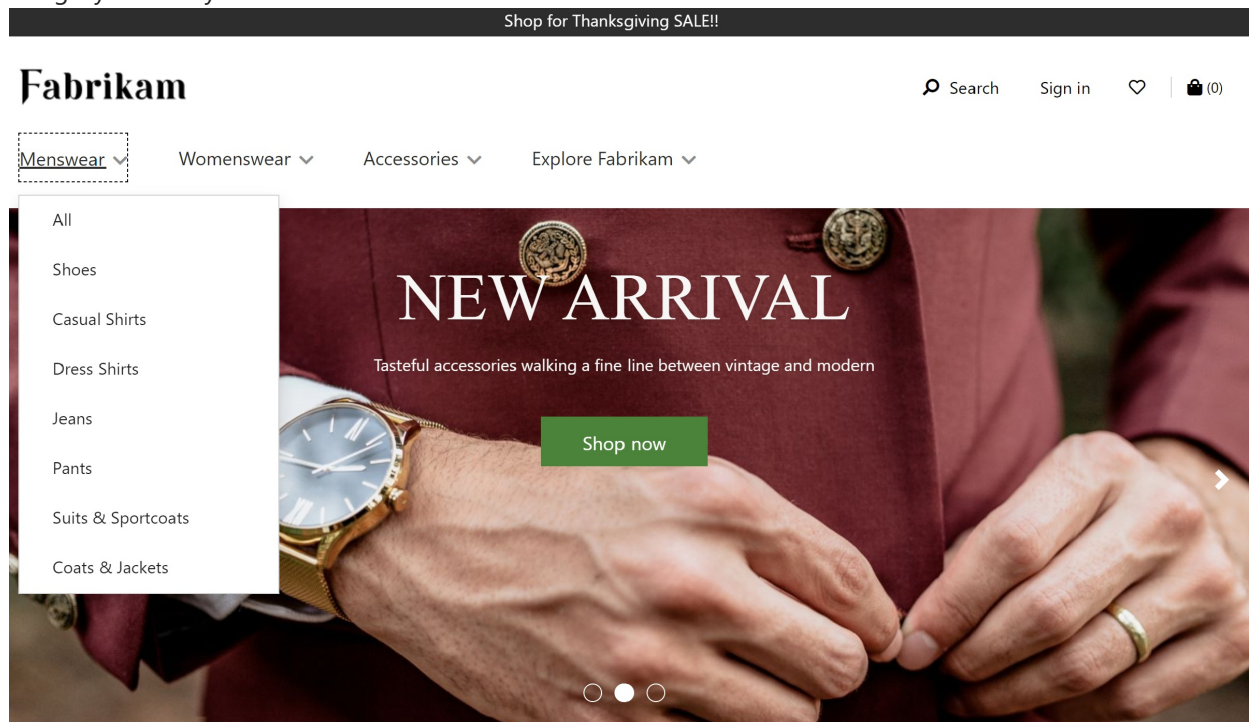
Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic covers navigation menu modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

The primary purpose of navigation menu modules is to allow site users to browse products and site pages according to the channel navigation hierarchy defined in Dynamics 365 Commerce headquarters. Items configured in a navigation menu module appear as site header navigation. Navigation menu modules also support static menu items that link to other pages on an e-Commerce site.

The navigation menu module can be added to the header module of a page. In the Fabrikam theme, the navigation menu shows two levels by default. In the Starter theme, the navigation menu shows three levels by default. To change to the number of levels, a view extension is required on the theme.

The following illustration shows an example of a navigation menu for the Fabrikam site with two levels of category hierarchy and some static menu items.



## Navigation menu module properties

PROPERTY NAME	VALUE	DESCRIPTION
---------------	-------	-------------

PROPERTY NAME	VALUE	DESCRIPTION
Source	<b>Retail, Manual authoring, Retail and manual authoring</b>	The <b>Retail</b> value allows the channel navigation hierarchy from Commerce headquarters to be displayed on the navigation menu. The <b>Manual authoring</b> value allows static menu items to be curated. The <b>Retail and manual authoring</b> value allows a mix of both.
Show category images	<b>True or False</b>	When enabled, this property displays category images on the navigation menu as defined in Commerce headquarters for each category. Added in Commerce release 10.0.14.
Show promotions	<b>True or False</b>	When this property is enabled, promotions can be configured by using images, links, and text. This property was added in the Commerce version 10.0.17 release.
Add promotions	Text, image, or link	When the <b>Show promotions</b> property is enabled, you can add text, an image, or a link as promotional content on the navigation menu.
Enable multi-level navigation menu	<b>True or False</b>	When this property is enabled, the navigation menu can show multiple levels of the navigation hierarchy. This feature is available in the Commerce version 10.0.15 release.
Number of levels	integer	This property defines the numbers of levels that should be shown if the <b>Enable multilevel navigation menu</b> property is set to <b>True</b> .
Static menu item	Array of values	Static menu items that associate a menu item name with a link to a static site page. You can create menu items below other menu items. By default, static menus appear at the root level and will be appended to the channel navigation hierarchy if it exists.
Show root menu	<b>True or False</b>	When this property is enabled, the navigation menu can be defined under a custom root (for example, <b>Shop now</b> ). This feature is available in the Dynamics 365 Commerce 10.0.15 release.
Root menu	string	This property can be used to define text for a custom root if the <b>Show root menu</b> property is set to <b>True</b> .

The following illustration shows an example of a category image displayed on the navigation menu for the Fabrikam site.

Menswear ▼ Accessories ▼ Womenswear ▼ Contact ▼

- All
- Shoes
- Casual Shirts
- Coats & Jackets
- Pants
- Dress Shirts
- Jeans
- Suits & Sportcoats



## Add a navigation menu module to a header module

For details about how to add a navigation menu module to a header module, see [Header module](#).

## Additional resources

[Module library overview](#)

[Breadcrumb module](#)

[Site selector module](#)

[Buy box module](#)

[Cookie compliance](#)

[Header module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Site selector module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the site selector module and describes how to add it to site pages in Microsoft Dynamics 365 Commerce.

When a business has different sites across markets, regions, and locales, site users need an easy way to switch between sites and select their preferred shopping site. To accommodate this scenario, the site selector module lets users browse across multiple sites.

The site selector module must be configured with the list of sites (markets, regions, or locales) that site users can browse.

## NOTE

The site selector module is available in the Dynamics 365 Commerce 10.0.14 release.

The following illustration shows an example of a site selector module that is featured in the header of a site page.



## Site selector module properties

PROPERTY NAME	VALUE	DESCRIPTION
Heading	Text	The heading for the module.
Site options	Name, Image, URL	This property specifies a name, a link to the site's home page, and an optional image to show for each site that is included in the module. The image can be a flag, or some representation of a market, region, or locale.

## Add a site selector module to a page

The site selector module can be added to the [Header module](#) under the site selector slot. After it's added, you can define the module heading and site options.

## Additional resources

[Module library overview](#)

[Header module](#)

[Breadcrumb module](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Search results module

2/18/2021 • 3 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic covers search results modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

The search results module returns product search results and a list of applicable refiners for the products. Search results modules on Dynamics 365 Commerce sites can be used to render pages for the following scenarios:

- Search results that are initiated by a user search
- Search results that show a specific set of products, such as "Shop similar looks"
- Lists of products that belong to a category

For more information about category and search results pages, see [Default category landing page and search results page overview](#).

The following illustration shows an example of a search results page for a category on the Fabrikam site.

**Color** -

- Black
- Blue
- Grey
- Red
- Violet
- White

**Size** +

**Style** +

**Gender** +

**Fabric** -

- Cashmere
- Nylon
- Organic cotton
- Satin

**Fit type** -

- Baggy
- Regular
- Relaxed
- Slim
- Tailored










**Fly type** +

**Category** +

**Price** -

●-----●

\$90 \$150

	<p><b>Striped Dress</b></p> <p><b>\$100.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>V-Neck Dress</b></p> <p><b>\$120.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>The Red Dress</b></p> <p><b>\$150.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>Floral print dress</b></p> <p><b>\$90.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>Red Bow Tie Dress</b></p> <p><b>\$140.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>Floral White Dress</b></p> <p><b>\$100.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>Violet Party Dress</b></p> <p><b>\$110.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>Grey Party Dress</b></p> <p><b>\$110.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
	<p><b>Grey Sheen Dress</b></p> <p><b>\$100.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>

## Search results module properties

The following table lists the properties of search result modules, together with their values and descriptions.

PROPERTY	VALUES	DESCRIPTION
Items per page	Integer	The number of items that should be shown on each page.



PROPERTY	VALUES	DESCRIPTION
Allow back on PDP	<b>True</b> or <b>False</b>	If this property is set to <b>True</b> , when a user selects a product on the search results page, the breadcrumb navigation on the product details page (PDP) that is opened will show a "Back to results" link.
Expand Refiners	<b>All</b> , <b>1</b> , <b>2</b> , <b>3</b> , or <b>4</b>	The number of top refiners that should be expanded when a page is loaded. For example, if this property is set to <b>3</b> , the first three refiners on the page will be expanded.
Hide category hierarchy display	<b>True</b> or <b>False</b>	If this property is set to <b>True</b> , the category hierarchy display on the page will be hidden. This property should be set to <b>True</b> if you're using the <a href="#">breadcrumb module</a> to show the category hierarchy.
Include product attributes in search results	<b>True</b> or <b>False</b>	If this property is set to <b>True</b> , attributes will be returned for the products in the search results. Although these attributes can be shown on a Commerce site, an extension is required.
Show affiliation prices	<b>True</b> or <b>False</b>	If this property is set to <b>True</b> , affiliation prices for products will be shown in the search results when a signed-in user browses the page.

#### IMPORTANT

In the Dynamics 365 Commerce 10.0.16 release and later, the **Show affiliation prices** configuration can be used to show affiliation prices on the page.

## Supported modules

The search results module supports the [quick view module](#), which lets users view product information and add items to the cart from the search results page.

## Add a search results module to a category page

To add a search results module to a category page, follow these steps.

1. Go to **Templates**, and select **New** to create a new template.
2. In the **New template** dialog box, enter the name **Search results**, and then select **OK**.
3. In the **Body** slot, select the ellipsis (...), and then select **Add Module**.
4. In the **Add Module** dialog box, select the **Default Page** module, and then select **OK**.
5. In the **Main** slot of the **Default Page** module, select the ellipsis (...), and then select **Add Module**.
6. In the **Add Module** dialog box, select the **Container** module, and then select **OK**.
7. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.

8. In the **Add Module** dialog box, select the **Breadcrumb** module, and then select **OK**.
9. In the **Breadcrumb** properties pane, enter the value **1** for **Min Occurs**.
10. In the **Container** slot, select the ellipsis (...), and then select **Add Module**.
11. In the **Add Module** dialog box, select the **Search results** module, and then select **OK**.
12. In the **Search results** properties pane, enter the value **1** for **Min Occurs**, and then set any other required properties for the search results module. By setting these properties in the template, you ensure that any customizations to a specific category page will automatically include these settings.
13. Select **Finish editing**, and then select **Publish** to publish the template.
14. Go to **Pages**, and select **New** to create a new page.
15. In the **Choose a template** dialog box, select the **Search results** template that you created, enter **Category page** for **Page name**, and then select **OK**. Because all the values are set in the template, the page is ready to be published.
16. Select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Additional resources

[Default category landing page and search results page overview](#)

[Module library overview](#)

[Quick view module](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cookie consent module

2/18/2021 • 2 minutes to read • [Edit Online](#)

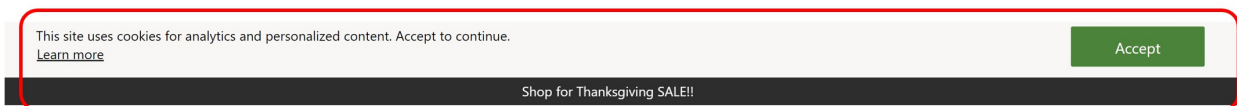
This topic covers cookie consent modules and describes how to add them to site pages in Microsoft Dynamics 365 Commerce.

The cookie consent module prompts site users to explicitly provide consent to allow cookies for any feature or module that tracks browser cookies. The consent is required the first time a site user browses a site in a new browser session. When consent is received, it is tracked and the site user will not be prompted for consent again. For more information, see [Cookie compliance](#).

If site user cookie consent is not received, any features or modules that require cookie consent will not be rendered on the page. For example, the checkout module, social share module, and preferred store feature all require cookie consent and will not be rendered if site user consent is not received.

A cookie consent module can be configured on a page's header fragment so that it can be enforced when the page loads. The cookie consent module should have a clear message informing the site user about cookie usage on the site and should provide a link to the site's privacy page.

The following illustration highlights an example of a cookie consent message with a link to the site's privacy policy page displayed on the header of a site page.



**Fabrikam**

Search Sign in | (0)

Menswear ▾ Womenswear ▾ Accessories ▾ Gift cards Contact ▾

## NEW ARRIVALs

Into the summer breeze with all new designer dresses

Shop now



## Cookie consent module properties

PROPERTY NAME	VALUE	DESCRIPTION
Rich Text	Rich Text	Specifies a Rich Text notification to site users that the site uses browser cookies and that users should accept the use of cookies for the site to be fully functional.

PROPERTY NAME	VALUE	DESCRIPTION
Links	URL	One or more links can be added to a site's privacy page that describes the types of cookies that are tracked on the site.

## Add a cookie consent module to site pages

To efficiently add a cookie consent module to multiple site pages, it can be added to a shared page header fragment. After the header fragment is added to all site pages, a cookie consent notification will automatically be rendered the first time a site user navigates to any site page.

For more information about header fragments and modules, see [Header module](#).

## Additional resources

[Module library overview](#)

[Header module](#)

[Cookie compliance](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Default category landing page and search results page overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides an overview of the default category landing page and search results page in Microsoft Dynamics 365 Commerce e-Commerce.

## Default category landing page

The default category landing page is the page that website users typically are taken to when they select a category in the navigation hierarchy. The category page lets you browse, and you can also sort and refine the categorized products.

**Color** -

Black

Blue

Grey

Red

Violet

White

---

**Size** +

---

**Style** +

---

**Gender** +

---

**Fabric** -

Cashmere

Nylon

Organic cotton

Satin

---

**Fit type** -

Baggy

Regular

Relaxed

Slim

Tailored

---

**Fly type** +

---










**Category** +

---

**Price** -

●──●

\$90 \$150

		
<p><b>Striped Dress</b></p> <p><b>\$100.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>	<p><b>V-Neck Dress</b></p> <p><b>\$120.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>	<p><b>The Red Dress</b></p> <p><b>\$150.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
		
<p><b>Floral print dress</b></p> <p><b>\$90.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>	<p><b>Red Bow Tie Dress</b></p> <p><b>\$140.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>	<p><b>Floral White Dress</b></p> <p><b>\$100.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>
		
<p><b>Violet Party Dress</b></p> <p><b>\$110.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>	<p><b>Grey Party Dress</b></p> <p><b>\$110.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>	<p><b>Grey Sheen Dress</b></p> <p><b>\$100.00</b></p> <p>Our dresses will make you feel and look your best. Every piece is selected for beauty that sets it apart.</p>

At the top of the page is a header that shows all the product categories and other pages that the merchandising manager has categorized. Configuration is done as part of the configuration of the channel navigation hierarchy. At the bottom of the page is a footer that includes quick links to various topics that a shopper might be interested in.

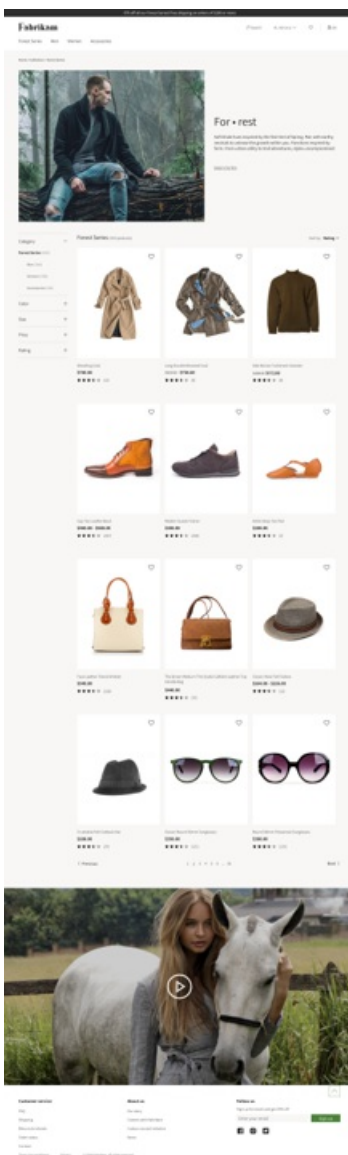
The following components are essential for a category:

- **Product placement tiles** show the products that the merchandising manager has defined in a category as part of the configuration of the navigation hierarchy.
- **Refiners and choice summary** are filters that provide counts and that can be used to refine items. The merchandising manager configures them as part of the configuration of the metadata related to channel categories and product attributes.

- **Sorting options** are used by website visitors to sort the products. By default, the following sorting options are available:
  - Price – low to high
  - Price – high to low
  - Product name – [A-Z]
  - Product name – [Z-A]
  - Ratings – low to high
  - Ratings – high to low
- **Pagination** lets website visitors move from one page of categorized product results to another page.
- **Total count** provides the total number of products that are defined in a category.

## Enrich a category landing page

If you want a category landing page to have a more tailored experience for a specific category, you can "enrich" the category landing page for that category. For example, you can add a marketing video and some category storytelling to get the shopper's attention. For more information, see [Enrich a category landing page](#).



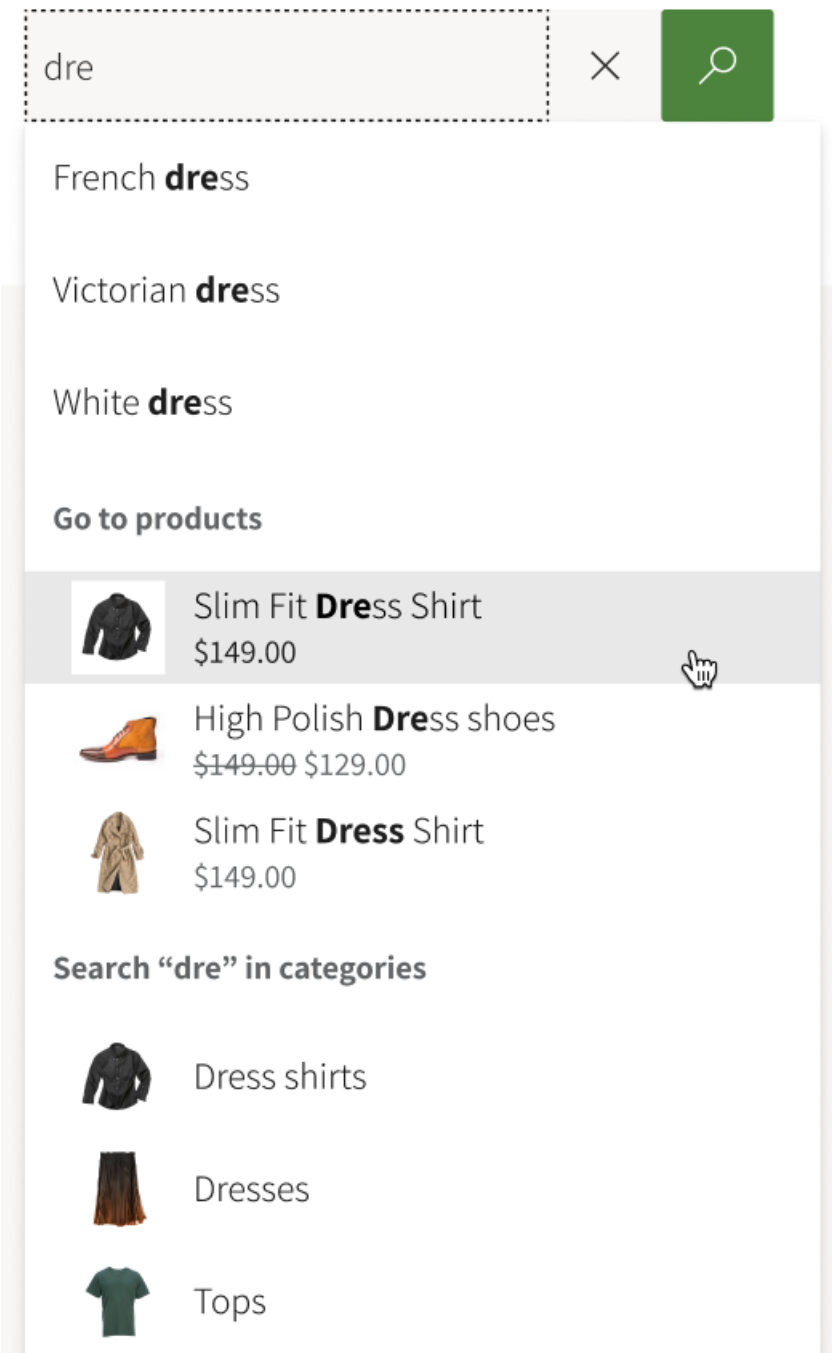
## Auto-suggest and search results pages

Website users can explore a site either by going to a category from the navigation hierarchy or by entering a search term in the search field.

As soon as users start to type in the search field, they experience the immersive auto-suggest functionality that suggests search terms.

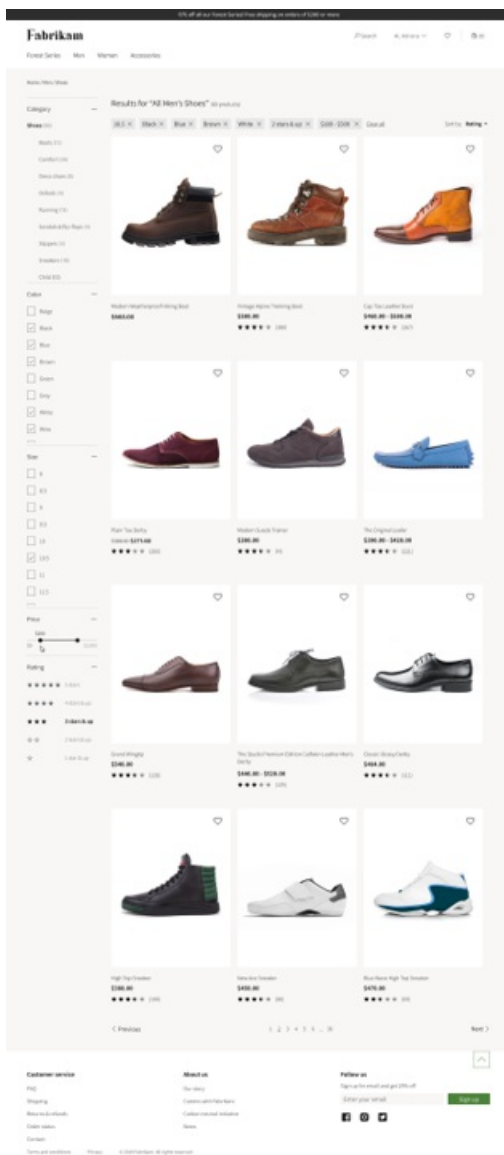
Here are some of the types of suggestions that might be shown:

- **Keywords** are used to find items across all products that are assorted to the channel.
- **Products** provide direct links to the product details page.
- **Scoped category search suggestions** list various categories and let users search for the keyword in a specific category.



When users select one of the keyword or scoped category search suggestions, or when there are no suggestions for the search term that they enter, they are redirected to a search results page. The users can then browse, sort, and refine the list of search results to find the desired item.





The following components are essential for a search results page:

- **Product placement tiles** show the products for the user's search. By default, these tiles are sorted by the cloud-powered search relevancy score for the user search.
- **Refiners and choice summary** are filters that provide counts and that can be used to refine items. The merchandising manager configures them as part of the configuration of the "channel categories and product attributes" metadata.
- **Sorting options** are used by website visitors to sort the products. By default, the following sorting options are available:
  - Price – low to high
  - Price – high to low
  - Product name – [A-Z]
  - Product name – [Z-A]
  - Ratings – low to high
  - Ratings – high to low
  - Default
- **Pagination** lets website visitors move from one page of categorized product results to another page.
- **Total count** provides the total number of products that are defined in a category and that match the search criteria.

## NOTE

These cloud-powered search capabilities are available starting in version 10.0.8. Ensure that under **Commerce Parameters > Configuration Parameters** there is an entry for "ProductSearch.UseAzureSearch set to 'true'".

### Commerce parameters

✓ Name ↑	Value	System maintai...	Description
EcoResProductVariantDimensio...	true	<input type="checkbox"/>	
✓ ProductSearch.UseAzureSearch	true	<input type="checkbox"/>	Enable cloud powered search experiences for POS & eCommerce
✓ RatingsAndReviews.EnableProd...	true	<input type="checkbox"/>	

## Additional resources

[Cloud-powered search overview](#)

[Home page overview](#)

[Product details pages overview](#)

[Cart and checkout pages overview](#)

[Account management pages overview](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cloud-powered search overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic gives an overview of cloud-powered search in Microsoft Dynamics 365 Commerce.

## Overview

Product discoverability helps guarantee that customers can quickly and easily find products by browsing categories, searching, and filtering. Retailers consider product discovery a primary tool for customer interaction across all channels.

Customers are accustomed to the nearly instantaneous response times of web search engines, sophisticated e-Commerce websites, social apps, automatic suggestions that appear as they type search terms, faceted navigation, and highlighting. If customers can't find the product that they are looking for quickly enough in one e-Commerce store, they won't hesitate to go to a different e-Commerce store.

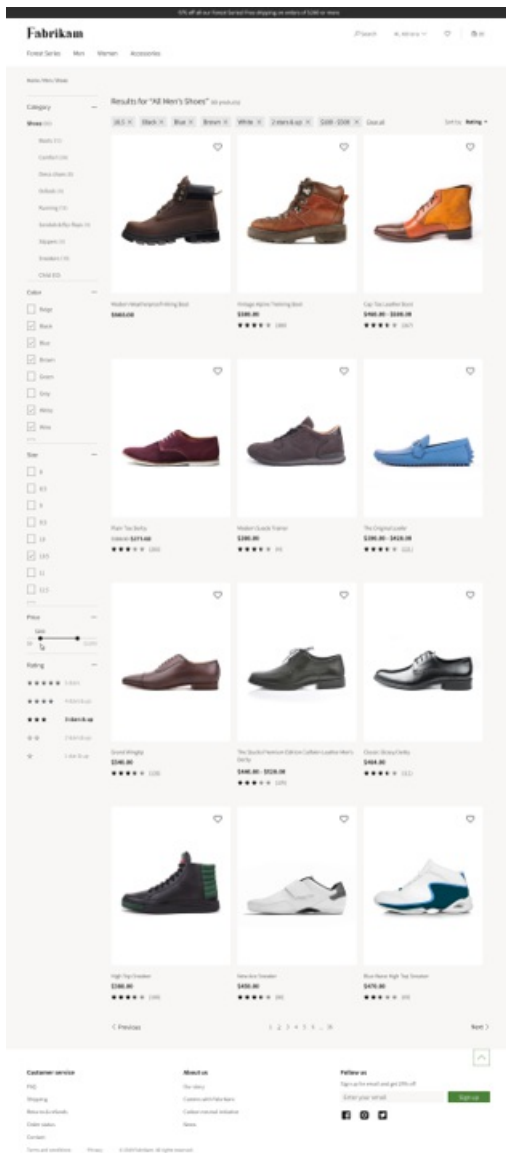
The cloud-powered product discoverability in Dynamics 365 Commerce helps retailers continue to increase consumer retention and conversion rates across all channels, both e-Commerce channels and point of sale (POS) channels.

The Dynamics 365 Commerce search experience has improved capabilities to help retailers achieve better product discoverability. At the same time, these capabilities deliver the scalability and performance that are required for e-Commerce traffic.

## Browse and search

Search relevance and performance are key factors in the omnichannel experience, because product discovery relies primarily on search functionality for information retrieval and content navigation. An effective and efficient browse and search experience helps increase conversion.

The following illustration shows an example of typical browse and search functionality.

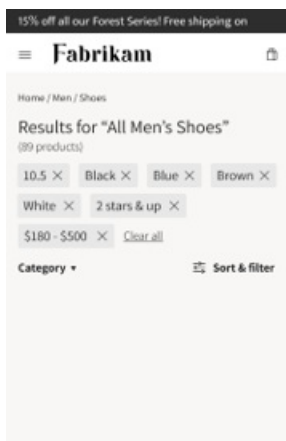


## Faceted navigation and choice summary

Faceted navigation helps customers more easily browse for content by letting them filter on refiners that are linked to terms in a term set. After a customer has selected and applied refiners, a summary of the choices is shown.

By using faceted navigation, you can configure different refiners for different terms in a term set, without having to create additional pages.

The following illustration shows an example where faceted navigation is used in a search.

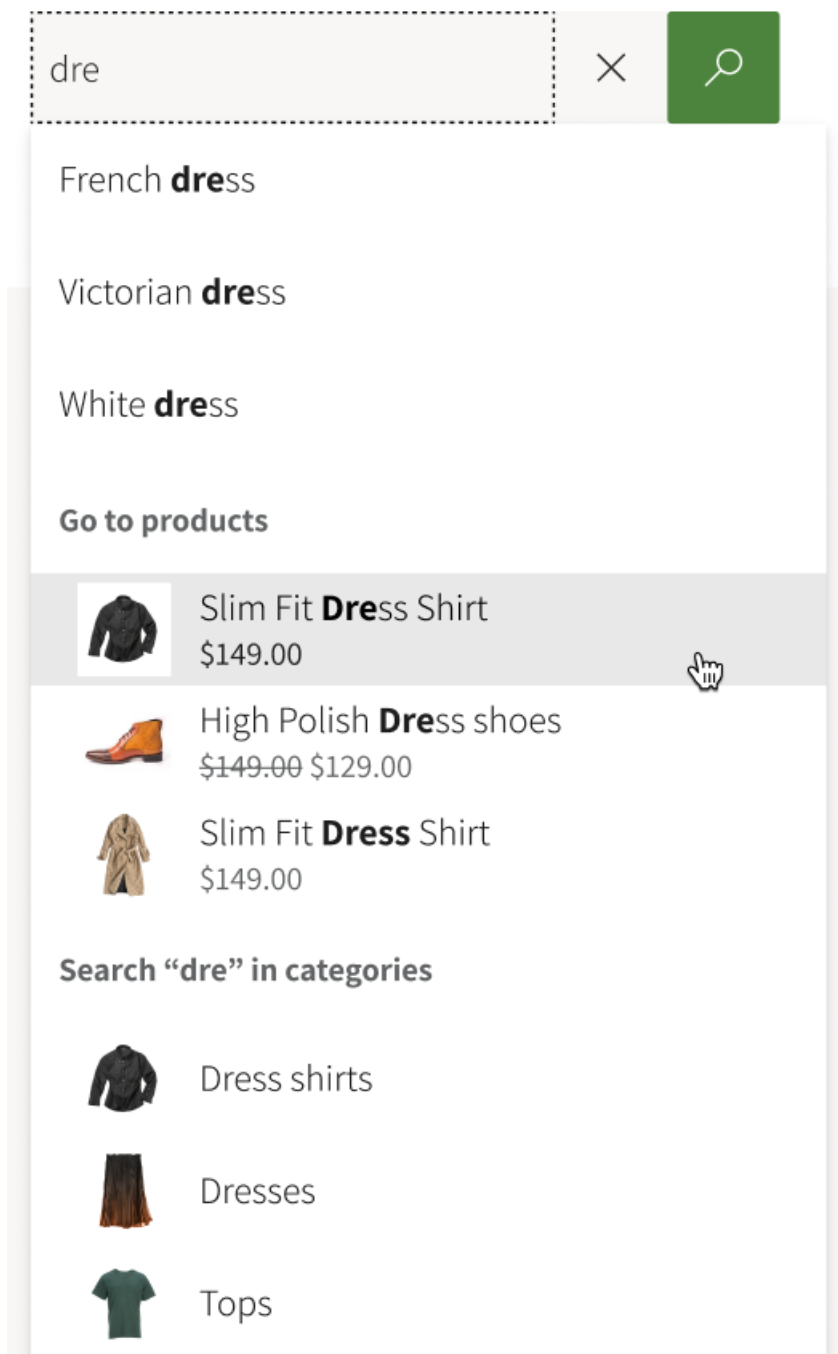


# Immersive autosuggest

Current autosuggest functionality just shows keywords that trigger a search for the matching keyword. Because of new enhancements in Dynamics 365 Commerce, customers can often discover links to products before they have finished typing.

Dynamics 365 Commerce also supports functionality for keyword matches in various categories. This functionality lets customers see the number of matching keywords across categories and trigger a search for a keyword in other categories.

The following illustration shows an example where immersive autosuggest is being used.



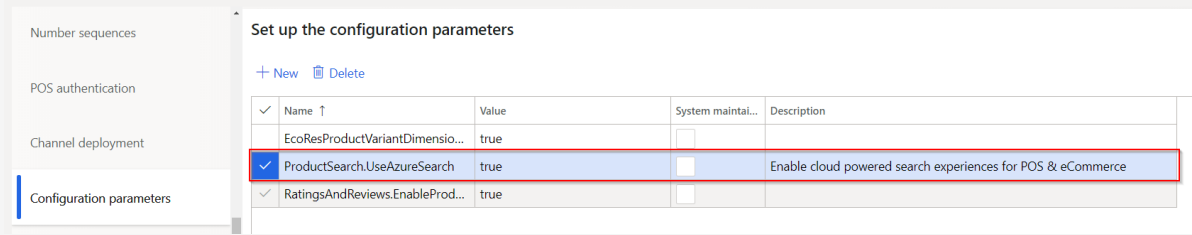
## Sort

Enhanced sorting in Dynamics 365 Commerce lets customers sort, search, and browse search results, and refine them by criteria such as price, product name, and product number. Customers can also sort results based on whether a product is new, top-selling, or recently added.

## NOTE

These cloud-powered search capabilities are available starting in version 10.0.8. Ensure that under **Commerce Parameters > Configuration Parameters** there is an entry for "ProductSearch.UseAzureSearch set to 'true'".

### Commerce parameters



The screenshot shows a configuration interface for Commerce parameters. On the left, there is a sidebar with categories: Number sequences, POS authentication, Channel deployment, and Configuration parameters (selected). The main area is titled 'Set up the configuration parameters' and contains a table with columns: Name, Value, System maintai..., and Description. The table has three rows, with the second row highlighted in blue and a red border around it.

✓ Name ↑	Value	System maintai...	Description
EcoResProductVariantDimensio...	true	<input type="checkbox"/>	
✓ ProductSearch.UseAzureSearch	true	<input type="checkbox"/>	Enable cloud powered search experiences for POS & eCommerce
✓ RatingsAndReviews.EnableProd...	true	<input type="checkbox"/>	

## Additional resources

[Default category landing page and search results page overview](#)

[Manage SEO metadata](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Generate online channel reports

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to generate reports for your online channel in Microsoft Dynamics 365 Commerce.

## Overview

You can generate and view several reports in Commerce to see how your online channel is performing.

## Channel summary report

The **Channel summary** report shows a summary of the following transactions for the selected channel:

- Sales transactions
- Payment transactions
- Tax transactions
- Discounted transactions

To generate a **Channel summary** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Channel summary report**.
2. In the **From date** field, enter a date.
3. In the **To date** field, enter a date.
4. In the **Channel** field, select the online channel.
5. Select **OK**.

## Channel sales by year report

The **Channel sales by year** report shows a comparison of year-over-year sales for a specific store. You select the year to compare the sales against, and the report compares sales for the selected year with sales for the previous year.

To generate a **Channel sales by year** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Channel sales by year report**.
2. In the **From calendar year** field, enter a year.
3. In the **To calendar year** field, enter a year.
4. In the **Channel** field, select the online channel.
5. Select **OK**.

## Channel sales by hour report

The **Channel sales by hour** report shows sales metrics per hour for a selected channel or operating unit.

To generate a **Channel sales by hour** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Channel sales by hour report**.
2. In the **From date** field, enter a date.
3. In the **To date** field, enter a date.

4. In the **Channel** field, select the online channel.
5. Select **OK**.

## Top customers report

The **Top customers** report shows sales metrics for the top  $N$  customers for a selected channel or operating unit. The value  $N$  is a number from 10 to 100 and is based on a user-selected aggregate measure.

To generate a **Top customers** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Top customers report**.
2. In the **From date** field, enter a date.
3. In the **To date** field, enter a date.
4. In the **Channel** field, select the online channel.
5. Select **OK**.

## Top discounts report

The **Top discounts** report shows sales metrics for the top  $N$  discounts for a selected channel or operating unit. The value  $N$  is a number from 10 to 100 and is based on a user-selected aggregate measure.

To generate a **Top discounts** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Top discounts report**.
2. In the **From date** field, enter a date.
3. In the **To date** field, enter a date.
4. In the **Channel** field, select the online channel.
5. Select **OK**.

## Top products report

The **Top products** report shows sales metrics for the top  $N$  products for a selected channel or operating unit. The value  $N$  is a number from 10 to 100 and is based on a user-selected aggregate measure.

To generate a **Top products** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Top products report**.
2. In the **From date** field, enter a date.
3. In the **To date** field, enter a date.
4. In the **Channel** field, select the online channel.
5. Select **OK**.

## Category sales report

The **Category sales** report shows sales metrics over a selected period for each node of a category hierarchy for a selected channel or operating unit.

To generate a **Category sales** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Category sales report**.
2. In the **From date** field, enter a date.
3. In the **To date** field, enter a date.
4. In the **Channel** field, select the online channel.
5. Select **OK**.



# Organization sales report

The **Organization sales** report shows the performance of your stores by organization unit. This report includes the sales quantity and amount by store, and the profit margin for each store. The organization unit is based on the default reporting hierarchy.

To generate an **Organization sales** report, follow these steps.

1. Go to **Retail and Commerce > Inquiries and reports > Sales reports > Organization sales report**.
2. In the **From date** field, enter a date.
3. In the **To date** field, enter a date.
4. Select **OK**.

## Additional resources

- [Commerce home page](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Experimentation in Dynamics 365 Commerce

2/18/2021 • 2 minutes to read • [Edit Online](#)

Use experimentation in Dynamics 365 Commerce to validate hypotheses about the effectiveness of your e-Commerce pages and make decisions with data-driven confidence. Commerce supports A/B testing on pages, modules, and fragments and enables you to measure the impact of proposed changes to your website.

You can create, edit, and manage page and content treatments known as **variations** in Commerce site builder. Commerce integrates with third-party services that you can use to create experiments and treatment assignments. Real-time event streams captured in Commerce enable the analytics that define the experiment results in the third-party service. You can then leverage these analytics to help support or refute your hypothesis.

## Set up prerequisites

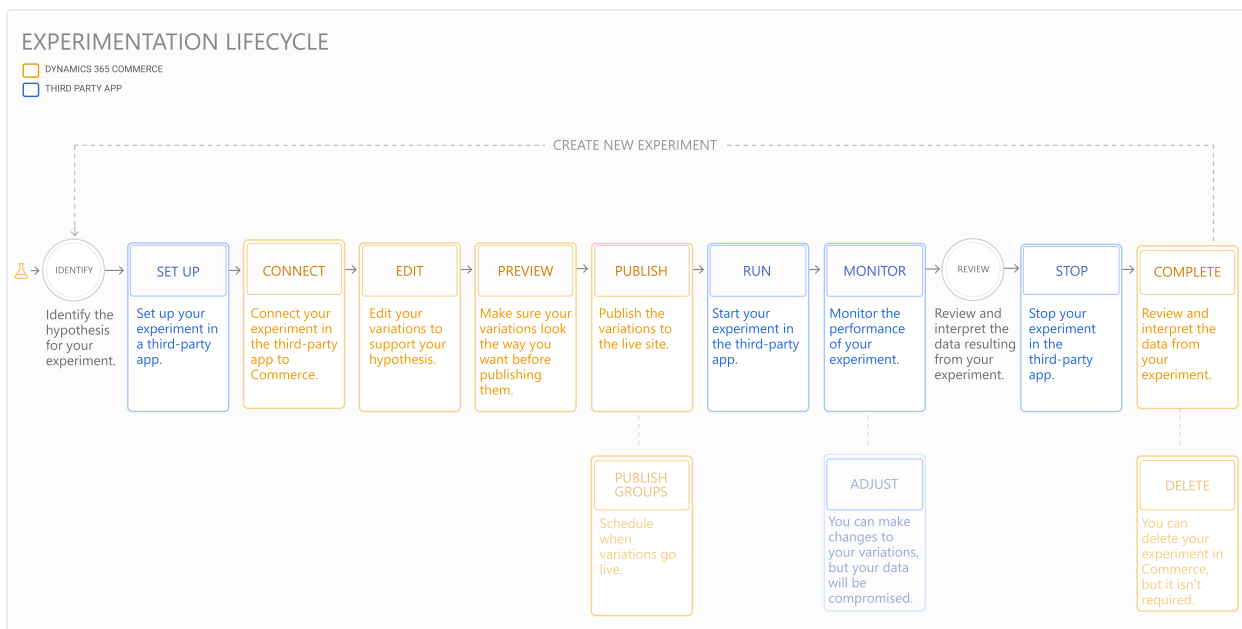
1. **Get the correct version of Commerce** - Upgrade your module library, online channel extensibility software development kit (SDK), and Commerce Scale Unit to Commerce version 10.0.13 or later.
2. **Set up an experimentation connector** - An experimentation connector allows Commerce to connect with third-party services to retrieve the list of experiments and determine when to show an experiment to a user. You can purchase a third-party connector from [AppSource](#). Follow the setup instructions provided by the publisher. You can alternatively use the sample test connector from Commerce to test the experimentation workflow without needing to configure an external service. For more information, see [Configure and enable connectors](#).
3. **Turn on the experimentation feature flags in Commerce** - You can enable experimentation at the tenant level by going to **Tenant Settings > Features** or at the site level at **Site Settings > Features**.
  - Enable the **Experimentation** flag to create experiment variations of modules within a page without affecting or copying other content that isn't part of the experiment. This ensures that ongoing content updates outside the experiment stay in sync during the experiment lifecycle. Disabling this flag stops all experiments from being shown to users and removes all editing functions within site builder.
  - Enable the **Experiment on pages or fragments** flag to run experiments on a page or fragment. This creates a full instance copy of the entire page or fragment for all modules within the page or fragment. Use this mode when you want to test comprehensive content changes, or where synchronizing ongoing content changes across instances isn't a concern. Disabling this flag prevents creation and editing of new experiments on pages and fragments.

### NOTE

The **Experimentation** flag must also be enabled for the **Experiment on pages or fragments** functionality to work.

## Experimentation lifecycle

Setting up an experiment, creating variations, and running an experiment is an iterative process. The diagram below illustrates the experimentation lifecycle in Commerce and the third-party service.



To learn more about each step in the experimentation process, refer to the following topics.

- [Identify a hypothesis and determine metrics for an experiment](#)
- [Set up an experiment](#)
- [Connect and edit an experiment](#)
- [Preview and publish an experiment](#)
- [Run and monitor an experiment](#)
- [Promote a variation and complete an experiment](#)

#### NOTE

To learn where an experiment is in the lifecycle, select **Experiments** in the left navigation pane of site builder. A list of experiments is displayed with the status of each experiment in both Commerce and the third-party service. For more information, see [Review the status of an experiment](#).

## Next step

[Identify a hypothesis and determine success metrics for an experiment](#)

#### NOTE

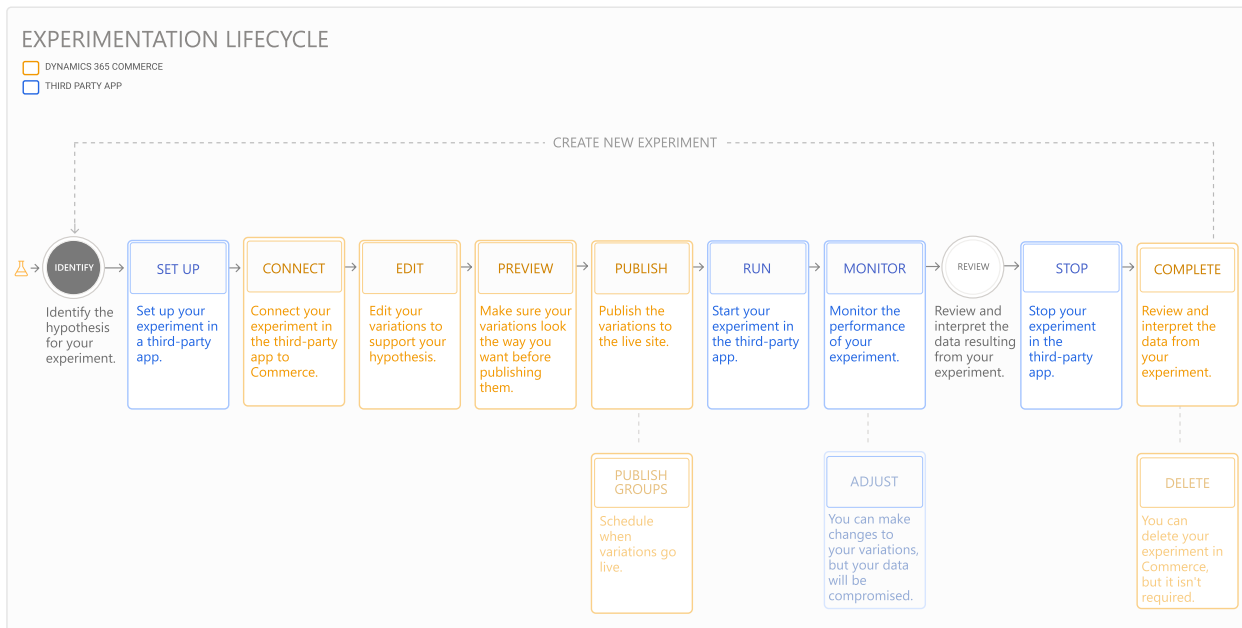
Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Identify a hypothesis and determine success metrics for an experiment

2/18/2021 • 2 minutes to read • [Edit Online](#)

The first phase in the experimentation lifecycle includes identifying the hypothesis for the experiment and determining the metrics you'll track to evaluate success. The following diagram shows all of the steps involved in [setting up and running an experiment](#) on an e-Commerce website in Dynamics 365 Commerce. Additional steps are covered in separate topics.



A hypothesis is a statement where you predict the outcome of the experiment. Many factors go into defining a hypothesis, for example, research about user behavior and website data you've collected. With the hypothesis, you'll define the assumption or theory you want to validate with your experiment. An example of a hypothesis for your experiment may be *"a picture of a white t-shirt on my home page will drive a higher clickthrough rate than a navy sweater during summer months because people want to wear something lightweight and light colored in the summer."* In that case, you'll create variations that include a white t-shirt and a navy sweater, and publish both at the same time.

To validate a hypothesis, the success or failure of an experiment should be directly tied to user actions; for example, if the website user clicks on a link or button. These actions must correspond with events that will be reported to the third-party service tracking the experiment. Over time, the percentage of users that take the action will be tallied as a metric you can use to generate reports and conduct analyses. To review all of the available events and attributes, see [Commerce component events for diagnostics and troubleshooting](#).

## Previous step

[Experimentation in Dynamics 365 Commerce](#)

## Next step

[Set up an experiment](#)

**NOTE**

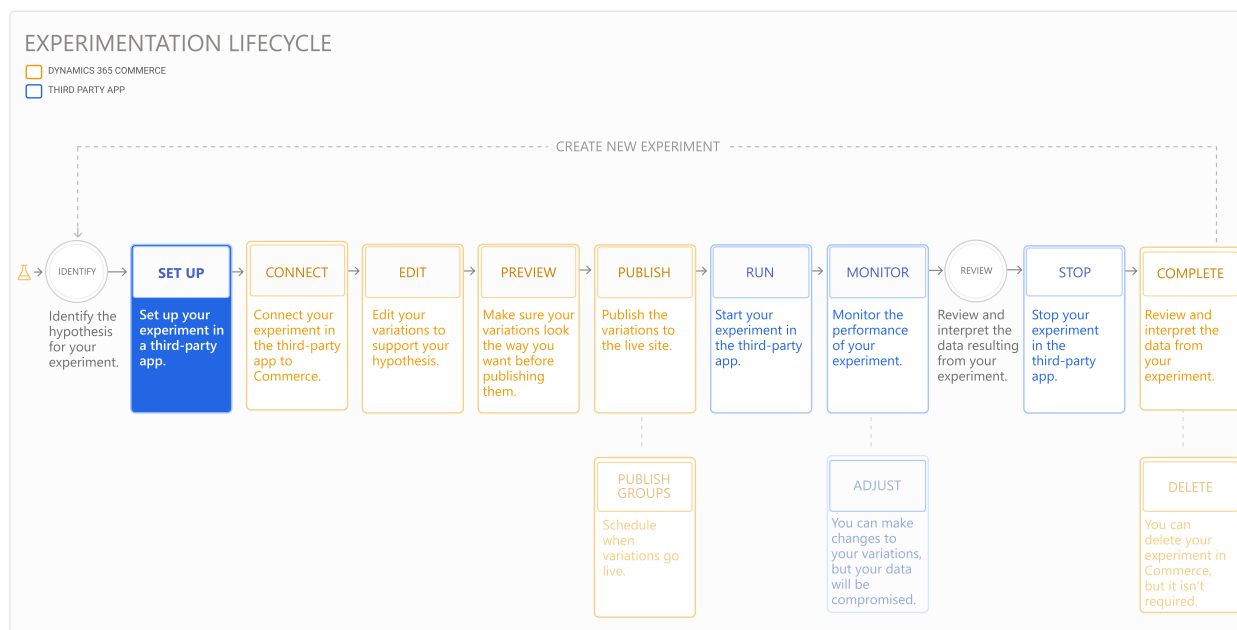
Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up an experiment

2/18/2021 • 2 minutes to read • [Edit Online](#)

After you [define a hypothesis](#) and [determine what success metrics you want to use](#), you'll need to set up your experiment in the third-party service. The following diagram shows all of the steps involved in setting up and running an experiment on an e-Commerce website in Dynamics 365 Commerce. Additional steps are covered in separate topics.



## Set up your experiment in the third-party service

By now you should have chosen your third-party service to run and monitor your experiment, and set up the experimentation connector. These prerequisites are listed in [Experimentation in Dynamics 365 Commerce](#).

Follow the steps required to create your experiment in the third-party service. If the connector is configured properly, the complete list of experiments you set up in the third-party service will appear in Commerce site builder within about 5 minutes.

## Set up your success metrics

Every experiment needs metrics to measure the impact of the variations and to validate the hypothesis. Follow the steps below to enable the computation of metrics in the third-party service using live telemetry events from Dynamics 365 Commerce.

To set up your success metrics, follow these steps.

1. In Commerce site builder, select **Pages** in the left navigation pane, and then select the page that you want to collect metrics for.
2. Go to the **Event IDs to track** section in the right property pane of the page or module you want to track.
3. Select **View**. A list of all event IDs is displayed. Copy the event you want to track, and paste the event key into the designated location in the third-party service. If you need more than one event, copy the keys one at a time.
  - To learn how to view all of the available events and attributes, including page views and revenue tracking, see [Commerce component events for diagnostics and troubleshooting](#).

4. Take any other steps for tracking metrics as required in the third-party service.

## Previous step

[Identify a hypothesis and determine metrics for an experiment](#)

## Next step

[Connect and edit an experiment](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

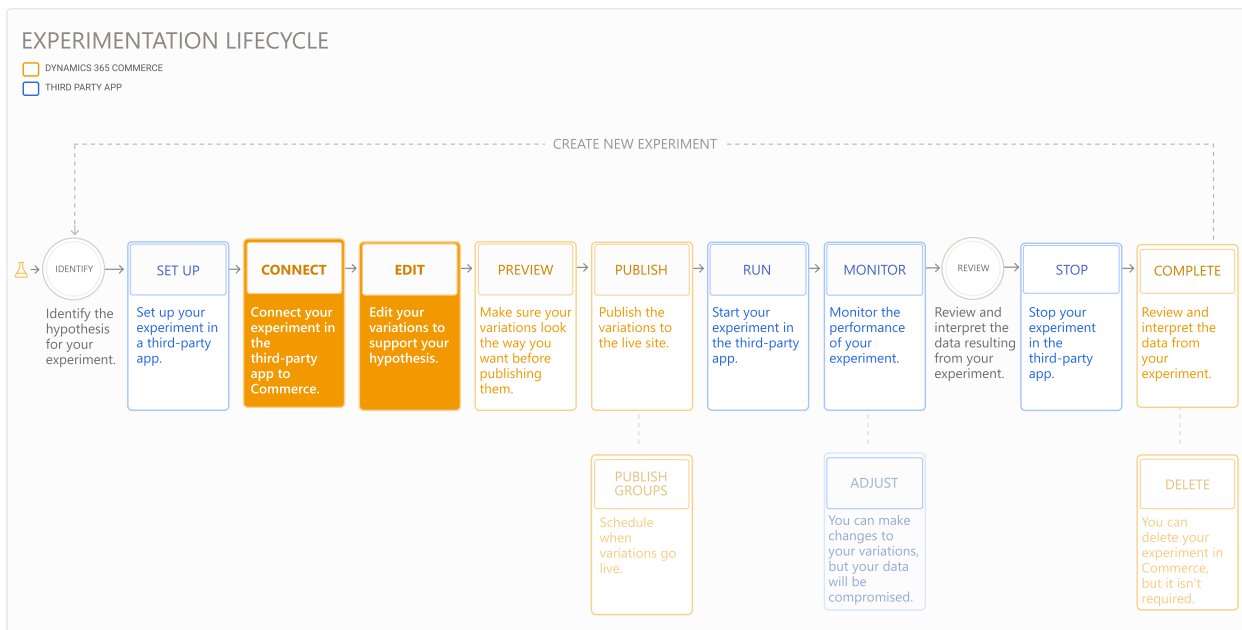
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Connect an experiment and edit variations

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to connect your experiment in Commerce and make changes to your variations so that they align with your hypothesis.

The following diagram shows all of the steps involved in setting up and running an experiment on an e-Commerce website in Dynamics 365 Commerce. Additional steps are covered in separate topics.



After you've [set up your experiment](#) in a third-party service, you'll connect the experiment in Dynamics 365 Commerce and edit the experiment variations.

## Planning considerations

Before you connect your experiment in Commerce, you'll need to make some decisions that impact the way Commerce manages your content.

### Determine the scope of your experiment

When you connect an experiment, you are prompted to define the scope of the experiment. Experiments are defined as **partial** scope or **entire** scope.

- Choose **partial** if you want to conduct an experiment on a specific portion of a page. If you select this option, you must identify which modules are included in the experiment. Changes that are made to parts of the default page or fragment that aren't related to the experiment are automatically synchronized across variations.
- Choose **entire** if you want to conduct an experiment on an entire page or fragment. Separate copies of the default page or fragment are created. You won't have to select which modules are included in the experiment because the whole editing surface is available to change. You can add, delete, or re-order modules as needed. However, if any changes are made to the default page or fragment that the experiment is associated with, those changes have to be manually synchronized across all variations.



#### NOTE

If you associate your experiment with a page that uses a layout, you can only scope the experiment as **entire**.

### Decide if you want to schedule when your experiment is published

If you want to schedule when your experiment is published to your live site, make sure the content you want to associate with the experiment is available in a publish group *before* you connect the experiment.

For more information about publish groups, refer to [Work with publish groups](#).

## Connect your experiment

To connect your experiment, you'll launch the **Connect experiment** wizard. The wizard walks you through the steps required to connect your experiment. When you complete the wizard, your experiment is connected and variations are created and ready to be edited.

To get started connecting your experiment in Commerce site builder, follow these steps.

1. To launch the **Connect experiment** wizard, select **Experiments** in the left navigation pane, and then select **Connect**. Alternatively, you can access the wizard from a page or fragment editor by editing it and selecting **Connect experiment** on the command bar.

#### NOTE

A page can only be connected to one experiment at a time. To connect a page to a different experiment, first delete the experiment that the page is currently connected to.

2. Choose the page or fragment you want to run your experiment on.
3. Set the experimentation scope to **partial** or **entire**, based on the choice you made in the [Determine the scope of your experiment](#) section above.

#### NOTE

The **Experiment on pages or fragments** feature flag must be enabled if you want to experiment on a full page or fragment. Refer to the [Experimentation in Dynamics 365 Commerce](#) topic for more information.

4. In the final step of the wizard, select **Generate variations and exit wizard**. Variations are created for the experiment.

## Edit your variations

When you exit the wizard, variations are created for you.

Next, you'll edit the variations so they reflect the choices that you need to verify in the experiment hypothesis. Choose one of following procedures that corresponds to the scope you chose for your experiment in the [Determine the scope of your experiment](#) section above.

### Edit variations for experiments with partial scope

Follow these steps if you defined the scope of your experiment as **partial** in the **Connect experiment** wizard.

1. In editor view, use the variations drop-down menu below the command bar to edit each variation based on your original hypothesis. You may also want to establish a control or base variation by leaving one of the variations unchanged.

2. Select the module to be experimented on, select the ellipsis (...), and then select **Add to experiment**.

### **Edit variations for experiments with entire scope**

If you defined the scope of your experiment as **entire** in the **Connect experiment** wizard, while in editor view, use the variations drop-down menu below the command bar to edit each variation based on your original hypothesis.

#### **NOTE**

In either case, you may also want to establish a control or base variation by leaving one of the variations unchanged.

## Previous step

[Set up an experiment](#)

## Next step

[Preview and publish an experiment](#)

#### **NOTE**

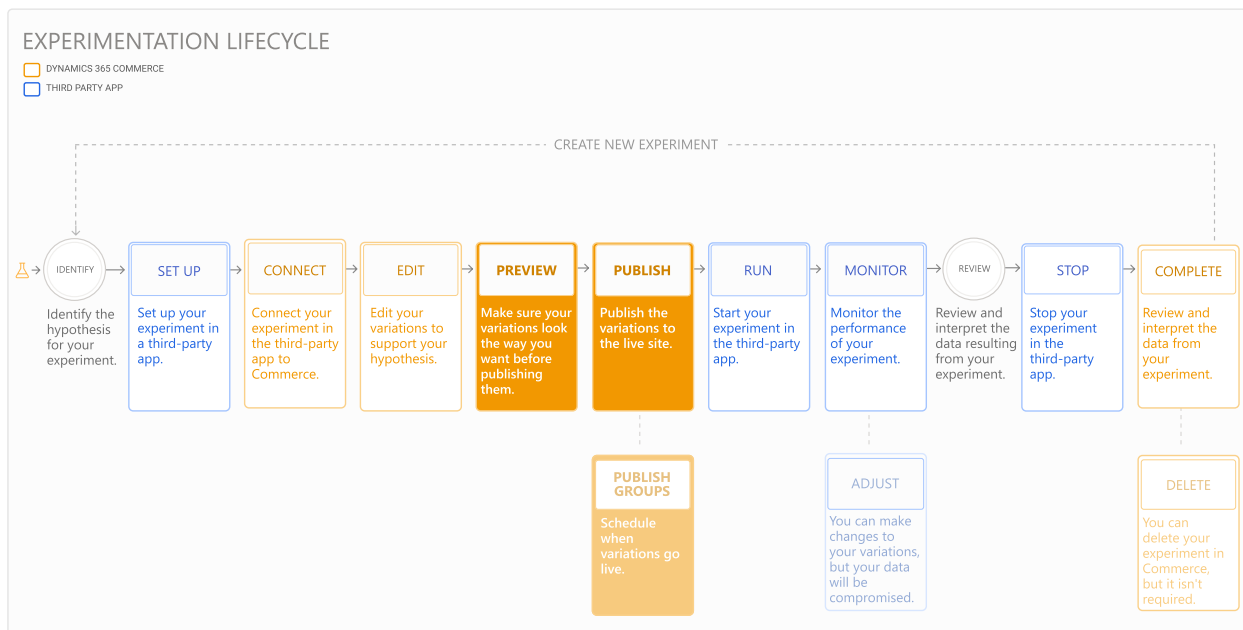
Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Preview and publish an experiment

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to preview and publish your experiment in Dynamics 365 Commerce after you've [connected your experiment and edited your variations](#). The following diagram shows all of the steps involved in setting up and running an experiment on an e-Commerce website in Dynamics 365 Commerce. Additional steps are covered in separate topics.



## Preview your experiment variations

You can preview your variations and continue editing them until they look the way you want them to.

To preview your experiment variations in Commerce site builder, follow these steps.

1. From the variations drop-down menu below the command bar, select the content you want to preview.
2. On the command bar, select **Preview**. A preview of what the content will look like when it's published is displayed.
3. To preview a different variation, select it from the variation drop-down menu and select **Preview** again.

## Publish your experiment

If you aren't using a publish group to schedule when your experiment goes live and you want to publish immediately, select **Publish** in the command bar. All variations that belong to the experiment will be published.

### IMPORTANT

If the page has an unpublished URL, you must first publish the URL or it won't be visible to your website users. For details, see [Save, preview, and publish a page](#).

### Use publish groups to schedule when your experiment goes live

Variations created in site builder can be scheduled for publishing by using a publish group. Within a publish group, you can connect a page or fragment to your experiment by selecting **Experiments** in the left navigation pane. You can also do this by selecting **Pages** or **Fragments** and following the instructions in [Connect an](#)

[experiment and edit variations](#). For information about publish groups, see [Work with publish groups](#).

When using publish groups with experiments, there are some important considerations to be aware of.

- When you add a page or fragment that has an experiment running on it to a publish group, the experiment will be removed from the page or fragment in the publish group.
- Experiments that are connected to pages in a live site aren't available to pages within publish groups and vice-versa. Similarly, pages that have experiments running on them in a live site aren't available to other experiments in publish groups and vice versa.
- When you publish or schedule a publish group, all content in the publish group is published, regardless of whether there's an experiment associated with the publish group.
- Because a publish group continues to persist after it's been published to a live site, experiments in the publish group will also persist. Therefore, you won't be able to associate other experiments with the same page or fragment. To avoid this limitation, delete any publish groups with persisting experiments. Similarly, if you want to delete an experiment in a live site that also exists in a publish group, delete it from the publish group first.

## Previous step

[Connect and edit an experiment](#)

## Next step

[Run and monitor an experiment](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

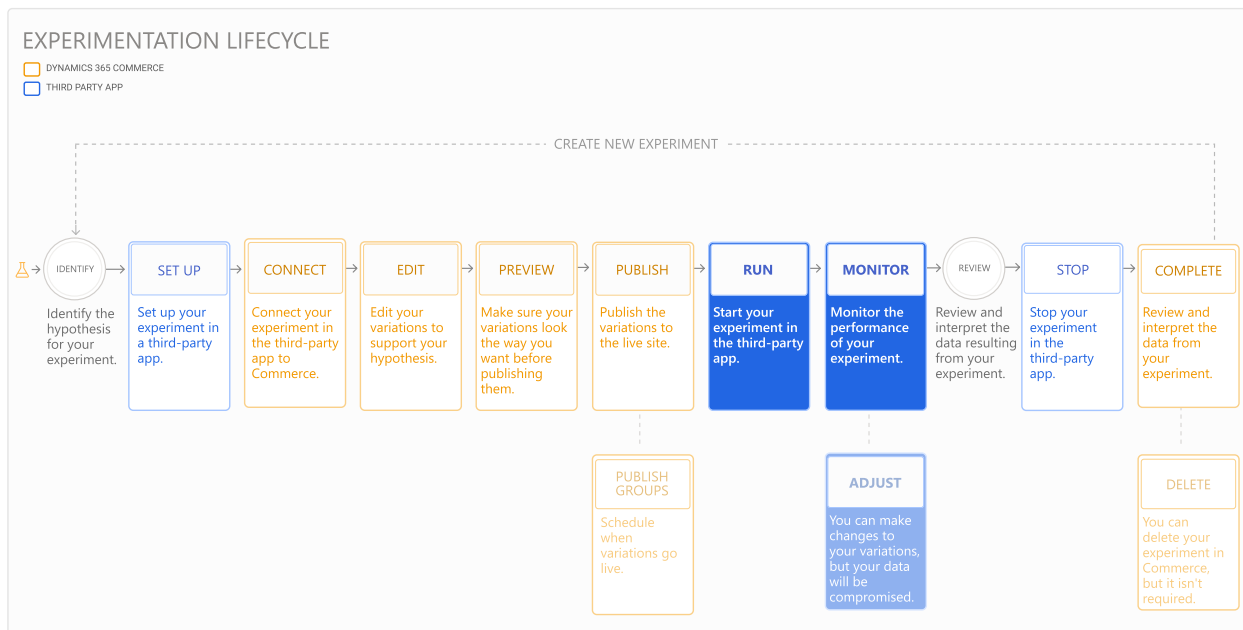
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Run and monitor an experiment

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to run and monitor your experiment in a third-party app, and change variations if needed. Before you complete the steps in this topic, you'll first need to [publish](#) your experiment in Commerce.

The following diagram shows all of the steps involved in setting up and running an experiment on an e-Commerce website in Dynamics 365 Commerce. Additional steps are covered in separate topics.



After you publish your variations, all of the steps you need to do in Commerce to run your experiment are complete. The next step is determining which variation to show to each user when they request a page. The third-party service makes that determination, but first you have to activate the experiment within the service. Since the steps for activating an experiment vary from service to service, you'll need to follow the instructions included with your service or provider. If the experiment is not activated, users will only see the default version of the page (no variations will be displayed).

You'll need to keep the experiment running long enough to gather data for statistically valid results. Use the third-party service to monitor the experiment-related data and analytics while the experiment is running.

## Adjust your variations

If for any reason you need to modify your variations, follow the steps below.

### IMPORTANT

If you make changes to a live experiment in Commerce or the third-party service, your results may be significantly impacted. Consider letting the experiment run its course and then creating a new experiment for major changes.

1. In Commerce site builder, select **Experiments** in the left navigation pane, and then select the experiment.
2. Select the variation you want to update from the drop-down menu.
3. Make any needed changes, and then preview and publish the variations. For more information, see [Preview and publish an experiment](#).
4. Go to the third-party service to make any experiment setup-related changes.

## Previous step

[Preview and publish an experiment](#)

## Next step

[Promote a variation and complete an experiment](#)

### **NOTE**

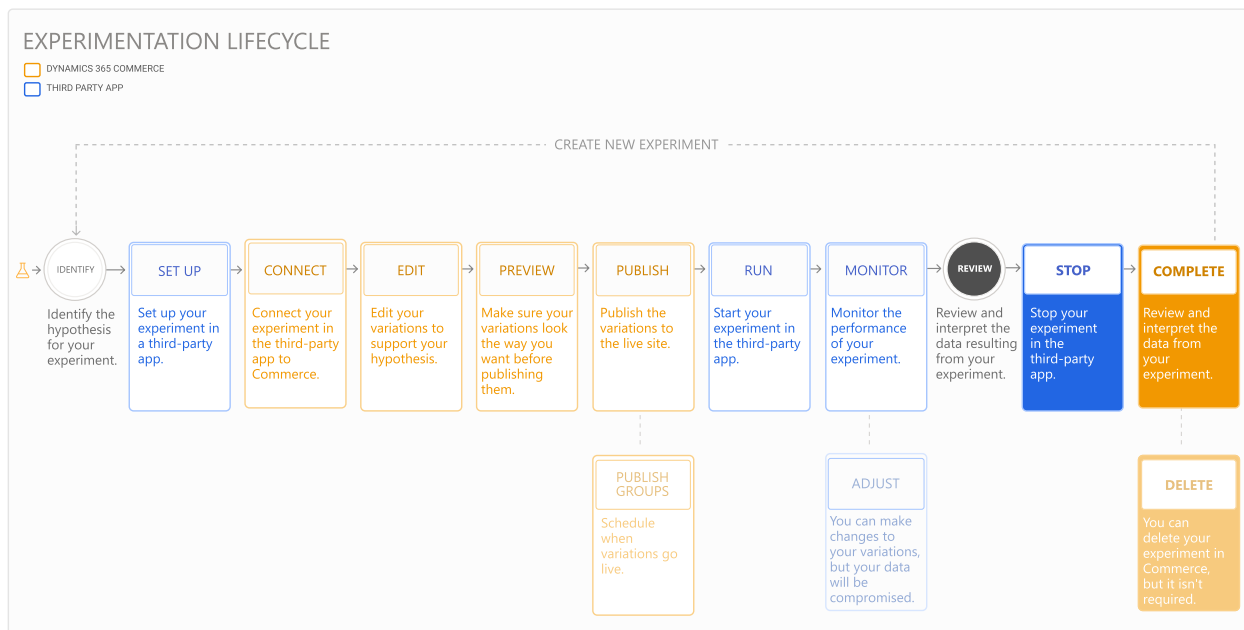
Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Promote a variation and complete an experiment

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to promote the variation that produced the best results in your experiment, and how to complete the experiment. The following diagram shows all of the steps involved in setting up and running an experiment on an e-Commerce website in Dynamics 365 Commerce. Additional steps are covered in separate topics.



After you've [run your experiment](#) and collected sufficient data to determine which variation you want to use on your live site, you'll promote the variation and end the experiment.

## Promote a variation

Use the data and analytics related to the experiment in the third-party service to decide which variation produced the best results. You can then promote it by replacing the current content on the live site with the winning variation so that it's available to all users of your website.

To promote the winning variation, follow these steps.

1. In Commerce site builder, select **Experiments** in the left navigation pane, and then select the experiment.
2. On the command bar, select **Complete experiment**.
3. In the **Complete the experiment** dialog box, select **Review the experiment data**. The third-party service opens where you can validate the metrics and determine which variation performed the best.
4. In the **Complete the experiment** dialog box, select the winning variation, and then select **Next**.
5. Open the third-party service and stop the experiment.
6. In site builder, select **Complete** to overwrite the original live page and publish the winning variation so that it's available to all users of your website.

### NOTE

If you choose to keep the current live page and not publish a variation, select **Republish the original page**.

# Delete your experiment

While it's not required that you delete a completed experiment in Commerce, you may choose to delete it to save space or clean up your workspace.

To delete an experiment in Commerce site builder, follow these steps.

1. Select **Experiments** in the left navigation pane, and then select the experiment.

## NOTE

If the experiment is still active, stop the experiment in the third-party service before proceeding.

2. On the command bar, select **Unpublish** to remove the variation content from the live site.
3. Select **Delete** to delete the experiment.

## Previous step

[Run and monitor an experiment](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Review the status of an experiment

2/18/2021 • 2 minutes to read • [Edit Online](#)

There are many steps involved in setting up and running an experiment in Dynamics 365 Commerce. For information about the experimentation lifecycle, see [Experimentation in Dynamics 365 Commerce](#).

To learn where an experiment is in the lifecycle, in Commerce site builder select **Experiments** in the left navigation pane. A list of experiments is displayed with the status of each experiment in both Commerce and the third-party service that is being used to enable the creation of experiments, assign variations, and analyze data.

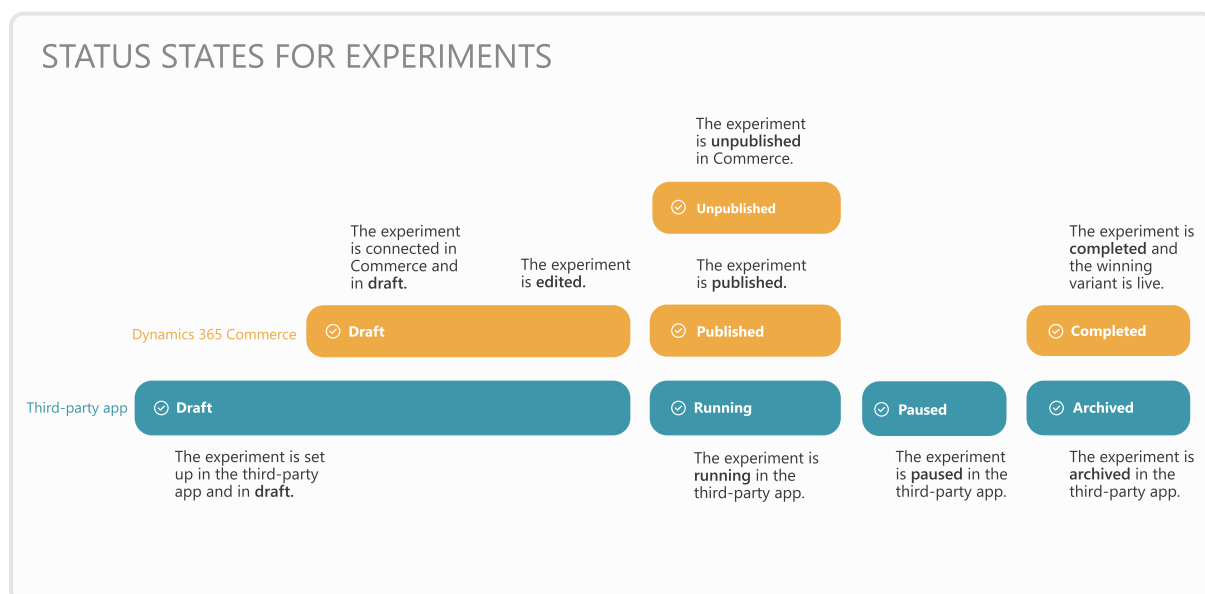
In the **Commerce status** column, the following values may be displayed.

- **Draft** - The experiment is connected to a page or fragment in Commerce and is being edited.
- **Published** - The experiment variations are ready to be displayed on your website. If the experiment is running in the third-party service, website users will see a variation of the page or fragment as selected by the third-party service.
- **Unpublished** - The experiment is no longer available on your website. Website users will only see the default version of the page or fragment even if the experiment is running in the third-party service.
- **Completed** - The experiment has run its course and a variation was promoted to live for all website users.

Similarly, in the **third-party status** column, the following values may be displayed to indicate what status the experiments are in the third-party service.

- **Draft** - The experiment is set up in the third-party service but hasn't been started.
- **Running** - The experiment was started in the third-party service and is collecting data.
- **Paused** - The experiment is paused and not collecting data. You must resume the experiment for it to collect data again.
- **Archived** - The experiment has run its course and has been cataloged in the third-party service for future reference.

The diagram below shows both sets of statuses and how they relate to each other.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Build custom response pages for 4xx/5xx status code errors

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to build custom response pages for 4xx and 5xx status code errors by using the authoring tools in Microsoft Dynamics 365 Commerce.

## Overview

If a request isn't successful, the server issues HTTP status code error responses. The 404 status code is captured and returned if a page isn't found, and the 500 status code is captured and returned if a server error occurs. In Dynamics 365 Commerce, application users can build custom status code error response pages that are shown to users for these status code error responses.

## Build a status code error response page

To start to build a status code error response page, follow these steps.

1. In your preferred web browser, sign in to Dynamics 365 Commerce.
2. Select the site that you want to build a 4xx/5xx status code error response page for.

### Build the template

To build the template for the status code error response page, follow these steps.

1. Go to **Templates**.
2. Select **New** to create a page template.
3. In the **New Template** dialog box, under **Template Name**, enter a name for the new template, and then select **OK**.
4. Build the template, based on the structure that you want the status code error response page to have.
5. Select **Save**, select **Finish editing** to check in the template, and then select **Publish** to publish it.

### Build the status code error response page

To build the status code error response page, follow these steps.

1. Go to **Pages**.
2. Select **New** to create a page.
3. In the **Choose a template** dialog box, select a template, and then, under **Page name**, enter a name for the status code error response page. Leave the **Page URL** field blank.
4. Build the page.
5. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

#### NOTE

You can create separate status code error response pages for 4xx and 5xx status code errors. Alternatively, you can use the same general status code error response page for both error categories.

### Set up a redirect for the status code error response page

To set up a redirect for the status code error response page, follow these steps.

1. Go to **URLs > New > New Alias**, and select the status code error response page that you built earlier.
2. In the **Alias** field, enter either **default-4xx** or **default-5xx**, depending on the status code error response page that you're setting up a redirect for. These aliases must be published. Otherwise, the redirect won't work.
3. Select **OK** to commit the linking.

#### **NOTE**

If you're using a single status code error response page for both error categories, repeat this procedure to link an alias for the other error category to the same page.

## Additional resources

[Work with templates](#)

[Add a new site page](#)

[Create a page URL](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Fraud Protection integration with Dynamics 365 Commerce

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic describes out-of-box integrations that are available between Microsoft Dynamics 365 Commerce and Dynamics 365 Fraud Protection.

## Key terms

TERM	DESCRIPTION
Purchase protection	The Fraud Protection module that analyzes purchases for fraud, based on risk levels that are determined by the merchant.
Storefront	The out-of-box e-commerce storefront that is provided with Commerce.
Azure Data Lake Storage Gen2	Data Lake Storage Gen2 is used to make Commerce data available for processing by the <b>Loss prevention</b> module.

## Overview

Fraud Protection is a service that offers fraud protection solutions to help retailers prevent fraudulent activity and identify places where fraud might be unnoticed. This topic describes out-of-box integrations between Fraud Protection and Commerce. It will be updated as new integrations between the two services are released in future releases. For more information about Fraud Protection, including information about modules that aren't yet supported by an out-of-box integration with Commerce, visit the [Fraud Protection landing page](#). You can also [request a callback](#) from a Dynamics 365 sales representative to discuss how Fraud Protection can help boost profitability, reduce operational expenses, and improve customer experiences.

Starting October 2020, a limited capacity of Fraud Protection is included in the Microsoft Dynamics 365 Commerce license. Commerce customers can now use Fraud Protection for no extra charge, up to the limits specified below:

- **Purchase protection**- Up to 2,000 assessments per month.
- **Account protection**- Up to 20,000 assessments per month.
- **Loss prevention**- Up to 8,000 transactions per month.

You have the option of purchasing additional Fraud Protection add-ons if your usage requires higher limits.

To start using Fraud Protection if you are an existing Commerce customer, visit the [Fraud Protection portal](#), sign in with tenant global administrator credentials, and complete a one-time setup for your environment.

## Purchase protection in Commerce

### Purchase protection overview

The first generally available offering from Fraud Protection is a purchase protection module that lets merchants sign in to the Fraud Protection dashboard for their organization and define fraud rules for online purchases. Based on the settings that a merchant configures in Fraud Protection, e-commerce transactions can be validated

with Fraud Protection before they are sent for payment authorization.

When an order is sent to the Fraud Protection purchase protection module, Fraud Protection analyzes the purchase and provides a risk assessment, based on merchant-defined fraud rules, insights that are driven by artificial intelligence (AI), and consortium-based fraud analytics. If the fraud score that is returned for the order exceeds the merchant's risk tolerance, Fraud Protection instructs the storefront to reject the order. If an order isn't rejected, Fraud Protection returns a fraud score that the storefront can use to determine the next steps in the order fulfillment process. Those steps might include putting the order on hold for manual review or follow-up with the customer who placed the order.

## Supported capabilities in the purchase protection integration

### Purchase events

Currently, Commerce is in public preview. However, when it becomes generally available, the purchase protection integration will support the receipt of Fraud Protection risk assessments and termination of orders in the online storefront.

Here is the flow for purchase events.

1. A storefront customer adds items to the basket and proceeds to checkout.
2. The customer enters shipping and payment details.
3. After the prerequisites are completed, the customer selects **Place order**.
4. Order details are sent to Fraud Protection for purchase protection assessment.
5. If the merchant rules that are defined in Fraud Protection determine that the order should be rejected, a response is sent to the storefront, and the order is terminated.

If Fraud Protection purchase protection causes an order to be terminated, the user receives the following error message: "The order cannot be processed at this time. Please try again later."



[Sign in](#) | [Account](#)

[Phones](#) [Audio](#) [Cameras](#) [Home appliances](#) [TV and video](#) [Computers](#)


The order cannot be processed at this time. Please try again later.

Delivery information

Billing information

Review and confirm

## Review and confirm

PRODUCT DETAILS	DELIVERY METHOD	EACH	QUANTITY	LINE TOTAL
 <a href="#">The Phone Company Mobile Phone</a> M300 Black 5301 <a href="#">Remove</a>	Ground	\$239.00	<input type="text" value="1"/>	\$239.00

### ORDER INFORMATION

LOYALTY CARD [Edit](#)

You have not added loyalty card number to your order

SHIPPING ADDRESS [Edit](#)

Karen Berg  
One Microsoft Way  
Redmond WA 98052 USA

PAYMENT METHOD [Edit](#)

CREDIT CARD  
\$254.54

BILLING ADDRESS

One Microsoft Way  
Redmond WA 98052 US

### ORDER SUMMARY

Subtotal: \$239.00

Shipping and handling: \$0.00

Tax: \$15.54

**Order total: \$254.54**

Total savings: \$0.00

Alternatively, if the merchant rules determine that the order should be approved, the response that is sent to the storefront includes the fraud score and the reason code that were determined by Fraud Protection. For the initial

integration, the Fraud Protection assessment isn't used in any way, and the response for both approval and rejection scenarios isn't saved.

Rejected orders aren't sent to payment processors for authorization, and they don't go through the order creation process in the back office.

#### **Bank events**

If an online order is approved based on the Fraud Protection assessment, the next step is to authorize payments for that order, if payment authorizations are applicable. Once the order is authorized with the payment processor, Fraud Protection is notified of the authorization result. By sending these results to Fraud Protection, the advanced AI can be trained to better predict future authorization results, thereby boosting the quality of future Fraud Protection assessments.

#### **Purchase status events**

Purchase status events resemble bank events. After an order is created in the Commerce back office, a signal is sent to Fraud Protection to indicate that the order was successfully created. Both bank events and purchase status events are informational events. Therefore, no response is expected from Fraud Protection.

#### **Setup**

The out-of-box purchase protection integration requires a Fraud Protection environment. To set up Fraud Protection, [request a callback](#) from a Dynamics 365 sales representative.

After the merchant's Fraud Protection environment is available, and purchase protection settings have been configured, the setup can continue in the Commerce back office.

#### **Key Vault setup**

The integration setup requires that a secret be used when Commerce communicates with Fraud Protection to get a purchase protection result. That secret must be stored by using an Azure Key Vault client. For information about how to set up a Key Vault client, see [Setting up Azure Key Vault Client](#).

The Fraud Protection certificate that is stored in Key Vault can be referenced only if it's referenced by Key Vault parameters in the Commerce back office. To set up Key Vault parameters, go to **Retail and Commerce > Headquarters setup > Parameters > Key Vault parameters** in Commerce.

Next, select the Key Vault URL that is used to store the Fraud Protection secret, and select **Add**. Then specify the name, description, and path of the Key Vault secret that is used to authenticate Commerce when it sends orders for purchase protection assessment.

#### **Commerce parameters setup**

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**.
2. On the **Dynamics Fraud Protection** tab, set the **Enable Dynamics Fraud Protection integration** option to **Yes**.
3. On the **Configuration** FastTab, add the Azure Active Directory (Azure AD) client ID, and then select the name of the Key Vault secret that you configured earlier.

By default, the **Assessment type** field is set to **Evaluate**. In this case, Fraud Protection will passively check orders for fraud but won't actively reject orders. Therefore, merchants can compare Fraud Protection risk assessments with their current fraud tools to understand the impact of Fraud Protection on acceptance rates.

Alternatively, the **Assessment type** field can be set to **Protect**. In this case, Fraud Protection will return "Reject" assessments, and fraudulent orders will be terminated before they are sent for authorization or created in the back office.

4. The **Dynamics Fraud Protection endpoint URL** field must be set. This URL is provided by Fraud Protection and will vary across user acceptance testing (UAT) and production environments.

**Commerce parameters**

- Channel deployment
- Configuration parameters
- Transaction validation
- Workforce management
- Electronic documents
- POS search criteria
- Clienteling
- Dynamics Fraud Protection**

**Set up Dynamics 365 Fraud Protection integration for Commerce**

**Enable Dynamics 365 Fraud Protection integration** ^

Yes

By enabling this feature, you consent to allow authorized users of the Dynamics 365 online service to activate, configure, and enable certain functionality which transmits certain of your data to an external system. Please consult the feature technical documentation for [more information](#) and review our [Privacy Statement](#).

**Configuration** ^

Azure Active Directory Client ID

KeyVault app secret name

Assessment type

Dynamics Fraud Protection endpoint ...

**NOTE**

The Key Vault and Fraud Protection settings are company-specific. To enable Fraud Protection for production environments, you don't enter the Azure AD client ID through the user interface (UI). Instead, you must create and submit a [service request](#). In the title of your request, clearly indicate that the request is to configure Fraud Protection purchase protection for a production Commerce or Retail environment.

## Loss prevention in Commerce

The **Loss prevention** capability in Fraud Protection will be made generally available in the third quarter (Q3) of 2020. The out-of-box integration for Loss prevention will be available in Commerce version 10.0.12.

### Loss prevention overview

Fraud that arises from abuse of return and discount policies is a top source of shrinkage for retailers. Existing physical deterrents are easy to work around. Therefore, to catch the most sophisticated forms of abuse, it's critical that retailers use artificial intelligence (AI) to identify loss.

The **Loss prevention** module in Fraud Prevention analyzes in-store returns and discounts to identify anomalies that might be caused by abuse of return and discount policies. By using AI, the **Loss prevention** module can identify patterns and anomalies that indicate potential fraud, and can uncover hard-to-detect sources of shrinkage.

The **Loss prevention** module analyzes Commerce data that is available through Data Lake Storage Gen2. This integration is an opt-in integration and isn't turned on by default.

After **Loss prevention** has finished analyzing the data, the results are surfaced in a [Fraud Protection](#) dashboard. From there, users can evaluate the results and observe trends that might indicate potential fraud.

### Turning on the Loss prevention integration with Commerce

#### Set up Fraud Protection

To set up Fraud Protection, [request a callback](#) from a Dynamics 365 sales representative. When the merchant's Fraud Protection environment is available, and Loss prevention settings have been configured, you can continue the setup by turning on Data Lake Storage Gen2 for the Commerce environment.

#### Turn on Data Lake Storage Gen2 for your Commerce environment

Before the data can be available in Data Lake Storage Gen2, the service must be turned on for the Commerce environment. For information about how to turn on Data Lake Storage Gen2 for your Commerce environment, see [Enable Azure Data Lake Storage in a Dynamics 365 Commerce environment](#).

#### Turn on Loss prevention



You can turn on the Loss prevention integration through the **Feature management** workspace in Commerce. The feature is named "Dynamics 365 Fraud Protection (DFP) Loss prevention." No other setup in Commerce is required to turn on the integration.

#### **Configure Loss prevention to connect to Data Lake Storage Gen2**

The final step is to return to the Fraud Protection environment and connect Loss prevention to the Data Lake Storage Gen2 pool that is associated with the Commerce account.

## Privacy notice

When you turn on the feature, some of your data is shared with other Microsoft online services. This data includes information about payments, credit, returns, and transaction status, or personal data. Fraud Protection purchase protection assessments aren't stored by the Retail or Commerce online services.

Your privacy is important to Microsoft. To learn more, read the [Microsoft Privacy Statement](#).

## Related articles

- [Payments FAQ](#)
- [Dynamics 365 payment data use](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up a B2B e-commerce site

2/18/2021 • 17 minutes to read • [Edit Online](#)

Business-to-business (B2B) e-commerce sites provide some key capabilities that optimize the workflow for a B2B user. This topic describes how to set up a B2B e-commerce site in Microsoft Dynamics 365 Commerce. It goes through the modules and site settings that must be configured to enable B2B-specific scenarios.

## Prerequisites

- To setup a B2B e-commerce site, you must enable and configure specific features in Commerce headquarters, as described in this topic.
- Core experiences, such as product discovery, product details pages, the cart, and checkout are powered by the same modules that are used for business-to-consumer (B2C) e-commerce sites. Site authors should be familiar with all the modules that Dynamics 365 Commerce supports. For more information, see [Module library overview](#).
- This topic assumes that site authors understand the basics of Commerce site builder, templates, fragments, and pages, so that they can enable the B2B features for e-commerce sites.

## Site-level settings

You can access site-level settings in site builder, at **Site Settings > Extensions**. The following two site-level settings apply to B2B scenarios:

- **Enable customer account payments** – This property enables users to pay for orders by using customer accounts. The available values are **Enabled for B2B customers**, **Enabled for B2C customers**, **Enabled for all customers**, and **Disabled for all customers**. If your B2B site supports customer accounts, you should select **Enabled for B2C customers**.
- **Enable order quantity limits** – This property lets you set limits on the number of items that can be ordered for each product or category. The available values are **Enabled for B2B customers**, **Enabled for B2C customers**, **Enabled for all customers**, and **Disabled for all customers**.

### NOTE

When you upgrade to the latest version of the module library, you must follow additional steps to ensure that the previously described site settings are available in your environment. For more information, see [Update the app.settings.json file](#).

## Create business partner sign-up pages

To become a business partner, users must first submit a business partner request. A link to the business partner request page will be available on the B2B home page, so that users can initiate the process. After users submit a business partner request, they will receive confirmation that the request has been submitted.

### Create a business partner request page

The **Partner sign up** module on a business partner request page is used to initiate user requests to become business partners. This module lets you collect the user information that is required for the sign-up process. Additionally, the **Business account address** module is used to capture the user's address.

To set up and configure the business partner request page in site builder, follow these steps.

1. Create a template that is named **Sign-up**. This template should include the following modules:
  - Partner sign up
  - Breadcrumb
  - Header
  - Footer
  - Content block
  - Text block
  - Product collection
2. Use the **Sign-up** template to create a page that is named **Business Partner Request**.
3. In the **Header** slot, add the header fragment that is preconfigured with the site header.
4. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
5. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
6. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the home page, and enter **Home** as the link text.
7. In the **Container** slot, add a **Partner sign up** module below the **Breadcrumb** module. In the module properties pane, under **Heading**, enter **Become a business partner**.
8. In the **Partner sign up** slot, add a **Business account address** module.
9. In the **Container** slot, add a **Text block** module below the **Partner sign up** module. Here, you can define some terms and conditions for the sign-up process.
10. Select **Save**, and then select **Preview** to preview the page.
11. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
12. Publish the URL for the page.

### **Create a business partner request confirmation page**

After a business partner request is submitted, a confirmation page should be shown to the user to acknowledge the submission.

To set up and configure the request confirmation page in site builder, follow these steps.

1. Use the **Sign-up** template that you created earlier to create a page that is named **Partner Request Confirmation**.
2. In the **Header** slot, add the header fragment that is preconfigured with the site header.
3. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
4. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
5. In the **Container** slot, add a **Content block** module. In the module properties pane, under **Heading**, enter **Request submitted**. In the **Rich Text** field, enter **Your request has been submitted**. Under **Links**, configure a link to the home page, and enter **Back to shopping** as the link text.
6. Add another **Container** module, and add a **Product Collection** module to it.
7. Configure the **Product Collection** module with the recommendation or category list that you want to showcase on the page.
8. Select **Save**, and then select **Preview** to preview the page.
9. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
10. Publish the URL for the page.

To add a link to the request confirmation page in site builder, follow these steps.

1. Go to the **Business Partner Request** page that you created earlier, and select **Edit**.
2. Select the **Partner sign up** module slot. In the properties pane, under **Link to the Sign-Up Confirmation page**, configure the link to the business partner request page that you created earlier.
3. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

### **Add a business partner request link to the home page**

After the business partner request sign-up and confirmation pages are created, you must make the sign-up page accessible through a link on the home page. You can complete this task by adding the link to any **Content block** module on the home page.

To add a business partner request link to the home page in site builder, follow these steps.

1. Go to the home page for your site, and select **Edit**.
2. Select a **Content block** module slot. In the module properties pane, under **Links**, configure a link to the business partner request page that you created earlier, and enter **Sign up to be a business partner** or similar text as the link text. Add an image as appropriate.
3. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Account management landing page

The account management landing page includes all the account management information that is required for both B2B and B2C e-commerce sites. Only signed-in users can view this page.

To create and configure a B2B account management landing page in site builder, follow these steps.

1. Create a template that is named **Account management**. This template should include the following modules:
  - Header
  - Footer
  - Breadcrumb
  - Account welcome tile
  - Account generic tile
  - Account Address tile
  - Account wishlist tile
  - Account address tile
  - Account loyalty tile
  - Account customer balance tile
  - Account order templates tile
  - Organization users
  - Business organization list
  - Customer account balance
  - Order template lines
  - Order template list
  - Account invoice tile
  - Invoices list
  - Invoice details
2. Use the **Account management** template to create a page that is named **My Account**.
3. In the **Header** slot, add the header fragment that is preconfigured with the site header.

4. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
5. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
6. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the home page, and enter **Home** as the link text.
7. In the **Container** slot, add a **Welcome tile** module. In the module properties pane, under **Heading**, enter **Welcome**.
8. In the **Main** slot, add another **Container** module (**Container 2**). In the module properties pane, set the **Width** value to **Fill Container**. Set the **Children Shown** value to **Two**.
9. In the **Container 2** slot, add an **Account generic tile** module. In the module properties pane, under **Heading**, enter **My Profile**. Under **Links**, configure a link to the **My profile** page.
10. In the **Container 2** slot, add another **Account generic tile** module. In the module properties pane, under **Heading**, enter **Order history**. Under **Links**, configure a link to the order history page.
11. In the **Main** slot, add another **Container** module (**Container 3**). In the module properties pane, set the **Width** value to **Fill Container**. Set the **Children Shown** value to **Two**.
12. In the **Container 3** slot, add an **Account address tile** module. In the module properties pane, under **Heading**, enter **My Address**. Under **Links**, configure a link to the **My address** page.
13. In the **Container 3** slot, add an **Account wishlist tile** module. In the module properties pane, under **Heading**, enter **My Wishlist**. Under **Links**, configure a link to the **My wishlist** page.
14. In the **Main** slot, add another **Container** module (**Container 4**). In the module properties pane, set the **Width** value to **Fill Container**. Set the **Children Shown** value to **Two**.
15. In the **Container 4** slot, add an **Organization users** module. In the module properties pane, under **Heading**, enter **Organization Users**.
16. In the **Container 4** slot, add an **Account customer balance tile** module. In the module properties pane, under **Heading**, enter **Account credit**.
17. In the **Main** slot, add another **Container** module (**Container 5**). In the module properties pane, set the **Width** value to **Fill Container**. Set the **Children Shown** value to **Two**.
18. In the **Container 5** module, add an **Account order templates tile** module. In the module properties pane, under **Heading**, enter **Order Templates**.
19. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

#### **NOTE**

Some of the sections that you added in step 13 through 18 won't appear on the "what you see is what you get" (WYSIWIG) canvas in site builder, because they require a signed-in B2B account.

## Create and configure customer balance pages and modules

Customer accounts can be used to pay for orders. The available balance in a customer account can be viewed from a user's account management page.

### **Create a customer balance page**

Before signed-in B2B users can view their customer balance, you must create a customer balance page.

To create a customer balance page in site builder, follow these steps.

1. Use the **Account management** template that you created earlier to create a page that is named **Customer Balance**.
2. In the **Header** slot, add the header fragment that is preconfigured with the site header.
3. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
4. In the **Main** slot, add another **Container** module (**Container 3**). In the module properties pane, set the **Width** value to **Fill Container**. Set the **Children Shown** value to **Two**.
5. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the account management landing page, and enter **My Account** as the link text.
6. In the **Container** slot, add a **Customer Account Balance** module. In the module properties pane, under **Heading**, enter **Account Balance**.
7. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
8. Publish the URL for the page.
9. Go to the account management landing page (**My Account**) that you created earlier.
10. In the properties pane for the **Account customer balance tile** module, add a link to the customer balance page.
11. Save and publish the page.

The customer balance page has now been created, and users can access it from the account management landing page.

### **Configure a checkout page so that the customer balance can be used as a form of payment**

The **Customer Account Payment** module is required to enable the customer balance to be used as a form of payment. This module should be added to the checkout page as a form of payment. For information about how to configure a checkout page and the modules that are required for checkout, including all payment details, see [Checkout module](#).

After you've configured a checkout page, you must add the **Customer Account Payment** module to the payment section, and then save and publish the page. B2B users will then be able sign in to the e-commerce site and apply their available customer balance to orders during checkout.

In addition, at **Site Builder > Extensions**, you must make sure that the **Enable customer account payments** property is set to **Enabled for B2B customers**.

## **Create order template pages**

Two order template pages can be set up for a B2B e-commerce site: an order templates list page and an order template lines page.

An order templates list page shows a list of all order templates that are available. It's set up by using the **Order templates list** module. The order templates list page lets you create or delete a template, and add the items in a template to the cart.

An order template lines page shows the details of the order template that is selected on an order templates list page. It's set up by using the **Order templates lines** module. When a user selects the name of a template on an order templates list page, the order template lines page appears and shows the details of the template. The user can then view and edit the items in the template.

### **Create an order templates list page**

To create an order templates list page in site builder, follow these steps.

1. Use the **Account management** template that you created earlier to create a page that is named **Order templates**.

2. In the **Header** slot, add the header fragment that is preconfigured with the site header.
3. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
4. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
5. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the account management landing page, and enter **My Account** as the link text.
6. In the **Container** slot, add an **Order templates list** module.
7. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
8. Publish the URL for the page.
9. Go to the account management landing page (**My Account**) that you created earlier.
10. In the properties pane for the **Account order templates tile** module, under **Links**, configure a link to the order templates list page that you just created.
11. Save and publish the page.

The order templates list page has now been created, and users can access it from the account management landing page.

### Create an order template lines page

To create an order template lines page in site builder, follow these steps.

1. Use the **Account management** template that you created earlier to create a page that is named **Order template lines**.
2. In the **Header** slot, add the header fragment that is preconfigured with the site header.
3. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
4. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
5. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the account management landing page, and enter **My Account** as the link text.
6. In the **Container** slot, add an **Order template lines** module.
7. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
8. Publish the URL for the page.

## Onboard business partner users

The organization users page lets the administrator of a business partner organization onboard additional users from that organization to the B2B e-commerce site. It's set up by using the **Business organization list** module. From the organization users page, an administrator can add or remove users, define account balances, and request statements for a user.

To create an organization users page in site builder, follow these steps.

1. Use the **Account management** template that you created earlier to create a page that is named **Organization Users**.
2. In the **Header** slot, add the header fragment that is preconfigured with the site header.
3. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
4. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
5. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the account management landing page, and enter **My Account** as the link text.
6. In the **Container** slot, add a **Business organization list** module. In the module properties pane, under **Heading**, enter **Organization Users**.

7. In the **Business organization list** module properties pane, enable the **Table Sort** and **Table pagination** properties. Set the pagination count to **5**.
8. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
9. Publish the URL for the page.
10. Go to the account management landing page (**My Account**) that you created earlier.
11. In the properties pane for the **Organization users tile** module, under **Links**, configure a link to the organization users page that you just created.
12. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

## Create invoice pages

An invoices list page shows a list of all available invoices. It's set up by using the **InvoicesList** module. From the invoice list page, users can pay off or request invoices.

An invoice details page shows the details of the invoice that is selected on an invoices list page. It's set up by using the **Invoice details** module. When a user selects an invoice ID on an invoice list page, the invoice details page appears and shows the details of the invoice.

### Create an invoices list page

To create an invoices list page in site builder, follow these steps.

1. Use the **Account management** template that you created earlier to create a page that is named **Invoices List**.
2. In the **Header** slot, add the header fragment that is preconfigured with the site header.
3. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
4. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
5. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the account management landing page, and enter **My Account** as the link text.
6. In the **Container** slot, add an **InvoicesList** module. In the module properties pane, under **Heading**, enter **Invoices**.
7. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
8. Publish the URL for the page.
9. Go to the account management landing page (**My Account**) that you created earlier.
10. In the properties pane for the **Account invoices tile** module, under **Links**, configure a link to the invoices list page that you just created.
11. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.

### Create an invoice details page

To create an invoice details page in site builder, follow these steps.

1. Use the **Account management** template that you created earlier to create a page that is named **Invoice Details**.
2. In the **Header** slot, add the header fragment that is preconfigured with the site header.
3. In the **Footer** slot, add the footer fragment that is preconfigured with the site footer.
4. In the **Main** slot, add a **Container** module. In the module properties pane, set the **Width** value to **Fill Container**.
5. In the **Container** slot, add a **Breadcrumb** module. In the module properties pane, under **Links**, configure a link to the account management landing page, and enter **My Account** as the link text. Then configure a link to the invoices list page, and enter **Invoice Lists** as the link text.
6. In the **Container** slot, add an **Invoice details** module.



7. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.
8. Publish the URL for the page.

## Additional resources

[Module library overview](#)

[Authoring page overview](#)

[Templates and layouts overview](#)

[Work with fragments](#)

[Add a new site page](#)

[Checkout module](#)

[Content block module](#)

[Product Collection](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create org modeling hierarchies for B2B organizations

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create organizational modeling hierarchies for business-to-business (B2B) organizations in Microsoft Dynamics 365 Commerce.

In Commerce headquarters, business partner organizations are represented by customer and customer hierarchy entities. The business partner organization and its users are represented as customers, and customer hierarchies are used to associate those customers with each other.

When a business partner request to join a B2B e-commerce site is approved, the following actions are performed:

- Two new customer records are created in the system: a **Type Organization** customer record for the business partner organization and a **Type Person** customer record for the requestor (that is, the business partner user who submitted the request).
- A new customer hierarchy record is created under **Customer > Customer hierarchy**. This record has the following header properties:
  - **Customer hierarchy ID** – A unique ID for the customer hierarchy. This ID uses the number sequence that is defined in Commerce shared parameters in Commerce headquarters.
  - **Name** – The organization name of the business partner, as specified in the onboarding request.
  - **Purpose** – This property is set to the predefined value **B2B organization**.
  - **Organization** – The customer ID of the business partner.

Here are the details of the customer hierarchy record:

- The customer record of the requestor is associated with the customer hierarchy.
- The administrator role is associated with the requestor.

When the administrator adds more users to the business partner organization on a B2B site, a new customer record for each user is created in Commerce headquarters. This customer record is also added to the relevant customer hierarchy record for the business partner and has the role of a "user."

## NOTE

In most cases, the corresponding property values of all customer records in a hierarchy should match. For example, because all business partner users should get similar prices for products, their price group and associated configurations should match. However, the system doesn't enforce this consistency. Therefore, the relevant Commerce headquarters users are responsible for ensuring that the property values and configurations match for all customers in a given hierarchy.

Commerce headquarters users can look at the property values for all customer records in the hierarchy in a side-by-side view. Select the relevant customer record properties by selecting the tab names on the drop-down menu. Users can directly view and edit the property values from this view. Alternatively, if you want to apply all the values from the administrator customer record to all the user customer records, select **Override** in the customer hierarchy details.

## Additional resources

[Set up a B2B e-commerce site](#)

[Manage business partner users on B2B e-commerce sites](#)

[Configure the customer account payment method for B2B e-commerce sites](#)

[Set product quantity limits for B2B e-commerce sites](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage business partner users on B2B e-commerce websites

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes how administrators can add, edit, and delete business partner users on business-to-business (B2B) e-commerce websites.

B2B e-commerce websites require that organizations register to become business partners. After an organization submits registration details to a B2B e-commerce website, it goes through a qualification process. If the organization is successfully qualified, it's onboarded as a business partner.

After an organization is onboarded as a business partner, the organization user who initiated the request to become a business partner is identified as the administrator user and is granted privileges to onboard additional authorized users of the B2B e-commerce website. These authorized users can then place orders on behalf of the business partner.

## Turn on the B2B e-commerce capabilities feature in Commerce headquarters

The B2B e-commerce capabilities feature in Commerce headquarters enables organizations to onboard business partners and define administrator users. This feature also enables administrators to create and manage business partner users and teams, and to assign specific roles to them. Finally, it enables business partner users to create order templates and use existing orders to reorder products.

To turn on the B2B e-commerce capabilities feature in Commerce headquarters, follow these steps.

1. Go to **Workspaces > Feature Management**.
2. On the **All** tab, filter on the **Module** field by using the term **Retail and commerce**.
3. Find and select the feature that is named **Enable the use of B2B eCommerce capabilities**, and then select **Enable now**.

## Create a number sequence and add it to Commerce shared parameters

Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require identifiers. For more information about number sequences, see [Number sequences overview](#).

To create a number sequence and add it to Commerce shared parameters in Commerce headquarters, follow these steps.

1. Go to **Retail and Commerce > Headquarters setup > Number sequences > Number sequences**, and create a number sequence.
2. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**, and add the new number sequence to the **Customer hierarchy ID** reference.

## Set up the administrator user for a new business partner

Potential business partners can initiate the onboarding process to a B2B e-commerce website by submitting an onboarding request via a link on the site. After a potential business partner user selects the link, they can

provide the details that are required for onboarding and sign-up. After the request is submitted, a submission confirmation page appears. If the submission is approved, the requestor (that is, the user who initiated the onboarding request) becomes the business partner administrator user.

To approve and set up a business partner administrator user in Commerce headquarters, follow these steps.

1. Go to **Retail and Commerce IT > Distribution schedule**.
2. Run the **P-0001** job to pull all business partner onboarding requests into Commerce headquarters.
3. After the **P-0001** job has successfully run, go to **Retail and Commerce IT > Customer**, and run the **Synchronize customers and business partners from async mode** job. After this job has successfully run, the onboarding requests are created as prospects records in Commerce headquarters. The **Type ID** field of these records is set to **B2B prospect**.
4. Go to **Customers > All prospects**, and open the prospects page.
5. Select the prospect record for the new business partner to open the prospect details page.
6. On the **General** tab, select **Convert > Approve/Reject** to approve or reject the onboarding request. When a confirmation message appears, confirm that you want to continue with the process, and approve the request. An email is then sent to the requestor's email address to confirm that their organization has been approved as a business partner.

After you approve the request, the **Status** field of the prospect record is set to **Approved**. Additionally, two new customer records are created in the system: a **Type Organization** customer record for the business partner organization and a **Type Person** customer record for the requestor. A customer hierarchy record for the business partner is also created.

7. Go to **Retail and Commerce IT > Distribution schedule**, and run the **1010 (Customers)** job to push the newly created customer and customer hierarchy records to the channel database.

After the request is approved, and the customer and customer hierarchy records are synced to the channel database, the requestor can sign in to the B2B e-commerce website by using the email address that they provided when they submitted the request. Users can use the sign-up flow to define the password for their account.

## Onboard additional business partner users

The business partner administrator user can onboard additional business partner users to the B2B e-commerce website as required.

To onboard additional business partner users to a B2B e-commerce website, follow these steps.

1. Sign in to the B2B e-commerce website as an administrator.
2. Go to **My Account > Organization users > View details**, and select **Add a user**.
3. Enter the required information, and then select **Save**. The status of the new user is set to **Pending**.

After the **P-0001** and **Synchronize customers and business partners from async mode** jobs are run, a **Type Person** customer record for the new user is created in Commerce headquarters. This customer record is also associated with the relevant business partner's customer hierarchy record. Additionally, an email is sent to the new user's email address to notify them that they have been added as a user of the business partner organization and can now sign in to the B2B e-commerce website.

4. Run the **1010 (Customers)** job to sync the new business partner user to the channel database.

After the customer record is synced, the status of the user on the B2B e-commerce website is set to **Active**, and the new user can sign in to the B2B e-commerce website by using their email address. Users can use the sign-up

flow to define the password for their account.

## Edit business partner user details

To edit the details of business partner users, follow these steps.

1. Sign in to the B2B e-commerce website as an administrator.
2. Go to **My Account > Organization users > View details**, select the **Edit** button (pencil symbol), make the required changes, and then select **Save**. The changes take effect only after the **P-0001, Synchronize customers and business partners from async mode**, and **1010 (Customers)** jobs have been run.

## Remove a business partner user

As required, an administrator can remove existing users of a business partner organization from the list of users who can access the B2B e-commerce website.

To remove a business partner user, follow these steps.

1. Sign in to the B2B e-commerce website as an administrator.
2. Go to **My Account > Organization users > View details**, and select the **Remove** button ("X" symbol). When a confirmation message appears, confirm that you want to remove the user. The change takes effect only after the **P-0001, Synchronize customers and business partners from async mode**, and **1010 (Customers)** jobs have been run.

### NOTE

When you remove a user from the list of users who can access the B2B e-commerce website, the corresponding customer record is removed from the business partner's customer hierarchy record. However, the customer record itself isn't deleted from Commerce headquarters.

## Onboard business partner and users in Commerce headquarters

Administrators can onboard business partners and users directly in Commerce headquarters.

To onboard business partners and users in Commerce headquarters, follow these steps.

1. Create a **Type Organization** customer record for the business partner organization.
2. Create **Type Person** customer records for business partner users. Make sure that a primary email address is specified for every customer.
3. For each **Type Person** customer record that must be designated as an administrator user of the business partner organization, on the **Retail** FastTab, set the **B2B administrator** option to **Yes**.
4. Create a customer hierarchy ID. In the **Name** field, enter a name.
5. In the **Organization** field, enter the business partner organization customer.
6. Select **Add**, and then select a customer in the **Name** field.
7. Repeat this process to add additional customers to the hierarchy.

## Additional information

- All the jobs that are mentioned in this topic can be configured to run on a schedule in a batch format. The expectation is that business partners will configure batch jobs as required.
- Currently, only one user/customer record can be designated as an administrator user, and that role can be changed only in Commerce headquarters. There is no support for self-service capabilities that let business partners to designate multiple administrators or change administrators from B2B e-commerce websites.

- Although spending limits can be defined for users, enforcement of spending limits during the order entry process hasn't yet been implemented.
- All business logic and validation for a user's experience on a B2B e-commerce website are based on the configuration of the customer record that is mapped to the user in Commerce headquarters.

## Additional resources

[Set up a B2B e-commerce site](#)

[Create org modeling hierarchies for B2B organizations](#)

[Configure the customer account payment method for B2B e-commerce sites](#)

[Set product quantity limits for B2B e-commerce sites](#)

[Number sequences overview](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure the customer account payment method for B2B e-commerce sites

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to configure the customer account payment method for business-to-business (B2B) e-commerce sites.

Retailers can accept various types of payment in exchange for the products and services that they sell in an e-commerce channel. Each payment type that a retailer accepts must be configured in Microsoft Dynamics 365 Commerce when the system is set up. The customer account (or "on account") payment method must be supported on B2B e-commerce sites.

## Prerequisites

1. Add the customer account payment method in Commerce headquarters.
2. Associate the customer account payment method with the e-commerce channel.
3. Make sure that **Allow on account** is enabled for the customer at **Retail and Commerce > Customers > All customers > Payment defaults** in Commerce headquarters. Also make sure that **Credit limit** parameters are set appropriately for the customer at **Retail and Commerce > Customers > All customers > Credit and Collections** in Commerce headquarters.

## Enable the customer account payment method in Commerce site builder

To enable the customer account payment method in Commerce site builder, follow these steps.

1. Go to **Site Settings > Extensions**.
2. Set the **Enable customer account payments** property to **Enabled for B2B customers**.
3. Select **Save and Publish**.

### NOTE

The new site settings take effect only after the app.settings.json file is updated. For more information, see [SDK and Module library updates](#).

## Enable the customer account payment method on the checkout page for the B2B e-commerce site

To enable the customer account payment method on the checkout page for the B2B e-commerce site, follow these steps.

1. In Commerce site builder, find and edit the checkout page or fragment that contains the checkout module for the B2B e-commerce site.
2. In the **Checkout section container** slot, select **Add Module**, and then add a **Customer account payment** module.
3. Position the **Customer account payment** module by selecting the ellipsis (...), and then selecting **Move Up** or **Move Down** as required.
4. Select **Save**, select **Finish editing** to check in the page, and then select **Publish** to publish it.



# Confirm that the customer account payment method has been enabled and published

To confirm that the customer account payment method has been enabled, follow these steps.

1. Sign in to the e-commerce site.
2. Add a product to the cart.
3. Go to the checkout page. You should see the new **Customer Account** payment method.

## Additional resources

[Set up a B2B e-commerce site](#)

[Create org modeling hierarchies for B2B organizations](#)

[Manage business partner users on B2B e-commerce sites](#)

[Set product quantity limits for B2B e-commerce sites](#)

[SDK and Module library updates](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set product quantity limits for B2B e-commerce sites

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to set product quantity limits for business-to-business (B2B) e-commerce sites.

Most products have a unit of measure that defines their grouping. The grouping affects how the products can be sold. Some products might have an additional grouping for quantities. This grouping determines whether the products can be sold as individual units or multiples, and whether there is a minimum or maximum order quantity limit that must be followed.

The quantity limiting feature ensures that the minimum, maximum, multiple, and standard quantities that are configured in Microsoft Dynamics 365 Commerce (in the default order settings or the Commerce site builder site settings) are applied to customer orders that are placed on an e-commerce site. Product quantity limits aren't currently supported for the point of sale (POS) and call centers.

Many retailers provide the option of customer orders (also known as special orders) to meet various product and fulfillment requirements. Here are some typical scenarios:

- A customer wants products of specific variants to be sold in multiples of a few.
- A customer wants to pick up products from a store or location that differs from the store or location where the customer purchased those products. However, the packing standards for the stores differ from the packing standards for online sales distribution.
- A customer wants to buy a limited-edition product that has a maximum quantity limit for items that can be purchased.

## Turn on the default order settings feature in Commerce headquarters

Before you can set product quantity limits, the default order settings feature must be turned on in Commerce headquarters.

To turn on the default order settings feature, follow these steps.

1. Go to **System administration > Workspaces > Feature management**.
2. Find and select the **Support the Site and Default order settings in the customer order** feature.
3. At the bottom of the right pane, select **Enable now**.

## Define quantity settings

You can define the quantity settings on the **Default order settings** page.

To define the quantity settings, follow these steps.

1. Go to **Product Retail and Commerce > Products and categories > Released products by category**.
2. Select a released product.
3. On the Action Pane, on the **Manage inventory** tab, in the **Order settings** group, select **Default order settings**.
4. On the **Default order settings** page, on the **Sales order** FastTab, in the **Sales quantity** section, set the

product sales quantities:

- **Multiple** – The quantity that the product can be bought in multiples of.
- **Minimum Order Quantity** – The minimum number of products that must be purchased.
- **Maximum Order Quantity** – The maximum number of products that can be purchased.
- **Standard Order Quantity** – The default quantity that is automatically entered when the product is selected.

## Turn on the B2B order quantity limits feature in Commerce site builder

To turn on the B2B order quantity limits feature in Commerce site builder, follow these steps.

1. Go to **Site settings > Extensions**
2. Under **Enable Order Quantity Limits**, select **Enabled for B2B customers** in the drop-down menu.

### NOTE

Updated site builder settings take effect only after the app.settings.json file has been updated. For more information, see [SDK and Module library updates](#).

## Additional resources

[Set up a B2B e-commerce site](#)

[Create org modeling hierarchies for B2B organizations](#)

[Manage business partner users on B2B e-commerce sites](#)

[Configure the customer account payment method for B2B e-commerce sites](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Monitor sales and margin performance

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can monitor sales and margin performance in real time using Dynamics 365 Commerce.

As part of Commerce, users can monitor sales and margin performance in real time across different levels of the organization hierarchy for the following dimensions:

- Products
- Categories
- Discounts
- Years as time period
- Registers/terminals
- Staff/employees
- Customers
- Operating units

Additionally, two unique reports that take advantage of hierarchical grid structuring let users monitor sales and margin performance by drilling down from the top category node to individual leaf nodes of the category in the default product category hierarchy. Users can also drill-down from the top operating unit to an individual channel in the organization hierarchy that is defined as the default organization hierarchy for reporting. You can open the reports from any of the following locations:

- **Store management** workspace > **Retail and Commerce** > **Channels** > **Store management** > **Reports**
- **Category and product management** workspace > **Retail and Commerce** > **Product and categories** > **Store management** > **Reports**
- **Pricing and discount management** workspace > **Retail and Commerce** > **Pricing and discounts** > **Store management** > **Reports**
- **Inquiries and reports** section > **Retail and Commerce** > **Inquiries and reports** > **Sales reports**

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Analyze sales trends and patterns

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can study sales trends and patterns in real time in Dynamics 365 Commerce.

As part of Commerce, users can study sales trends and patterns in real time across different levels of the organization hierarchy over a period of years by using the out-of-box **Channel sales by year** report. You can open this report from any of the following locations:

- **Store management** workspace > **Retail and Commerce** > **Channels** > **Store management** > **Reports** > **Channel sales by year report**
- **Store financials** workspace > **Retail and Commerce** > **Channels** > **Store financials** > **Reports** > **Channel sales by year report**
- **Inquiries and reports** section > **Retail and Commerce** > **Inquiries and reports** > **Sales reports** > **Channel sales by year report**

Users can also study sales trends and patterns by hour across different levels of the organization hierarchy over a selected period by using the out-of-box **Channel sales by hour** report. You can open this report from any of the following locations:

- **Store management** workspace > **Retail and Commerce** > **Channels** > **Store management** > **Reports** > **Channel sales by hour report**
- **Store financials** workspace > **Retail and Commerce** > **Channels** > **Store financials** > **Reports** > **Channel sales by hour report**
- **Inquiries and reports** section > **Retail and Commerce** > **Inquiries and reports** > **Sales reports** > **Channel sales by hour report**

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Assess sales performance by product

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can study profitability for the top 10 products using Dynamics 365 Commerce.

As part of Commerce, users can also study profitability for the top products (10 to 100) across different levels of the organization hierarchy, based on one of the following criteria:

- Sales amount
- Quantity
- Gross profit margin
- Margin percentage

For this assessment, you can use the out-of-box **Top products** report, which you can open from any of the following locations:

- **Store management workspace > Retail and Commerce > Channels > Store management > Reports > Top products report**
- **Category and product management workspace > Retail and Commerce > Channels > Store management > Reports > Top products report**
- **Inquiries and reports section > Retail and Commerce > Inquiries and reports > Sales reports > Top products report**

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Assess customer and product profitability

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article explains how you can use the in-memory and real-time analytics to access, explore, and gain insight about customers and product profitability from your Dynamics 365 Commerce data.

As part of Commerce, users can study profitability for the top customers (10 to 100) across different levels of the organization hierarchy, based on one of the following criteria:

- Sales amount
- Quantity
- Gross profit margin
- Margin percentage

For this assessment, you can use the out-of-box **Top customers** report, which you can open from any of the following locations:

- **Store management** workspace > **Retail and Commerce** > **Channels** > **Store management** > **Reports** > **Top customers report**
- **Inquiries and reports** section > **Retail and Commerce** > **Inquiries and reports** > **Sales reports** > **Top customers report**

Likewise, users can study profitability for the top products (10 to 100) across different levels of the organization hierarchy, based on one of the following criteria:

- Sales amount
- Quantity
- Gross profit margin
- Margin percentage

For this assessment, you can use the out-of-box **Top products** report, which you can open from any of the following locations:

- **Store management** workspace > **Retail and Commerce** > **Channels** > **Store management** > **Reports** > **Top products report**
- **Category and product management** workspace > **Retail and Commerce** > **Products and categories** > **Store management** > **Reports** > **Top products report**
- **Inquiries and reports** section > **Retail and Commerce** > **Inquiries and reports** > **Sales reports** > **Top products report**

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Analyze store performance

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article explains how you can use the in-memory and real-time analytics to access, explore, and gain insight about store performance, based on your Dynamics 365 Commerce data.

As part of Retail, users can study store performance in real time across different levels of the organization hierarchy over a selected period by opening the out-of-box **Channel summary** report from any of the following locations:

- **Retail store management workspace > Retail > Channels > Retail store management > Reports > Channel summary report**
- **Retail store financials workspace > Retail > Channels > Retail store financials > Reports > Channel summary report**
- **Inquiries and reports section > Retail > Inquiries and reports > Sales reports > Channel summary report**

This report provides a snapshot of following summaries as part of store performance:

- Gross sales summary
- Tender type summary
- Tax summary
- Price overrides summary
- Discounts summary

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Set up Recency, Frequency, and Monetary (RFM) analysis

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how to set up a Recency, Frequency, and Monetary (RFM) analysis of your customers.

Recency, frequency, and monetary (RFM) analysis is a marketing tool that your organization can use to evaluate the data that is generated by customer purchases. After you set up RFM analysis, customers are assigned a calculated RFM score as they make purchases. The RFM score can be a three-digit rating or an aggregate number, depending on how your organization has configured RFM analysis. Here's how the rating works if your organization uses a three-digit rating for the score:

- The first digit is the customer's recency rating, which is how recently the customer made a purchase from your organization.
- The second digit is the customer's frequency rating, which is how often the customer makes purchases from your organization.
- The third digit is the customer's monetary rating, which is how much the customer spends when he makes purchases from your organization.

For example, your organization has set the ratings on a scale of 1 through 5, where 5 is the highest rating. In this case, a customer rating of 535 tells you the following information about the customer:

- **Recency rating of 5** – The customer recently made a purchase.
- **Frequency rating of 3** – The customer purchases products from your organization moderately often.
- **Monetary rating of 5** – When the customer makes a purchase, he spends a significant amount of money.

If your organization uses an aggregate number for the score, the individual ratings are added together. For the same example, the customer has a rating of 13 (5 + 3 + 5).

## Set up RFM analysis for the customers in your organization

1. Go to **Call center > Periodic > RFM analysis**.
2. On **RFM analysis** page, select **New**. In the **RFM definition** field, enter a name for the RFM definition. For example, you could call the definition RFM-A.
3. Enter a start date and end date for this RFM definition.
4. On the **General** FastTab, do the following:
  - If each section of the RFM score must contain an equal count of customers, select the **Even distribution** check box.
  - Select the **Add scores** check box to aggregate the three scores. For example, this would give a customer an RFM score of 13 instead of 535.
  - Select the **Save history** check box to require the system to save the statistical data for customers so that the data can be used to calculate the RFM score.
5. On the **Recency** FastTab, do the following:
  - In the **Divisions** field, enter the number of divisions, or groups, which will be used to calculate the recency score for customers. For example, if you have 100 customers, a division of 5 means that there are 20 customers for each score. The 20 customers who have made purchases most recently have a

recency score of 5. The next 20 customers have a recency score of 4, and so on. If you have 50 customers, 10 customers have a recency score of 5, 10 have a recency score of 4, and so on.

- In the **Priority** field, select how much weight to give the recency parameter in relation to the other parameters when the RFM score is calculated for a customer. For example, you might place more value on the recency score than the monetary score.
- In the **Multiplier** field, enter the value by which to multiply the recency score. If you do not enter a value, the score will not be multiplied.
- In the **Period** field, select the time period by which the recency score is calculated. For example, by week or by month.

6. On the **Frequency** FastTab, do the following:

- In the **Divisions** field, enter the number of divisions, or groups, which will be used to calculate the frequency score for customers.
- In the **Priority** field, select how much weight to give the frequency parameter in relation to the others when the RFM score is calculated for a customer.
- In the **Multiplier** field, enter the value by which to multiply the frequency score. If you do not enter a value, the score will not be multiplied.

7. On the **Monetary** FastTab, do the following:

- In the **Divisions** field, enter the number of divisions, or groups, which will be used to calculate the monetary score for customers.
- In the **Priority** field, select how much weight to give the monetary parameter in relation to the others when the RFM score is calculated for a customer.
- In the **Multiplier** field, enter the value by which to multiply the monetary score. If you do not enter a value, the score will not be multiplied.
- In the **Gross/net** field, select whether the customer's monetary score should be calculated by using the gross or net invoice amount.
- If a customer's return amounts should be subtracted from the customer's total invoice calculation, select the **Subtract returns** check box.

## View a customer's RFM score

Use this procedure to view a customer's RFM score.

1. Go to **Call center > Journals > Customer service**.
2. On the **Customer service** page, in the **Customer service** pane, in the search fields, select the keyword type to search on and enter the search text.
3. Select **Search**.
4. On the **Customer search** page, select the customer record that you want, and then click **Select customer**.

The RFM score is displayed in the **Order history** group on the right side of the **Customer service** page.

## View or clear the history of an RFM analysis record

Use this procedure to view or clear the history of an RFM analysis record.

1. Go to **Call center > Periodic > RFM analysis**.
2. On the **RFM analysis** page, select the record that you want to view.
3. To view the record history, select the **History** FastTab.
4. To clear the history of the record, select **Clear history**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Overview of fiscal integration for Commerce channels

2/18/2021 • 10 minutes to read • [Edit Online](#)

## Introduction

This topic is an overview of the fiscal integration capabilities that are available in Dynamics 365 Commerce. Fiscal integration includes integration with various fiscal devices and services that enable fiscal registration of sales in accordance with local fiscal laws that are aimed at preventing tax fraud in the retail industry. Here are some typical scenarios that can be covered by using fiscal integration:

- Register a sale on a fiscal device that is connected to point of sale (POS), such as a fiscal printer, and print a fiscal receipt for the customer.
- Securely submit information that is related to sales and returns that are completed in Retail POS to an external web-service that is operated by the tax authority.
- Help guarantee inalterability of sales transaction data through digital signatures.

The fiscal integration functionality is a framework that provides a common solution for further development and customization of the integration between Retail POS and fiscal devices and services. The functionality also includes fiscal integration samples that support basic scenarios for specific countries or regions, and that work with specific fiscal devices or services. A fiscal integration sample consists of several extensions of Commerce components and is included in the software development kit (SDK). For more information about the fiscal integration samples, see [Fiscal integration samples in the Retail SDK](#). For information about how to install and use the Retail SDK, see [Retail software development kit \(SDK\) architecture](#).

To support other scenarios that aren't supported by a fiscal integration sample, to integrate Retail POS with other fiscal devices or services, or to cover requirements of other countries or regions, you must either extend an existing fiscal integration sample or create a new sample by using an existing sample as an example.

## Fiscal registration process and fiscal integration samples for fiscal devices

A fiscal registration process in Retail POS can consist of one or more steps. Each step involves fiscal registration of specific transactions or events in one fiscal device or service. The following solution components participate in the fiscal registration in a fiscal device that is connected to a Hardware station:

- **Commerce runtime (CRT) extension** – This component serializes transaction/event data in the format that is also used for interaction with the fiscal device, parses responses from the fiscal device, and stores the responses in the channel database. The extension also defines the specific transactions and events that must be registered. This component is often referred to as a *fiscal document provider*.
- **Hardware station extension** – This component initializes the communication with the fiscal device, sends requests and direct commands to the fiscal device based on the transaction/event data that is extracted from the fiscal document, and receives responses from the fiscal device. This component is often referred to as a *fiscal connector*.

A fiscal integration sample for a fiscal device contains the CRT and Hardware station extensions for a fiscal document provider and a fiscal connector, respectively. It also contains the following component configurations:

- **Fiscal document provider configuration** – This configuration defines an output method and a format for fiscal documents. It also contains a data mapping for taxes and payment methods, to make data from Retail

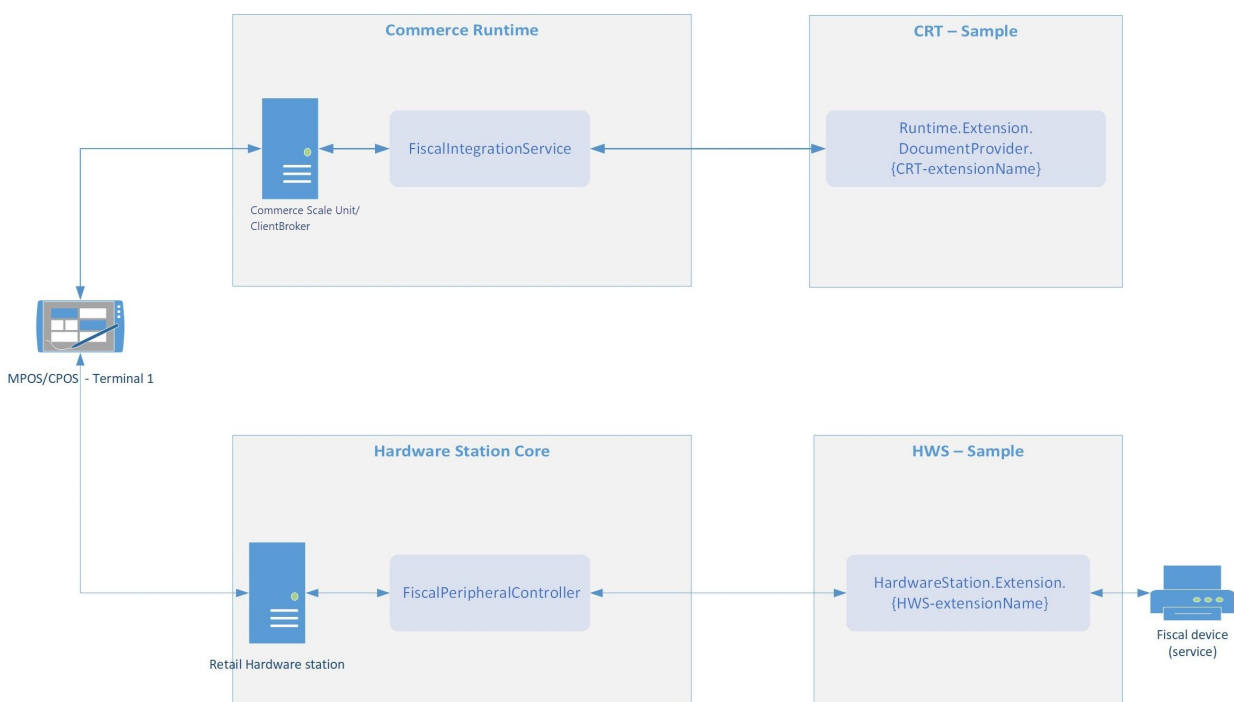
POS compatible with the values that are predefined in the fiscal device firmware.

- **Fiscal connector configuration** – This configuration defines the physical communication with the specific fiscal device.

A fiscal registration process for a specific POS register is defined by a corresponding setting in the POS functionality profile. For more details about how to configure a fiscal registration process, upload fiscal document provider and fiscal connector configurations, and change their parameters, see [Set up a fiscal registration process](#).

The following example shows a typical fiscal registration execution flow for a fiscal device. The flow starts with an event in the POS (for example, finalization of a sales transaction) and implements the following sequence of steps:

1. The POS requests a fiscal document from CRT.
2. CRT determines whether the current event requires fiscal registration.
3. Based on the fiscal registration process settings, CRT identifies a fiscal connector and corresponding fiscal document provider to use for the fiscal registration.
4. CRT runs the fiscal document provider that generates a fiscal document (for example, an XML document) that represents the transaction or event.
5. The POS sends the fiscal document that CRT prepares to a Hardware station.
6. The Hardware station runs the fiscal connector that processes the fiscal document and communicates it to the fiscal device or service.
7. The POS analyzes the response from the fiscal device or service to determine whether the fiscal registration was successful.
8. CRT saves the response to the channel database.



## Error handling

The fiscal integration framework provides the following options to handle failures during fiscal registration:

- **Retry** – Operators can use this option when the failure can be resolved quickly, and the fiscal registration can be rerun. For example, this option can be used when the fiscal device isn't connected, the fiscal printer is out of paper, or there is a paper jam in the fiscal printer.
- **Cancel** – This option lets operators postpone the fiscal registration of the current transaction or event if it fails. After the registration is postponed, the operator can continue to work on the POS and can complete any

operation that the fiscal registration isn't required for. When any event that requires the fiscal registration occurs in the POS (for example, a new transaction is opened), the error handling dialog box automatically appears to notify the operator that the previous transaction wasn't correctly registered and to provide the error handling options.

- **Skip** – Operators can use this option when the fiscal registration can be omitted under specific conditions and regular operations can be continued on the POS. For example, this option can be used when a sales transaction that the fiscal registration failed for can be registered in a special paper journal.
- **Mark as registered** – Operators can use this option when the transaction was actually registered in the fiscal device (for example, a fiscal receipt was printed), but a failure occurred when the fiscal response was being saved to the channel database.

#### **NOTE**

The **Skip** and **Mark as registered** options must be activated in the fiscal registration process before they are used. In addition, corresponding permissions must be granted to operators.

The **Skip** and **Mark as registered** options enable info codes to capture some specific information about the failure, such as the reason for the failure or a justification for skipping the fiscal registration or marking the transaction as registered. For more details about how to set up error handling parameters, see [Set error handling settings](#).

#### **Optional fiscal registration**

Fiscal registration might be mandatory for some operations but optional for others. For example, the fiscal registration of regular sales and returns might be mandatory, but the fiscal registration of operations that are related to customer deposits might be optional. In this case, failure to complete the fiscal registration of a sale should block further sales, but failure to complete the fiscal registration of a customer deposit should not block further sales. To distinguish mandatory and optional operations, we recommend that you handle them through different document providers, and that you set up separate steps in the fiscal registration process for those providers. The **Continue on error** parameter should be enabled for any step that is related to optional fiscal registration. For more details about how to set up error handling parameters, see [Set error handling settings](#).

#### **Manually running fiscal registration**

If the fiscal registration of a transaction or event has been postponed after a failure (for example, if the operator selected **Cancel** in the error handling dialog box), you can manually rerun the fiscal registration by invoking a corresponding operation. For more details, see [Enable manual execution of postponed fiscal registration](#).

#### **Fiscal registration health check**

The health check procedure for fiscal registrations verifies the availability of the fiscal device or service when specific events occur. If the fiscal registration can't currently be completed, the operator is notified in advance.

The POS runs the health check when the following events occur:

- A new transaction is opened.
- A suspended transaction is recalled.
- A sales or return transaction is finalized.

If the health check fails, the POS shows the health check dialog box. This dialog box provides the following buttons:

- **OK** – This button lets the operator ignore a health check error and continue to process the operation. Operators can select this button only if the **Allow skip health check error** permission is enabled for them.
- **Cancel** – If the operator selects this button, the POS cancels the last action (for example, an item isn't added to a new transaction).

#### NOTE

The health check is run only if the current operation requires fiscal registration, and if the **Continue on error** parameter is disabled for the current step of the fiscal registration process. For more details, see [Set error handling settings](#).

## Storing fiscal response in fiscal transaction

When fiscal registration of a transaction or event is successful, a fiscal transaction is created in the channel database and linked to the original transaction or event. Similarly, if the **Skip** or **Mark as registered** option is selected for a failed fiscal registration, this information is stored in a fiscal transaction. A fiscal transaction holds the fiscal response of the fiscal device or service. If the fiscal registration process consists of several steps, a fiscal transaction is created for each step of the process that resulted in a successful or failed registration.

Fiscal transactions are transferred to Headquarters by the *P-job*, together with transactions. On the **Fiscal transactions** FastTab of the **Store transactions** page, you can view the fiscal transactions that are linked to transactions.

A fiscal transaction stores the following details:

- Fiscal registration process details (process, connector group, connector, and so on). It also stores the serial number of the fiscal device in the **Register number** field, if this information is included in the fiscal response.
- The status of the fiscal registration: **Completed** for successful registration, **Skipped** if the operator selected the **Skip** option for a failed registration, or **Marked as registered** if the operator selected the **Mark as registered** option.
- Info code transactions that are related to a selected fiscal transaction. To view the info code transactions, on the **Fiscal transactions** FastTab, select a fiscal transaction that has a status of **Skipped** or **Marked as registered**, and then select **Info code transactions**.

## Fiscal texts for discounts

Some countries or regions have special requirements about additional texts that must be printed on fiscal receipts when different kinds of discounts are applied. The fiscal integration functionality lets you set up a special text for a discount that is printed after a discount line on a fiscal receipt. For manual discounts, you can configure a fiscal text for the info code that is specified as the **Product discount** info code in the POS functionality profile. For more details about how to set up fiscal texts for discounts, see [Set up fiscal texts for discounts](#).

## Printing fiscal X and fiscal Z reports

Fiscal integration functionality supports generation of end-of-day statements that are specific to the integrated fiscal device or service:

- New buttons that run corresponding operations should be added to the POS screen layout. For more details, see [Set up fiscal X/Z reports from the POS](#).
- In the fiscal integration sample, these operations should be matched to the corresponding operations of the fiscal device.

## Fiscal integration samples in the Retail SDK

The following fiscal integration samples are currently available in the Retail SDK:

- [Fiscal printer integration sample for Italy](#)
- [Fiscal printer integration sample for Poland](#)

- [Fiscal registration service integration sample for Austria](#)
- [Fiscal registration service integration sample for Czech Republic](#)
- [Control unit integration sample for Sweden](#)
- [Fiscal registration service integration sample for Germany](#)

The following fiscal integration functionality is also available in the Retail SDK but doesn't currently take advantage of the fiscal integration framework. Migration of this functionality to the fiscal integration framework is planned for later updates.

- [Digital signature for France](#)
- [Digital signature for Norway](#)

The following legacy fiscal integration functionality that is available in Retail SDK does not use the fiscal integration framework and will be deprecated in later updates:

- [Control unit integration sample for Sweden \(legacy\)](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Set up the fiscal integration for Commerce channels

2/18/2021 • 12 minutes to read • [Edit Online](#)

## Introduction

This topic provides guidelines for setting up the fiscal integration functionality for Commerce channels. For more information about the fiscal integration, see [Overview of fiscal integration for Commerce channels](#).

The process of setting up the fiscal integration includes the following tasks:

1. Configure fiscal connectors that represent fiscal devices or services that are used for fiscal registration purposes, such as fiscal printers.
2. Configure document providers that generate fiscal documents that will be registered in fiscal devices or services by fiscal connectors.
3. Configure the fiscal registration process that defines a sequence of fiscal registration steps and the fiscal connectors and fiscal document providers that are used for each step.
4. Assign the fiscal registration process to point of sale (POS) functionality profiles.
5. Assign connector technical profiles to hardware profiles.

## Set up a fiscal registration process

Before you use the fiscal integration functionality, you should configure the following settings.

1. Update Commerce parameters.
  - a. On the **Commerce shared parameters** page, on the **General** tab, set the **Enable fiscal integration** option to **Yes**. On the **Number sequences** tab, define the number sequences for the following references:
    - Fiscal technical profile number
    - Fiscal connector group number
    - Registration process number
  - b. On the **Commerce parameters** page, define the number sequence for the fiscal functional profile number.

### NOTE

Number sequences are optional. Numbers for all fiscal integration entities can be generated either from number sequences or manually.

2. Upload configurations of fiscal connectors and fiscal document providers.

A fiscal document provider is responsible for generating fiscal documents that represent transactions and events that are registered on the POS in a format that is also used for the interaction with a fiscal device or service. For example, a fiscal document provider might generate a representation of a fiscal receipt in an XML format.

A fiscal connector is responsible for the communication with a fiscal device or service. For example, a fiscal connector might send a fiscal receipt that a fiscal document provider created in an XML format to a fiscal printer. For more details about fiscal integration components, see [Fiscal registration process and fiscal integration samples for fiscal devices](#).

- a. On the **Fiscal connectors** page (**Retail and Commerce > Channel setup > Fiscal integration > Fiscal connectors**), upload an XML configuration for each device or service that you plan to use for fiscal integration purposes.

**TIP**

By selecting **View**, you can view all functional and technical profiles that are related to the current fiscal connector.

- b. On the **Fiscal document providers** page (**Retail and Commerce > Channel setup > Fiscal integration > Fiscal document providers**), upload an XML configuration for each device or service that you plan to use.

**TIP**

By selecting **View**, you can view all functional profiles that are related to the current fiscal document provider.

For examples of configurations of fiscal connectors and fiscal document providers, see [Fiscal integration samples in the Retail SDK](#).

**NOTE**

Data mapping is considered part of a fiscal document provider. To set up different data mappings for the same connector (for example, state-specific regulations), you should create different fiscal document providers.

3. Create connector functional profiles and connector technical profiles.

- a. On the **Connector functional profiles** page (**Retail and Commerce > Channel setup > Fiscal integration > Connector functional profiles**), create a connector functional profile for each combination of a fiscal connector and a fiscal document provider that is related to this fiscal connector.

- a. Select a connector name.
- b. Select a document provider.

You can change the data mapping parameters in a connector functional profile. To restore the default parameters that are defined in the fiscal document provider configuration, select **Update**.

**Examples**

	FORMAT	EXAMPLE
VAT rates settings	value : VATrate	1 : 2000, 2 : 1800
VAT codes mapping	VATcode : value	vat20 : 1, vat18 : 2
Tender types mapping	TenderType : value	Cash : 1, Card : 2

**NOTE**

Connector functional profiles are company-specific. If you plan to use the same combination of a fiscal connector and a fiscal document provider in different companies, you should create a connector functional profile for each company.

- b. On the **Connector technical profiles** page (**Retail and Commerce > Channel setup > Fiscal integration > Connector technical profiles**), create a connector technical profile for each fiscal connector.
  - a. Select a connector name.
  - b. Select a connector type. For devices that are connected to a Hardware station, select **Local**.

**NOTE**

Only local connectors are currently supported.

Parameters on the **Device** and **Settings** tabs in a connector technical profile can be changed. To restore the default parameters that are defined in the fiscal connector configuration, select **Update**. While a new version of an XML configuration is loaded, you receive a message that states that the current fiscal connector or fiscal document provider is already being used. This procedure doesn't override manual changes that were previously made in connector functional profiles and connector technical profiles. To apply the default set of parameters from a new configuration, on the **Connector functional profiles** page or the **Connector technical profiles** page, select **Update**.

4. Create fiscal connector groups.

A fiscal connector group combines functional profiles of fiscal connectors that perform identical functions and are used at the same step of a fiscal registration process. For example, if several fiscal printer models can be used in a store, fiscal connectors for those fiscal printers can be combined in a fiscal connector group.

- a. On the **Fiscal connector group** page (**Retail and Commerce > Channel setup > Fiscal integration > Fiscal connector groups**), create a new fiscal connector group.
  - b. Add functional profiles to the connector group. On the **Functional profiles** tab, select **Add**, and select a profile number. Each fiscal connector in a connector group can only have one functional profile.
  - c. To suspend use of the functional profile, set the **Disable** option to **Yes**. This change affects only the current connector group. You can continue to use the same functional profile in other connector groups.
5. Create a fiscal registration process.

A fiscal registration process is defined by the sequence of registration steps and the connector group that is used for each step.

- a. On the **Fiscal registration process** page (**Retail and Commerce > Channel setup > Fiscal integration > Fiscal registration processes**), create a new record for each unique process of fiscal registration.
- b. Add registration steps to the process:
  - a. Select **Add**.
  - b. Select a fiscal connector type.

- c. In the **Group number** field, select an appropriate fiscal connector group.
6. Assign entities of the fiscal registration process to POS profiles.
    - a. On the **POS functionality profiles** page (**Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**), assign the fiscal registration process to a POS functionality profile. Select **Edit**, and then, on the **Fiscal registration process** tab, in the **Process number** field, select a process.
    - b. On the **POS hardware profile** page (**Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**), assign connector technical profiles to a hardware profile. Select **Edit**, add a line on the **Fiscal peripherals** tab, and then, in the **Profile number** field, select a connector technical profile.

#### NOTE

You can add several technical profiles to the same hardware profile. However, a hardware profile or POS functionality profile should have only one intersection with any fiscal connector group.

The fiscal registration flow is defined by the fiscal registration process and also by some parameters of fiscal integration components: the Commerce runtime extension for the fiscal document provider and the Hardware station extension for the fiscal connector.

- The subscription of events and transactions to fiscal registration is predefined in the fiscal document provider.
  - The fiscal document provider is also responsible for identifying the fiscal connector that is used for fiscal registration. It matches the connector functional profiles that are included in the fiscal connector group that is specified for the current step of the fiscal registration process with the connector technical profile that is assigned to the hardware profile of the Hardware station that the POS is paired to.
  - The fiscal document provider uses the data mapping settings from the fiscal document provider configuration to transform transaction/event data such as taxes and payments while a fiscal document is generated.
  - When the fiscal document provider generates a fiscal document, the fiscal connector can either send it to the fiscal device as is, or parse it and transform it into a sequence of commands of the device application programming interface (API), depending on how the communication is handled.
7. On the **Fiscal registration process** page (**Retail and Commerce > Channel setup > Fiscal integration > Fiscal registration processes**), select **Validate** to validate the fiscal registration process.

We recommend that you run this type of validation in the following cases:

- After you've completed all the settings for a new registration process, including when you assign registration processes to POS functionality profiles and hardware profiles.
  - After you make changes to an existing fiscal registration process, and those changes might cause a different fiscal connector to be selected at runtime (for example, if you change the connector group for a fiscal registration process step, enable a connector functional profile in a connector group, or add a new connector functional profile to a connector group).
  - After you make changes in the assignment of connector technical profiles to hardware profiles.
8. On the **Distribution schedule** page, run the **1070** and **1090** jobs to transfer data to the channel database.

## Set up fiscal texts for discounts

In some cases, a special text must be printed on a fiscal receipt if a discount is applied. You can set up fiscal texts

for discounts on the **Fiscal connector group** page (**Retail and Commerce > Channel setup > Fiscal integration > Fiscal connector groups**).

- For manual discounts that are applied at the POS, you should set a fiscal text for the info code or info code group that is specified as the **Product discount** info code in the POS functionality profile.
  1. On the **Fiscal connector group** page, select **Text for fiscal receipt**.
  2. On the **Info codes** tab, select **Add**, and select an info code or info code group.
  3. In the **Info code number**, select a value.
  4. In the **Subcode number** field, select a value if a subcode is required for the selected info code.
  5. In the **Text for fiscal receipt** field, specify a fiscal text that should be printed on a fiscal receipt.
  6. Set the **Print user input on fiscal receipt** option to **Yes** to override the text on a fiscal receipt with information that a user manually enters at the POS. This option applies only to info codes that have an input type of **Text**.

#### NOTE

You can specify a fiscal text for several info codes to support scenarios where info code groups, linked info codes, and triggered info codes are used. In these scenarios, the fiscal receipt will contain the fiscal texts from all info codes that are linked to the transaction line where the discount was applied.

- For channel-specific discounts, you should define a fiscal text for the discount ID.
  1. On the **Fiscal connector group** page, select **Text for fiscal receipt**.
  2. On the **Discounts** tab, select **Add**, and select a discount ID.
  3. In the **Text for fiscal receipt** field, specify a fiscal text that should be printed on a fiscal receipt.

#### NOTE

If several discounts are applied to the same transaction line, the fiscal receipt will contain fiscal texts from all discounts that are linked to those transaction line.

## Set error handling settings

The error handling options that are available in the fiscal integration are set in the fiscal registration process. For more information about error handling in the fiscal integration, see [Error handling](#).

1. On the **Fiscal registration process** page (**Retail and Commerce > Channel setup > Fiscal integration > Fiscal registration processes**), you can set the following parameters for each step of the fiscal registration process:
  - **Allow skip** – This parameter enables the **Skip** option in the error handling dialog box.
  - **Allow mark as registered** – This parameter enables the **Mark as registered** option in the error handling dialog box.
  - **Continue on error** – If this parameter is enabled, the fiscal registration process can continue on the POS register if the fiscal registration of a transaction or event fails. Otherwise, to run the fiscal registration of the next transaction or event, the operator must retry the failed fiscal registration, skip it, or mark the transaction or event as registered. For more information, see [Optional fiscal registration](#).

#### NOTE

If the **Continue on error** parameter is enabled, the **Allow skip** and **Allow mark as registered** parameters are automatically disabled.

2. The **Skip** and **Mark as registered** options in the error handling dialog box require the **Allow skip registration or mark as registered** permission. Therefore, on the **Permission groups** page (**Retail and Commerce > Employees > Permission groups**), enable the **Allow skip registration or mark as registered** permission.
3. The **Skip** and **Mark as registered** options let operators enter additional information when fiscal registration fails. To make this functionality available, you should specify the **Skip** and **Mark as registered** info codes on a fiscal connector group. The information that operators enter is then saved as an info code transaction that is linked to the fiscal transaction. For more details about info codes, see [Info codes and info code groups](#).

#### NOTE

The **Product** trigger function isn't supported for the info codes that are used for **Skip** and **Mark as registered** in fiscal connector groups.

- On the **Fiscal connector group** page, on the **Info codes** tab, select info codes or info code groups in the **Skip** and **Mark as registered** fields.

#### NOTE

One fiscal document and one non-fiscal document can be generated on any step of a fiscal registration process. A fiscal document provider extension identifies every type of transaction or event as related to fiscal or non-fiscal documents. The error handling feature applies only to fiscal documents.

- **Fiscal document** – A mandatory document that should be registered successfully (for example, a fiscal receipt).
- **Non-fiscal document** – A supplementary document for the transaction or event (for example, a gift card slip).

4. If the operator must be able to continue to process the current operation (for example, creation or finalization of a transaction) after a health check error occurs, you should enable the **Allow skip health check error** permission on the **Permission groups** page (**Retail and Commerce > Employees > Permission groups**). For more information about the health check procedure, see [Fiscal registration health check](#).

## Set up fiscal X/Z reports from the POS

To enable fiscal X/Z reports to be run from the POS, you should add new buttons to a POS layout.

- On the **Button grids** page, follow the instructions in [Add POS operations to POS layouts by using Button grid designer](#) to install the designer and update a POS layout.
  1. Select the layout to update.
  2. Add a new button, and set the **Print fiscal X** button property.
  3. Add a new button, and set the **Print fiscal Z** button property.
  4. On the **Distribution schedule** page, run the **1090** job to transfer changes to the channel database.

# Enable manual execution of postponed fiscal registration

To enable manual execution of a postponed fiscal registration, you should add a new button to a POS layout.

- On the **Button grids** page, follow the instructions in [Add POS operations to POS layouts by using Button grid designer](#) to install the designer and update a POS layout.
  1. Select the layout to update.
  2. Add a new button, and set the **Complete fiscal registration process** button property.
  3. On the **Distribution schedule** page, run the **1090** job to transfer your changes to the channel database.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# User-defined certificate profiles for retail stores

2/18/2021 • 6 minutes to read • [Edit Online](#)

## Overview

This topic provides an overview of the certificate profiles that are available in Microsoft Dynamics 365 Commerce. This functionality extends the [Manage secrets for retail channels](#) feature by adding support for local certificates.

While the point of sale (POS) is running in offline mode, it can't access the certificates that are stored in the key vault. The local certificate should be used instead. The following capabilities are supported:

- Using local certificates in key vault fallback scenarios
- Using local certificates without a key vault (for example in an on-premises installation)
- Gradual update of certificates, where some stores and terminals use a new version of the certificate, but other stores and terminals continue to use the previous version

The certificate profiles functionality lets you specify a default certificate and set the order that certificates in the same certificate profile are searched in. This functionality also provides a similar setup approach for local certificates and Key Vault certificates. You can add company-specific settings for certificates, but the unique cross-company identifier for each certificate can be used in the Commerce channels.

## Scenarios

The certificate profiles functionality supports the following scenarios in the Commerce channels:

- Use a local certificate in key vault fallback scenarios. Here are some examples of these fallback scenarios:
  - The key vault storage isn't accessible.
  - A certificate isn't found in the key vault storage.
  - The POS is running in offline mode.
- Use local certificates, but without storing them in the key vault (for example, in an on-premises installation).
- Do a gradual update of certificates, where a new version of the certificate is used only in stores or on terminals where the new version is already available.
- Use the same certificate in several companies.

## Set up certificate profiles

The following procedure explains how to set up certificate profiles. Before you use certificate profiles in the Commerce channels, follow these steps to configure the settings.

1. In the **Feature management** workspace, turn on the **User-defined certificate profiles for retail stores** feature.
2. Go to **System administration > Setup > Certificate profiles**.
3. Create a record, and set the **Certificate profile**, **Name**, and **Description** fields for it.



#### NOTE

The certificate profile is a unique identifier of a certificate across all companies and Commerce components.

4. On the **Legal entities** tab, add a line, and select the legal entity (company) that the current certificate profile should be used for. If the certificate profile should be used for multiple legal entities, repeat this step to add a line for each additional legal entity.
5. Select **Settings** to open the **Certificate profile settings** page, where you can enter company-specific settings for the certificate profile.

### Certificate profile settings

When you select **Settings** for certificate profile lines, the **Certificate profile settings** page appears. This page lets you specify which certificates can be used when the current certificate profile is called in the Commerce channels. You can also specify the order that certificates should be searched in.

#### NOTE

The **Priority** field is automatically set. A value of **1** represents the highest priority. When a new line is added on the **Certificate profile settings** page, its priority is set to a number that is one more than the priority of the previous line. To change the priority of a specific line, select the line, and then select either **Move up** to increase the priority or **Move down** to decrease the priority.

When you add a new line to the **Certificate profile settings** page, set the following fields:

- **Location type** – Select the location where the certificate is stored. This field has two possible values: **Local certificate** and **Key Vault**.
- **Key Vault certificate** – This field is required if you set the **Location type** field to **Key Vault**. Use it to specify a Key Vault certificate secret.

#### NOTE

Before you use a Key Vault certificate in certificate profiles, be sure to upload a certificate to the key vault storage, and follow the instructions in [Set up the Azure Key Vault client](#).

- **Store name** – This field is optional and is available only if you set the **Location type** field to **Local certificate**. Use it to specify a default store name that should be used to search local certificates.
- **Store location** – This field is optional and is available only if you set the **Location type** field to **Local certificate**. Use it to specify a default store location that should be used to search local certificates.

#### NOTE

The default store name and store location are added to simplify the process of searching local certificates in the Commerce runtime. X509StoreProvider has a list of folders where certificates are stored. If the default store name and the default store location aren't specified, X509StoreProvider tries to find a certificate in the other folders on its list.

- **Thumbprint** – This field is required and available only if you set the **Location type** field to **Local certificate**. Use it to specify the certificate thumbprint.
- **Comments** – This field is optional and lets users enter notes.

## Workflow: Searching certificates in the Commerce runtime

Here is the basic workflow that is used to search for a certificate when a certificate profile is called in the Commerce runtime.

1. The system identifies whether the certificate profile has company-specific settings for the current legal entity.
2. The system tries to find the certificate by using the values on the **Certificate profile settings** page for the line where the **Priority** field is set to **1**.
  - If the **Location type** field is set to **Key Vault**, the value of the **Key Vault certificate secret** field is used to search for the certificate on the **Key vault parameters** page. The certificate is then searched for in the key vault storage.
  - If the **Location type** field is set to **Local certificate**, X509StoreProvider first searches for the certificate by using the default store name and store location, if these parameters are specified. It then searches in all other folders on its list of folders.
3. If the certificate isn't found, the process is repeated for the line where the **Priority** field is set to **2**, and so on.

### NOTE

If the certificate profile has no settings for the current legal entity, or if the certificate search is unsuccessful for all lines on the **Certificate profile settings** page, the certificate isn't found.

## Caching the results of certificate searches

The results of certificate searches are cached. The default expiration time for a cache is one hour. However, this time can be customized and can be set to a maximum value of 24 hours.

## Gradual update

If a new version of the certificate is introduced, but it can't be updated in all stores at the same time, the certificate profiles functionality enables the certificate to be updated gradually.

1. Find a certificate profile and the line that should be updated, and then select **Settings**.
2. Add a line, and specify settings that are related to the latest version of the certificate.
3. Increase the **Priority** value of the new line. Use the **Move up** button to move the line so that it's above the line for the previous version of the same certificate.

### NOTE

In the Commerce runtime, the new version of the certificate will be called first. If the certificate hasn't yet been updated in a specific store or on a specific terminal, the previous version will be called.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Fiscal registration service integration sample for Austria

2/18/2021 • 19 minutes to read • [Edit Online](#)

## Introduction

To meet local fiscal requirements for cash registers in Austria, the Dynamics 365 Retail functionality for Austria includes a sample integration of the point of sale (POS) with an external fiscal registration service. The sample extends the [fiscal integration functionality](#). It's based on the [EFR \(Electronic Fiscal Register\)](#) solution from [EFSTA](#) and enables communication with the EFR service via the HTTPS protocol. The EFR service should be hosted on either the Retail Hardware station or a separate machine that can be connected to from the Hardware station. The sample is provided in the form of source code and is part of the Retail software development kit (SDK).

Microsoft doesn't release any hardware, software, or documentation from EFSTA. For information about how to get the EFR solution and operate it, contact [EFSTA](#).

## Scenarios

The following scenarios are covered by the fiscal registration service integration sample for Austria:

- Registration of cash transactions in the fiscal registration service:
  - Send detailed transaction data to the fiscal registration service. This data includes sales line information, and information about discounts, payments, and taxes.
  - Capture a response from the fiscal registration service. This response includes a digital signature and a link to the registered transaction.
  - Register taxes, and map them to the fiscal registration service's tax codes.
  - Print the QR code for a registered transaction on the receipt.
- Registration of gift card operations and customer deposits as non-cash transactions in the fiscal registration service:
  - Issue or add money to a gift card.
  - Register a customer account deposit.
  - Register a customer order deposit.
- Registration of non-sales transactions and events as non-cash transactions in the fiscal registration service:
  - Open shift and Close shift
  - Start amount, Float entry, and Tender removal
  - Price override
  - Tax override
  - Print copy of receipt
  - Open drawer
  - Print X report
  - Print Z report
- Printing end-of-day statements (X/Z reports) that have Austria-specific fields:
  - Total number of products or services that were delivered to customers
  - Breakdown of sales by tax rate

- Breakdown of payments by cashier/cash register operator
- Price discounts and returns that reduce daily sales
- Zero sales (giveaways)
- Error handling, such as the following options:
  - Retry fiscal registration if a retry is possible, such as if the fiscal registration service isn't available, isn't ready, or isn't responding.
  - Postpone fiscal registration.
  - Skip fiscal registration, or mark the transaction as registered, and include info codes to capture the reason for the failure and additional information.
  - Check the availability of the fiscal registration service before a new sales transaction is opened or a sales transaction is finalized.

### Default data mapping

The following default data mapping is included in the fiscal document provider configuration that is provided as part of the fiscal integration sample:

- Value-added tax (VAT) rates mapping:  
*A: 20.00; B: 10.00; C: 13.00; D: 0.00; E: 19.00; F: 7.00*

### Gift cards

The fiscal registration service integration sample implements the following rules that are related to gift cards:

- Exclude sales lines that are related to the *Issue gift card* and *Add to gift card* operations from a cash transaction. Instead of registering those lines as a part of a cash transaction, register them as a separate non-cash transaction in the fiscal registration service.
- Don't print a tax group breakdown and a QR code on a receipt if the receipt consists only of gift card lines.
- Print the total amount of gift cards that are issued or re-charged in a transaction separately from the cash transaction amount on the receipt.
- Save calculated adjustments of payment lines in the channel database with a reference to a corresponding fiscal transaction.
- Payment by gift card is considered a regular payment.

### Customer deposits and customer order deposits

The fiscal registration service integration sample implements the following rules that are related to customer deposits and customer order deposits:

- Register a non-cash transaction if a transaction is a customer deposit.
- Register a non-cash transaction if a transaction contains only a customer order deposit or a customer order deposit refund.
- Deduct the customer order deposit amount from payment lines when a hybrid customer order is created.
- Save calculated adjustments of payment lines in the channel database with a reference to a fiscal transaction for a hybrid customer order.

### Limitations of the sample

The fiscal registration service supports only scenarios where sales tax is included in the price. Therefore, the **Price include sales tax** option must be set to **Yes** for both stores and customers.

## Set up Commerce for Austria

This section describes the Commerce settings that are specific to and recommended for Austria. For more information set up information, see [Commerce home page](#).

To use the Austria-specific functionality, you must specify the following settings:

- In the primary address of the legal entity, set the **Country/region** field to **AUT** (Austria).
- In the POS functionality profile of every store that is located in Austria, set the **ISO code** field to **AT** (Austria).

You must also specify the following settings for Austria. Note that you must run appropriate distribution jobs after you complete the setup.

### Set up VAT per Austrian requirements

You must create sales tax codes, sales tax groups, and item sales tax groups. You must also set up sales tax information for products and services. For more information about how to set up and use sales tax features, see [Sales tax overview](#).

On sales receipts, you can print an abbreviated code for a sales tax code (for example, "A" or "B"). To make this functionality available, set the **Print code** field on the **Sales tax codes** page.

### Set up stores

On the **All stores** page, update the store details. Specifically, set the following parameters:

- In the **Sales tax group** field, specify the sales tax group that should be used for sales to the default customer.
- Set the **Prices include sales tax** option to **Yes**.
- Set the **Name** field to the company name. This change helps guarantee that the company name appears on a sales receipt. Alternatively, you can add the company name to the sales receipt layout as free-form text.
- Set the **Tax identification number (TIN)** field to the company identification number. This change helps guarantee that the company identification number appears on a sales receipt. Alternatively, you can add the company identification number to the sales receipt layout as free-form text.

### Set up functionality profiles

Set up POS functionality profiles:

- On the **Receipt numbering** FastTab, set up receipt numbering by creating or updating records for the **Sale**, **Sales order**, and **Return** receipt transaction types.

### Configure custom fields so that they can be used in receipt formats for sales receipts

You can configure the language text and custom fields that are used in the POS receipt formats. The default company of the user who creates the receipt setup should be the same legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

On the **Language text** page, add the following records for the labels of the custom fields for receipt layouts. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the table are just examples. You can change them to meet to your requirements. However, the **Text ID** values that you use must be unique, and they must be equal to or more than 900001.

Add the following POS labels to the **POS** section of **Language text** from the table:

LANGUAGE ID	TEXT ID	TEXT
en-US	900001	QR Code
en-US	900002	Continuous Number
en-US	900003	Tax Retail Print Code

LANGUAGE ID	TEXT ID	TEXT
en-US	900004	Total (sales)
en-US	900005	Total Tax (sales)
en-US	900006	Total Include Tax (sales)
en-US	900007	Tax Amount (sales)
en-US	900008	Tax Basis (sales)

On the **Custom fields** page, add the following records for the custom fields for receipt layouts. Note that **Caption text ID** values must correspond to the **Text ID** values that you specified on the **Language text** page:

NAME	TYPE	CAPTION TEXT ID
QRCODE	Receipt	900001
CONTINUOUSNUMBER	Receipt	900002
RETAILPRINTCODE	Receipt	900003
SALESTOTAL	Receipt	900004
SALESTOTALTAX	Receipt	900005
SALESTOTALINCLUDETAX	Receipt	900006
SALESTAXAMOUNT	Receipt	900007
SALESTAXBASIS	Receipt	900008

### Configure receipt formats

For every required receipt format, change the value of the **Print behavior** field to **Always print**.

In the Receipt format designer, add the following custom fields to the appropriate receipt sections. Note that field names correspond to the language texts that you defined in the previous section.

- **Header:** Add the following fields:
  - **Store name** and **Tax Identification Number** fields, which are used to print the company name and identity number on receipts. Alternatively, you can add the company name and identity number to the layout as free-form text.
  - **Store address**, **Date**, **Time 24H**, **Receipt Number**, and **Register number** fields.
  - **Continuous Number** fields, to identify the number of the cash transaction in the fiscal registration service.
- **Lines:** Add the following fields:
  - **Item name**.
  - **Qty**.
  - **Total price with tax**.
  - **Tax Retail Print Code**, which is used to print the abbreviated code that corresponds to the sales tax

code that applies to the item.

- **Footer:** Add the following fields:
  - Payment fields, so that the payment amounts for each payment method are printed. For example, add the **Tender name** and **Tender amount** fields to one line of the layout.
  - **Sales total** field group:
    - **Total (sales)** field, which is used to print the receipt's total cash sale amount. The amount excludes tax. Prepayments and gift card operations are excluded.
    - **Total Include Tax (sales)** field, which is used to print the receipt's total cash sale amount. The amount includes tax. Prepayments and gift card operations are excluded.
    - **Total Tax (sales)** field, which is used to print the receipt's total tax amount for cash sales. Prepayments and gift card operations are excluded.
  - **Tax break down** field group. The fields in this field group must be printed on a separate line.
    - **Tax Id** field, which is a standard field that enables a sales tax summary to be printed for each sales tax code. The field must be added to a new line.
    - **Tax Percentage** field, which is a standard field that is used to print the effective tax rate for the sales tax code.
    - **Tax Basis (sales)** field, which is used to print the receipt's total cash sale amount for the sales tax code. Prepayments and gift card operations are excluded.
    - **Tax Amount (sales)** field, which is used to print the receipt's tax amount for cash sales for the sales tax code.
    - **Tax Retail Print Code** field, which is used to print the abbreviated code that corresponds to the sales tax code.
  - **QR Code** field, which is used to print the reference to the registered cash transaction in the form of QR code.

For more information about how to work with receipt formats, see [Set up and design receipt formats](#).

### Configure fiscal integration

Complete the fiscal integration setup steps as described in [Set up the fiscal integration for Commerce channels](#):

- [Set up a fiscal registration process](#). Note also the settings for the fiscal registration process that are [specific to this fiscal registration service integration sample](#).
- [Set error handling settings](#).
- [Enable manual execution of postponed fiscal registration](#).

## Deployment guidelines for cash registers for Austria

The fiscal registration service integration sample for Austria is part of the Retail SDK. For information about how to install and use the SDK, see the [Retail software development kit \(SDK\) architecture](#).

This sample consists of extensions for the CRT, Hardware station, and POS. To run this sample, you must modify and build the CRT, Hardware station, and POS projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Azure DevOps, where no files have been changed yet.

Follow these steps to set up a development environment so that you can test and extend the sample.

### Enable Commerce runtime extensions

The CRT extension components are included in the CRT samples. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`, under `RetailSdk\SampleExtensions\CommerceRuntime`.

#### DocumentProvider.EFRSample component

1. Find the `Runtime.Extensions.DocumentProvider.EFRSample` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.EFRSample\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.EFRSample.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit:** Copy the assembly to the `\bin\ext` folder under the Microsoft Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.EFRSample" />
```

#### DocumentProvider.DataModelEFR component

1. Find the `Runtime.Extensions.DocumentProvider.DataModelEFR` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.DataModelEFR\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit:** Copy the assembly to the `\bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
```

#### Update extension configuration file

1. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
2. Register the CRT change in the extension configuration file.



```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsAustria" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.RegisterAuditEventAustria" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.XZReportsAustria" />
```

### Enable Hardware station extensions

The Hardware station extension components are included in the Hardware station samples. To complete the following procedures, open the solution, **HardwareStationSamples.sln.sln**, under **RetailSdk\SampleExtensions\HardwareStation**.

#### EFRSample component

1. Find the **HardwareStation.Extension.EFRSample** project, and build it.
2. In the **Extension.EFRSample\bin\Debug** folder, find following files:
  - The **Contoso.Commerce.HardwareStation.EFRSample.dll** assembly
  - The **Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll** assembly
3. Copy the assembly files to the Hardware station extensions folder:
  - **Shared hardware station:** Copy the files to the bin folder under the IIS Hardware station site location.
  - **Dedicated hardware station on Modern POS:** Copy the files to the Modern POS client broker location.
4. Find the extension configuration file for the Hardware station's extensions. The file is named **HardwareStation.Extension.config**.
  - **Shared hardware station:** The file is located under the IIS Hardware station site location.
  - **Dedicated hardware station on Modern POS:** The file is located under the Modern POS client broker location.
5. Add the following line to the **composition** section of the configuration file.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.EFRSample.dll" />
```

### Enable Modern POS extension components

1. Open the solution at **RetailSdk\POS\ModernPOS.sln**, and make sure that it can be compiled without errors. Additionally, make sure that you can run Modern POS from Microsoft Visual Studio by using the **Run** command.

#### NOTE

Modern POS must not be customized. You must enable User Account Control (UAC), and you must uninstall previously installed instances of Modern POS as required.

2. Enable the extensions to be loaded by adding the following lines in **extensions.json**.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/AuditEvent.AT"
    }
  ]
}
```

#### NOTE

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the readme.md file in the **Pos.Extensions** project.

3. Rebuild the solution.
4. Run Modern POS in the debugger, and test the functionality.

#### Enable Cloud POS extension components

1. Open the solution at **RetailSdk\POS\CloudPOS.sln**, and make sure that it can be compiled without errors.
2. Enable the extensions to be loaded by adding the following lines in **extensions.json**.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/AuditEvent.AT"
    }
  ]
}
```

#### NOTE

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the readme.md file in the **Pos.Extensions** project.

3. Rebuild the solution.
4. Run the solution by using the **Run** command and following the steps in the Retail SDK handbook.

#### Set up the registration process

To enable the registration process, follow these steps to set up Headquarters. For more details, see [Set up the fiscal integration for Commerce channels](#).

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**. On the **General** tab, set the **Enable fiscal integration** option to **Yes**.
2. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connectors**, and load the connector configuration. The file location is **RetailSdk\SampleExtensions\HardwareStation\Extension.EFRSample\Configuration\ConnectorEFRSample.xml**.
3. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal document providers**, and load the document provider configurations. The configuration files are located under **RetailSdk\SampleExtensions\CommerceRuntime\Extensions.DocumentProvider.EFRSample\C**onfiguration:
  - DocumentProviderEFRSampleAustria.xml
  - DocumentProviderNonFiscalEFRSampleAustria.xml
4. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector functional profiles**. Create two new connector functional profiles, one for each document provider that you loaded earlier, and select the connector that you loaded earlier. Update the data mapping settings as required.
5. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector technical**

**profiles.** Create a new connector technical profile, and select the connector that you loaded earlier. Update the connection settings as required.

6. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connector groups.** Create two new fiscal connector groups, one for each connector functional profile that you created earlier.
7. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal registration processes.** Create a new fiscal registration process, two fiscal registration process steps, and select the fiscal connector groups that you created earlier.
8. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles.** Select a functionality profile that is linked to the store where the registration process should be activated. On the **Fiscal registration process** FastTab, select the fiscal registration process that you created earlier. To enable registration of non-fiscal events on the POS, on the **Functions** FastTab, under **POS**, set the **Audit** option to **Yes**.
9. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles.** Select a hardware profile that is linked to the Hardware station that the fiscal printer will be connected to. On the **Fiscal peripherals** FastTab, select the connector technical profile that you created earlier.
10. Open the distribution schedule (**Retail and Commerce > Retail and Commerce IT > Distribution schedule**), and select jobs **1070** and **1090** to transfer data to the channel database.

### Production environment

The previous procedure enables the extensions that are components of the fiscal registration service integration sample. In addition, you must follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Make the following changes in the package configuration files under the **RetailSdk\Assets** folder:
  - In the **commerceruntime.ext.config** and **CommerceRuntime.MPOSOOffline.Ext.config** configuration files, add the following lines to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.EFRSample" />
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsAustria" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.RegisterAuditEventAustria" />
/>
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.XZReportsAustria" />
```

- In the **HardwareStation.Extension.config** configuration file, add the following line to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.EFRSample" />
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
```

2. Make the following changes in the **BuildTools\Customization.settings** package customization configuration file:

- Add the following lines to include the CRT extensions in the deployable packages.

```
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.EFRSample.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll" />
```

- Add the following line to include the Hardware station extension in the deployable packages.

```
<ISV_HardwareStation_CustomizableFile  
Include="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.EFRSample.dll" />  
<ISV_HardwareStation_CustomizableFile  
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll" />
```

3. Start the MSBuild Command Prompt for Visual Studio utility, and run **msbuild** under the Retail SDK folder to create deployable packages.
4. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).
5. Complete all the required setup tasks that are described in the [Set up Commerce for Austria](#) section.

## Design of extensions

### Commerce runtime extension design

The purpose of the extension that is a fiscal document provider is to generate service-specific documents and handle responses from the fiscal registration service.

The CRT extension is **Runtime.Extensions.DocumentProvider.EFRSample**.

For more details about the design of the fiscal integration solution, see [Overview of fiscal integration for Commerce channels](#).

#### Request handler

There are two request handlers for document providers:

- **DocumentProviderEFRFiscalAUT** – This handler is used to generate fiscal documents for the fiscal registration service.
- **DocumentProviderEFRNonFiscalAUT** – This handler is used to generate non-fiscal documents for the fiscal registration service.

These handlers are inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the connector document provider name that is specified in Headquarters.

The connector supports the following requests:

- **GetFiscalDocumentDocumentProviderRequest** – This request contains information about what document should be generated. It returns a service-specific document that should be registered in the fiscal registration service.
- **GetNonFiscalDocumentDocumentProviderRequest** – This request contains information about what non-fiscal document should be generated. It returns a service-specific document that should be registered in the fiscal registration service.
- **GetSupportedRegistrableEventsDocumentProviderRequest** – This request returns the list of events to subscribe to. Currently, the following events are supported: sales, printing X report, printing Z report, customer account deposits, customer order deposits, audit events, and non-sales transactions.
- **GetFiscalRegisterResponseToSaveDocumentProviderRequest** – This request returns the response from the fiscal registration service. This response is serialized to form a string so that it's ready to be saved.

#### Configuration

The configuration files are located in the **Configuration** folder of the extension project:

- **DocumentProviderFiscalEFRSampleAustria** – For fiscal documents.
- **DocumentProviderNonFiscalEFRSampleAustria** – For non-fiscal documents.

The purpose of these files is to enable settings for the document provider to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following setting is added:

- VAT rates mapping

### Hardware station extension design

The purpose of the extension that is a fiscal connector is to communicate with the fiscal registration service.

The Hardware station extension is **HardwareStation.Extension.EFRSample**. The Hardware station extension uses the HTTP protocol to submit documents that the CRT extension generates to the fiscal registration service. It also handles the responses that are received from the fiscal registration service.

### Request handler

The **EFRHandler** request handler is the entry point for handling requests to the fiscal registration service.

The handler is inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the fiscal connector name that is specified in Headquarters.

The connector supports the following requests:

- **SubmitDocumentFiscalDeviceRequest** – This request sends documents to the fiscal registration service and returns a response from it.
- **IsReadyFiscalDeviceRequest** – This request is used for a health check of the fiscal registration service.
- **InitializeFiscalDeviceRequest** – This request is used to initialize the fiscal registration service.

### Configuration

The configuration file is located in the **Configuration** folder of the extension project. The purpose of the file is to enable settings for the fiscal connector to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added:

- **Endpoint address** – The URL of the fiscal registration service.
- **Timeout** – The amount of time, in milliseconds, that the driver will wait for a response from the fiscal registration service.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce localization for Brazil

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes the scope of the Microsoft Dynamics 365 Commerce functionality that is specific to Brazil. It includes information about features and functionality that are designed to address specific federal tax, retail, accounting, financial, or statutory reporting laws or regulations that typically affect retail businesses in Brazil (within the scope of the [Brazilian localization](#)).

However, Commerce doesn't address all laws, regulations, or commercial requirements in Brazil, because laws and regulations vary in the way that they affect organizations. For more information, see the [Product localization and translation availability guide](#).

## Capabilities of the Commerce localization for Brazil

### Scope that is available in Brazil

The following features in Commerce headquarters and the point of sale (POS) are available to Commerce customers in Brazil:

- Retail product management and calculation of Brazil-specific taxes that are applicable to sales to final consumers. This feature includes an estimated tax breakdown in printed fiscal receipts.
- Generation of NFC-e (Nota Fiscal do Consumidor eletrônica) electronic fiscal documents for retail sales (model 65), submission of the electronic fiscal documents via the government's web services, and printing of DANFE (Documento Auxiliar da Nota Fiscal Eletrônica) NFC-e receipts.
- Cancellation of retail sales via the government's web services within the allowed timeframe.
- Generation of NF-e (Nota Fiscal eletrônica) electronic fiscal documents for retail returns (model 55), submission of the electronic fiscal documents via the government's web services, and printing of DANFE receipts.
- Generation of CF-e (Cupom Fiscal eletrônico) electronic fiscal documents for retail sales in São Paulo (model 59) and registration of the electronic fiscal documents in the SAT (Sistema Autenticador e Transmissor de Cupons Fiscais Eletrônicos) fiscal device.
- Electronic funds transfer (EFT) integration for the POS. This feature includes integration with popular local and global payment providers, and support for debit and credit card payments.
- Management of Brazil-specific customer registration numbers. This feature includes capabilities for entering, viewing, and modifying CNPJ (Cadastro Nacional da Pessoa Jurídica)/CPF (Cadastro de Pessoas Físicas) tax registration numbers or Foreigner IDs, and for registering these numbers in NFC-e, NF-e, and CF-e, and printed receipts.
- Postponed registration of electronic fiscal documents in the event of network failures (offline contingency mode), and subsequent transmission of electronic fiscal documents in contingency from Commerce headquarters.
- Control of electronic fiscal documents in Commerce headquarters. This feature includes capabilities for discarding the documents and registering a cancellation by substitution.
- Generation and submission of electronic fiscal documents for customer orders.
- The ability to issue linked NF-e over all types of registered fiscal documents.
- N-1 support, so that customers who run Microsoft Dynamics AX 2012 R3 in their stores can work with Microsoft Dynamics 365 Commerce headquarters after an upgrade.
- Support for Brazil-specific fields, such as CNPJ/CPF tax registration numbers, when customer master records are merged in a call center.

## Supported scenarios

The following scenarios are supported by Commerce localization for Brazil:

- Cash-and-carry sales of goods
- Cancellations and returns of cash-and-carry sales of goods
- Cash-and-carry sales of goods in offline contingency mode
- Cancellation by substitution
- Issuing gift cards and payments by gift cards
- Registering and processing customer orders in the POS
- Posting fiscal documents in retail statements in Commerce headquarters
- Sales via e-Commerce storefronts
- Call center sales

## Fiscal registration for Brazil

Fiscal registration is the immediate registration of retail sales per local fiscal laws that are aimed at preventing tax fraud in the retail industry. The following main fiscal registration methods are available in Brazil:

- Generation of an NFC-e electronic fiscal document for a retail sale, and submission of the document to tax authorities (SEFAZ [Secretaria de Fazenda]) via a dedicated web service that is maintained by SEFAZ.
- Registration of a retail sale, in a SAT fiscal device that is connected to the POS, by using a CF-e electronic fiscal document.
- Registration of a retail sale in a fiscal printer that is connected to the POS.

Commerce supports fiscal registration via the [Fiscal integration framework](#) and its extensions for specific countries or regions. Formats of electronic fiscal documents that meet the legal requirements in Brazil are configured by using the [Electronic reporting](#) functionality. Electronic fiscal documents are submitted via the [Electronic invoicing service](#). For more information about the submission process for electronic fiscal document and configuration for Brazil, see [Electronic invoicing add-on for Brazil](#).

### NOTE

Integration with fiscal printers isn't available in Commerce. You can take advantage of the [N-1 capabilities](#) to integrate Microsoft Dynamics AX 2012 R3 Retail Enterprise Point of Sale (EPOS), which supports the integration with fiscal printers for Brazil in Commerce headquarters.

## Availability of Commerce localization features for Brazil

FEATURE	PUBLIC PREVIEW	GENERAL AVAILABILITY (GA)	POST-GA	NOT PLANNED
Retail product management, and tax setup and calculation	X			
NFC-e (model 65) and DANFE for retail sales	X			
Communication with SEFAZ via <a href="#">Electronic invoicing service</a>	X			

FEATURE	PUBLIC PREVIEW	GENERAL AVAILABILITY (GA)	POST-GA	NOT PLANNED
Retail statements in Commerce headquarters	X			
Handling of fiscal customer information (for example, CPF/CNPJ)		X		
NF-e (model 55) and DANFE for sales returns		X		
CF-e (model 59) for sales in São Paulo and integration with a SAT device		X		
NFC-e/NF-e contingency in the POS (offline mode)		X		
Transmission of NFC-e/NF-e in contingency from Commerce headquarters		X		
NFC-e cancellation, discard, cancellation by substitution		X		
EFT integration (Adyen and basic capabilities)		X		
Searching for customers by registration numbers in the POS			X	
EFT integration (advanced capabilities and additional providers)			X	
Fiscal documents for customer orders from the POS			X	
NF-e linked to NFC-e/CF-e and DANFE			X	



FEATURE	PUBLIC PREVIEW	GENERAL AVAILABILITY (GA)	POST-GA	NOT PLANNED
N-1 support for upgrade from AX 2012 R3			X	
E-Commerce capabilities for Brazil			X	
Merging CNPJ/CPF in customer master records in the call center			X	
Retail fiscal documents in fiscal book statements*				X
Integration of the POS with fiscal printers				X

\* *Fiscal book statements* are SPED (Sistema Público de Escrituração Digital) Fiscal, SPED Contributions, and ICMS-ST (Imposto sobre Circulação de Mercadorias e Serviços - Substituição Tributária) compensation and restitution statements for the supported states.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Advance invoices for Commerce for Eastern Europe

2/18/2021 • 3 minutes to read • [Edit Online](#)

The information in this topic applies to the Eastern European localization and is specific to the commerce industry.

For Poland, Hungary, and Czech Republic, when a prepayment is received from a customer via Point of Sale (POS), the prepayment must be registered for tax purposes, and it's required to generate and print an advance invoice document that includes the prepayment amount. Additionally, for Poland, advance invoice transactions must be posted in the general ledger.

When the invoice for the sales order is finally posted, the final document should include the advance invoice, and any prepayments should be indicated.

If you generate sales orders from Accounts receivable, you must manually generate advance invoices by using the procedure in [Advance invoices for Eastern Europe](#). If you generate sales orders via POS, the system generates and posts the advance invoices for you.

## Supported scenarios

The following scenarios are supported:

- Create and post an advance invoice.
- Modify a deposit amount. If a customer decides to increase the amount of a deposit, an additional advance invoice is issued. For all other changes to the deposit amount (for example, if a customer order is edited), a credit note is created for the advance invoice that was previously generated, and a new advance invoice is generated and posted for the corrected amount.
- Cancel a sales order that has linked advance invoices. In this case, a credit note is created for the advance invoice.
- Post a sales order invoice that has linked advance invoices. The advance invoice that is linked to a sales order is reversed for the amount of the sales invoice. The advance invoice transactions are settled with advance invoice reversal transactions.

## Set up advance invoices

### Turn on the functionality for creating advance invoices

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**.
2. On the **Customer orders** tab, on the **Order** FastTab, set the **Create advance invoice for deposit** option to **Yes**.

### Define the parameters for posting advance invoices

1. Go to **Accounts receivable > Setup > Accounts receivable parameters**.
2. On the **Updates** tab, on the **Advance invoice** FastTab, set the **Posting profile**, **Sales tax group**, and **Item sales tax group** fields. If these fields are set correctly, the advance invoice will be posted. If these fields aren't set, advance invoices won't be posted.

### Turn off posting of the Sales tax on prepayment journal voucher

The Sales tax on prepayment journal voucher must not be posted if advance invoice posting is turned on. To verify that this requirement is met, follow these steps.

1. Go to **Accounts receivable > Setup > Accounts receivable parameters**.

2. On the **Ledger and sales tax** tab, on the **Payment** FastTab, make sure that the following fields are blank or set to **No**:

- Sales tax on prepayment journal voucher
- Posting profile with prepayment journal voucher
- Tax group for prepayment
- Item Sales tax group

## Print advance invoices

You can print advance invoices from POS on a Microsoft Windows printer that is connected to the hardware station. There are two options for printing an advance invoice from POS:

- **Print an advance invoice after a transaction is concluded in POS.** This option occurs automatically if an advance invoice was generated and a Windows printer was correctly set up. In this case, only the last advance invoice that is linked to the customer order is printed.
- **Reprint an advance invoice from the transaction journal.** Select **Show journal** to open the transactions journal, find the customer order, and then select **Print Advance invoice**. In this case, all advance invoices that are linked to the customer order are printed.

### Set up a Windows printer

Follow these steps to enable documents to be printed from POS on a Windows printer that is connected to the hardware station.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**.
2. Select a hardware profile that is related to the store where the printer is used.
3. In either the **Printer** section or the **Printer 2** section, update the settings:
  - In the **Printer** field, select **Windows driver**.
  - In the **Device name** field, enter the name of the printer.
4. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
5. Select job **1090**, and then select **Run now**.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Petty cash management for Commerce for Eastern Europe

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article contains information about Eastern European localization specific for the commerce industry.

In accordance with the Eastern Europe accounting requirements, you can set up operations for cash accounts to automate the processes for receipts, cash documents and cash reports. For more information, go to [\(EEUR\) Set up parameters for cash management](#).

Retailers can accept various types of payment in exchange for the products and services that they sell. Although cash is the most common form of payment, retailers can also receive payment in the form of checks, cards, or vouchers. In Retail point of sale (POS), cash, credit card receipts, and other payments are processed through a cash office.

You can do the following by using Cash management in Commerce:

- Create a cash account for the selected payment method for each store.
- Use cash journals to post cash transactions and customer payments that are received at a retail POS.
- Aggregate transactions in a statement line when you post a statement. You can aggregate safe drops, bank drops, voucher transactions, remove tender transactions, float entry transactions, income transactions, expense transactions, customer payments, sales transactions, and return transactions.

All transactions that take place in POS are posted using a ledger journal. You can use cash payment journals, customer payment journals, and general journals to create and post the statements. For more information, go to [Create, calculate, and post statements for a retail store](#).

On the **Posted statements** page, on the Action Pane, you can do the following:

- Go to **Inquiries > Cash payment journal** to access the cash payment journals that are related to the statement.
- Go to **Inquiries > General journal** to access the ledger journals that are related to the statement, other than customer payments and cash payments.

## Set up for cash management for POS

You must complete the following setup procedure before you use cash management:

- Set up a payment method for each payment type that the retailer accepts on the **Payment methods** page. You can use different payment methods for posting transactions in POS. For more information about payment methods, see [Payment methods](#).
- Set up parameters for cash operations.
- Set up a payment method for cash payments in a store.

### Set up parameters for cash operations

You can set up parameters to create and post cash transactions in Commerce. You can use cash payment journals, customer payment journals, or general journals to post sales transactions and payment transactions in the POS. You can aggregate transactions that have the same properties when you post a statement.

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**. In the left pane, click **Posting**.

2. In the **Posting** area, on the **Aggregation** FastTab, set **Tender remove/float** to **Yes** to aggregate the remove tender transactions or float entry transactions that are associated with a statement line when you post the statement. A remove tender transaction is created when you withdraw cash from the POS cash drawer. A float entry transaction is created when you deposit cash in the POS cash drawer.
3. Activate the individual parameters listed below to aggregate the transactions that are associated with a statement line when you post the statement:
  - **Bank drop** – Aggregate bank transactions.
  - **Safe drop** – Aggregate safe transactions.
  - **Income/Expense transactions** – Aggregate income transactions or expense transactions.
  - **Voucher transactions** – Aggregate voucher transactions.
  - **Customer payments** – Aggregate customer payments.
  - **Sales and returns** – Aggregate sales and returns transactions.
4. On the **Payments** FastTab, select a default journal name for the following options:
  - **Customer payment journal** – This journal is used to post customer payments.
  - **Cash payment journal** – This journal is used to post cash payments.
  - **General journal** – This journal is used to post transactions other than cash payments and customer payments.

### Set up a payment method for cash payments in a store

Use the following procedure to set up a payment method for cash payments in a store.

1. Go to **Retail and Commerce > Channels > Stores > All stores**.
2. On the **All stores** list page, select the store to set up a payment method for.
3. On the Action Pane, on the **Set up** tab, in the **Set up** group, click **Payment methods**.
4. On the **Payment method** page, create or select a payment method.
5. On the **Posting** FastTab, in the **Account** field group, in the **Account type** field, select **Cash account**.

#### NOTE

You can select **Cash account** in the **Account type** field only if you select **Normal** or **Tender remove/float** in the **Function** field.

6. In the **Account number** field, select a cash account number for the payment method.
7. In the **Tender remove/float** field group, in the **Offset account** field, select the offset account to post remove tender or float entry transactions for the payment method.

#### NOTE

You must set up offset accounts for both the cash tender payment method and the remove tender or float entry payment method for a store. This creates balanced general ledger entries for remove tender or float entry transactions.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customer invoices and return sales orders in Eastern European countries

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to set up information for customer invoices and return sales orders in Eastern European countries.

You can set up the following information for customer invoices and return sales orders that are generated in Retail point of sale (POS).

- You can use sales tax groups to process returns by using return sales orders. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**. Open the **Posting > Invoice** tab, and then set **Use sales tax group for returns** to **Yes**.
  - To specify the sales tax group for returns that are made by a customer, on the **Customers** page, on the **Commerce** FastTab, in the **Sales tax group for returns** field, select a sales tax group. When you post a return sales order for a customer, the return sales order line is updated with the sales tax group for returns that is specified in the **Customers** form.
  - To specify a sales tax group for returns that are made at a retail POS by a customer, on the **Stores** page, on the **General** FastTab, in the **Sales tax group for returns** field, select a sales tax group. When you post a return sales order for a customer of a store, the return sales order line is updated with the sales tax group for returns that are specified on the **Stores** page.
- You can use the posting date of a customer invoice or a return sales order as the sales date of the invoice or return if the invoice or return does not have a default sales date. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**. Open the **Posting > Invoice** tab, and then set **Use posting date as sales date** to **Yes**.
- You can use the number range that is provided by the tax authorities to number Latvian and Lithuanian customer invoices and return sales orders.
  - Go to **Organization administration > Number sequences > Counters management**. There should be a record where **Module = Sales** and **Type = Invoice**.
  - Go to **Organization administration > Number sequences > Invoice numbering setup**. Select the **Commerce** check box for the number sequence line that is used to number the customer invoices.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Deployment guidelines for Advance Invoice report printing for Czech Republic, Hungary, and Poland

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic shows how to enable the Dynamics 365 Commerce localization for Czech Republic, Hungary, and Poland. The localization consists of several extensions of Commerce components. These extensions let you print the **Advance Invoice** report from Point of Sale (POS). For more information about localization for Czech Republic, Hungary, and Poland, see [Advance invoices for Commerce for Eastern Europe](#).

The localization is part of the Retail software development kit (SDK). For information, see the [Retail software development kit \(SDK\) architecture](#).

The localization consists of extensions for the Commerce runtime (CRT) and POS. To enable this localization, you must modify the CRT configuration file and modify and build POS projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Microsoft Visual Studio Team Services, where no files have been changed yet.

## Development environment

Complete these procedures to set up a development environment, so that you can test the functionality.

### CRT extension components

1. Find the extensions configuration file for CRT.

The file is named `commerceruntime.ext.config`, and is located in the `bin\ext` folder under the IIS Commerce Scale Unit site location.

2. Register the CRT change in the extensions configuration file.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.UseAdvanceInvoice" />
```

#### WARNING

Do **not** edit the `commerceruntime.config` file. This file isn't intended for any customizations.

### Modern POS extension components

1. Open the solution at `RetailSdk\POS\ModernPOS.sln`, and make sure that it can be compiled without errors. Additionally, make sure that you can run Modern POS from Microsoft Visual Studio by using the **Run** command.

#### NOTE

Modern POS must not be customized. You must enable User Account Control (UAC), and you must uninstall previously installed instances of Modern POS as required.

2. Enable the extensions to be loaded in `extensions.json` by adding the following lines in the appropriate location.

```
{
  "extensionPackages": [
    {
      "baseUri": "Microsoft/AdvanceInvoice"
    }
  ]
}
```

3. Rebuild the solution.
4. Run Modern POS in the debugger and test the functionality.

### Cloud POS extension components

1. Open the solution at `RetailSdk\POS\CloudPOS.sln`, and make sure that it can be compiled without errors.
2. Enable the extensions to be loaded in `extensions.json` by adding the following lines in the appropriate location.

```
{
  "extensionPackages": [
    {
      "baseUri": "Microsoft/AdvanceInvoice"
    }
  ]
}
```

3. Rebuild the solution.
4. Run Cloud POS in the debugger and test the functionality.

### Set up required parameters in Headquarters

For more information, see [Advance invoices for Commerce for Eastern Europe](#).

## Production environment

Follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Complete the steps in the **Cloud POS extension components** or **Modern POS extension components** sections earlier in this topic.
2. Make the following change in the package configuration files under the `RetailSdk\Assets` folder.

In the `commercerruntime.ext.config` configuration files, add the following lines to the `composition` section.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.UseAdvanceInvoice" />
```

3. Run `msbuild` for the Retail SDK to create deployable packages.
4. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Fiscal registration service integration sample for Czech Republic

2/18/2021 • 18 minutes to read • [Edit Online](#)

## Introduction

To meet local fiscal requirements for cash registers in the Czech Republic, the Dynamics 365 Commerce functionality for the Czech Republic includes a sample integration of the point of sale (POS) with an external fiscal registration service. The sample extends the [fiscal integration functionality](#). It's based on the [EFR \(Electronic Fiscal Register\)](#) solution from [EFSTA](#) and enables communication with the EFR service via the HTTPS protocol. The EFR service ensures Electronic Registration of Sales (EET - Elektronická evidence tržeb), that is, the online transmission of the sales data to a fiscal web service of tax authorities.

The EFR service should be hosted on either the Commerce Hardware station or a separate machine that can be connected to from the Hardware station. The sample is provided in the form of source code and is part of the Retail software development kit (SDK).

Microsoft doesn't release any hardware, software, or documentation from EFSTA. For information about how to get the EFR solution and operate it, contact [EFSTA](#).

## Scenarios

The following scenarios are covered by the fiscal registration service integration sample for the Czech Republic.

- Registration of cash transactions in the fiscal registration service.
  - Send detailed transaction data to the fiscal registration service. This data includes sales line information, and information about discounts, payments, and taxes. The fiscal registration service further sends the data to the web-service of tax authorities and receives a confirmation from it that includes the fiscal identification code of the transaction.
  - Capture a response from the fiscal registration service. This response includes fiscal data such as the fiscal identification code and the security code of the transaction, etc.
  - Print the fiscal data for a registered transaction on the receipt.
- Registration of gift card operations and customer deposits in the fiscal registration service.
  - Issue or add money to a gift card.
  - Register a customer account deposit.
  - Create a customer order and register a deposit for the order.
  - Edit a customer order and override the deposit for the order.
  - Cancel a customer order and refund the deposit for the order.
- Error handling, such as the following options.
  - Retry fiscal registration if a retry is possible, such as if the fiscal registration service isn't available, isn't ready, or isn't responding.
  - Postpone fiscal registration.
  - Skip fiscal registration, or mark the transaction as registered, and include info codes to capture the reason for the failure and additional information.
  - Check the availability of the fiscal registration service before a new sales transaction is opened or a sales transaction is finalized.

### Default data mapping

The following default data mapping is included in the fiscal document provider configuration that is provided as part of the fiscal integration sample.

- Value-added tax (VAT) rates mapping:

*A: 21.00; B: 15.00; C: 10.00; Z: 0.00*

- Default VAT group mapping. Any VAT amounts that cannot be mapped to one of the predetermined VAT groups will be attributed to the default (basic) VAT group:

*A*

- Deposit VAT group mapping. Customer deposit amounts and customer order deposit amounts will be attributed to the deposit VAT group:

*Z*

### Gift cards

The fiscal registration service integration sample implements the following rules that are related to gift cards.

- Sales lines that are related to the *Issue gift card* or *Add to gift card* operations in a sales transaction are marked with a special attribute when the transaction is registered in the fiscal registration service.
- A payment by gift card is considered a regular payment and marked with a special attribute when the

transaction is registered in the fiscal registration service.

#### Customer account deposits and customer order deposits

The fiscal registration service integration sample implements the following rules that are related to customer account deposits and customer order deposits.

- A transaction that is related to a customer account deposit or a customer order deposit is registered in the fiscal registration service as a single line transaction and is marked with a special attribute. The deposit VAT group is specified in this line.
- When a hybrid customer order is created, that is, a customer order that contains products that can be carried out of the store by the customer, as well as products that will be picked up or shipped later, the transaction registered in the fiscal registration service contains lines for the products that are carried out, as well as a line for the order deposit.
- A payment from a customer account is considered a regular payment and marked with a special attribute when the transaction is registered in the fiscal registration service.
- The customer order deposit amount that is applied to a customer order *Pick up* operation is considered a regular payment and marked with a special attribute when the transaction is registered in the fiscal registration service.

#### Offline registration

If the fiscal registration service fails to transmit transaction data to the fiscal web service of tax authorities (e.g. due to the response timeout) and to receive a confirmation from the web-service (that is, the fiscal identification code of the transaction), it generates a local signature for the transaction and includes it and a special error code in the response. The fiscal registration service resends transactions in original order in background as soon as the network connection is restored.

#### Limitations of the sample

The fiscal registration service supports only scenarios where sales tax is included in the price. Therefore, the **Price include sales tax** option must be set to **Yes** for both stores and customers.

## Set up Commerce for Czech Republic

This section describes the Commerce settings that are specific to and recommended for the Czech Republic. For more information, see [Commerce home page](#).

To use the Czech-specific functionality, you must specify the following settings.

- In the primary address of the legal entity, set the **Country/region** field to **CZE** (Czech Republic).
- In the POS functionality profile of every store that is located in the Czech Republic, set the **ISO code** field to **CZ** (Czech Republic).

You must also specify the following settings for the Czech Republic. Note that you must run appropriate distribution jobs after you complete the setup.

#### Set up VAT per Czech Republic

You must create sales tax codes, sales tax groups, and item sales tax groups. You must also set up sales tax information for products and services. For more information about how to set up and use sales tax features, see [Sales tax overview](#).

#### Set up stores

On the **All stores** page, update the store details. Specifically, set the following parameters.

- In the **Sales tax group** field, specify the sales tax group that should be used for sales to the default customer.
- Set the **Prices include sales tax** option to **Yes**.
- Set the **Name** field to the company name. This change helps guarantee that the company name appears on a sales receipt. Alternatively, you can add the company name to the sales receipt layout as free-form text.
- Set the **Tax identification number (TIN)** field to the company identification number. This change helps guarantee that the company identification number appears on a sales receipt. Alternatively, you can add the company identification number to the sales receipt layout as free-form text.

#### Set up functionality profiles

Set up POS functionality profiles.

- On the **Receipt numbering** FastTab, set up receipt numbering by creating or updating records for the **Sale**, **Sales order**, and **Return** receipt transaction types.

#### Set up registration numbers

1. Go to **Organization administration > Global address book > Registration types > Registration types**. Create a new registration type. Specify the **Country/region** field to **CZE** (Czech Republic) and make it restricted to Organization.
2. Go to **Organization administration > Global address book > Registration types > Registration categories**. Create a new registration category. Select the registration type from the previous step and set the **Registration category** to **Business Premise ID**.
3. Go to **Organization administration > Organizations > Operating units**. For each store located in the Czech Republic, select the unit related to the store. On the **Address** FastTab expand the **More**

options drop-down list and select **Advanced**.

4. On the opened **Manage addresses** page you must specify following setting.

- On the **Address** FastTab set the **Country/region** field to **CZE**.
- On the **Registration ID** FastTab create a new record. Select the registration type created earlier and set the registration number.

#### Configure custom fields so that they can be used in receipt formats for sales receipts

You can configure the language text and custom fields that are used in the POS receipt formats. The default company of the user who creates the receipt setup should be the same legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

On the **Language text** page, add the following records for the labels of the custom fields for receipt layouts. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the table are just examples. You can change them to meet your requirements. However, the **Text ID** values that you use must be unique, and they must be equal to or more than 900001.

Add the following POS labels to the **POS** section of **Language text** from the table:

LANGUAGE ID	TEXT ID	TEXT
en-US	900001	ID provozovny/pokladny
en-US	900002	BKP
en-US	900003	FIK
en-US	900004	PKP
en-US	900005	Info
en-US	900006	Sequence number

On the **Custom fields** page, add the following records for the custom fields for receipt layouts. Note that **Caption text ID** values must correspond to the **Text ID** values that you specified on the **Language text** page:

NAME	TYPE	CAPTION TEXT ID
TLT	Receipt	900001
SEC	Receipt	900002
SIGN	Receipt	900003
FISCAL	Receipt	900004
INFO	Receipt	900005
CONTINUOUSNUMBER	Receipt	900006

#### Configure receipt formats

For every required receipt format, change the value of the **Print behavior** field to **Always print**.

In the Receipt format designer, add the following custom fields to the appropriate receipt sections. Note that field names correspond to the language texts that you defined in the previous section.

- **Header:** Add the following fields.
  - **Store name** and **Tax Identification Number:** these fields are used to print the company name and identity number on receipts. Alternatively, you can add the company name and identity number to the layout as free-form text.
  - **Store address**, **Date**, **Time 24H**, **Receipt Number**, and **Register number**.
  - **Sequence number:** this field identifies the number of the cash transaction in the fiscal registration service.
- **Lines:** Add the following fields.
  - **Item name**
  - **Qty**
  - **Total price with tax**
- **Footer:** Add the following fields.
  - Payment fields, so that the payment amounts for each payment method are printed. For example, add the **Tender name** and **Tender amount** fields to one line of the layout.
  - **ID provozovny/pokladny:** this field prints the identifiers of the business premises and the cash register.

- **BKP**: this field prints the taxpayer's security code that is assigned by the fiscal registration service.
- **FIK**: this field prints the fiscal identification code of the transaction that is assigned by the web-service of tax authorities in case of successful online registration.
- **PKP**: this field prints the taxpayer's signature code that is generated by the fiscal registration service in case of offline registration.
- **Info**: this field prints the additional information from the fiscal registration service.

For more information about how to work with receipt formats, see [Set up and design receipt formats](#).

### Configure fiscal integration

Complete the fiscal integration setup steps as described in [Set up the fiscal integration for Commerce channels](#).

- [Set up a fiscal registration process](#). Note also the settings for the fiscal registration process that are [specific to this fiscal registration service integration sample](#).
- [Set error handling settings](#).
- [Enable manual execution of postponed fiscal registration](#).

## Deployment guidelines for cash registers for Czech Republic

The fiscal registration service integration sample for the Czech Republic is part of the Retail SDK. For information about how to install and use the Retail SDK, see the [Retail software development kit \(SDK\) architecture](#).

This sample consists of extensions for the CRT and Hardware station. To run this sample, you must modify and build the CRT and Hardware station projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Azure DevOps, where no files have been changed yet.

Follow these steps to set up a development environment so that you can test and extend the sample.

### Enable Commerce runtime extensions

The CRT extension components are included in the CRT samples. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`, under `RetailSdk\SampleExtensions\CommerceRuntime`.

#### DocumentProvider.EFRSample component

1. Find the `Runtime.Extensions.DocumentProvider.EFRSample` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.EFRSample\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.EFRSample.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit**: Copy the assembly to the `\bin\ext` folder under the Microsoft Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS**: Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit**: The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS**: The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.EFRSample" />
```

#### DocumentProvider.DataModelEFR component

1. Find the `Runtime.Extensions.DocumentProvider.DataModelEFR` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.DataModelEFR\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit**: Copy the assembly to the `\bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS**: Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit**: The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS**: The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
```

#### Update extension configuration file

1. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
2. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsCzechia" />
```

#### Enable Hardware station extensions

The Hardware station extension components are included in the Hardware station samples. To complete the following procedures, open the solution, `HardwareStationSamples.sln.sln`, under `RetailSdk\SampleExtensions\HardwareStation`.

##### EFRSample component

1. Find the `HardwareStation.Extension.EFRSample` project, and build it.
2. In the `Extension.EFRSample\bin\Debug` folder, find following files:
  - The `Contoso.Commerce.HardwareStation.EFRSample.dll` assembly
  - The `Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll` assembly
3. Copy the assembly files to the Hardware station extensions folder:
  - **Shared hardware station:** Copy the files to the `bin` folder under the IIS Hardware station site location.
  - **Dedicated hardware station on Modern POS:** Copy the files to the Modern POS client broker location.
4. Find the extension configuration file for the Hardware station's extensions. The file is named `HardwareStation.Extension.config`.
  - **Shared hardware station:** The file is located under the IIS Hardware station site location.
  - **Dedicated hardware station on Modern POS:** The file is located under the Modern POS client broker location.
5. Add the following line to the `composition` section of the configuration file.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.EFRSample.dll" />
```

#### Set up the registration process

To enable the registration process, follow these steps to set up Headquarters. For more details, see [Set up a fiscal registration process](#).

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**. On the **General** tab, set the **Enable fiscal integration** option to **Yes**.
2. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connectors**, and load the connector configuration. The file location is `RetailSdk\SampleExtensions\HardwareStation\Extension.EFRSample\Configuration\ConnectorEFRSample.xml`.
3. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal document providers**, and load the document provider configuration. The configuration file is `RetailSdk\SampleExtensions\CommerceRuntime\Extensions.DocumentProvider.EFRSample\Configuration\DocumentProviderFisc`.
4. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector functional profiles**. Create a new connector functional profile. Select the document provider and the connector that you loaded earlier. Update the data mapping settings as required.
5. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector technical profiles**. Create a new connector technical profile, and select the connector that you loaded earlier. Update the connection settings as required.
6. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connector groups**. Create a new fiscal connector group, for the connector functional profile that you created earlier.
7. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal registration processes**. Create a new fiscal registration process, a fiscal registration process step, and select the fiscal connector group that you created earlier.
8. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. Select a functionality profile that is linked to the store where the registration process should be activated. On the **Fiscal registration process** FastTab, select the fiscal registration process that you created earlier.
9. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**. Select a hardware profile that is linked to the Hardware station that the fiscal printer will be connected to. On the **Fiscal peripherals** FastTab, select the connector technical profile that you created earlier.
10. Open the distribution schedule (**Retail and Commerce > Retail and Commerce IT > Distribution schedule**), and select jobs **1070** and **1090** to transfer data to the channel database.

#### Production environment

The previous procedure enables the extensions that are components of the fiscal registration service integration sample. In addition, you must follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Make the following changes in the package configuration files under the **RetailSdk\Assets** folder.

- In the **commercerruntime.ext.config** and **CommerceRuntime.MPOSOOffline.Ext.config** configuration files, add the following lines to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.EFRSample" />
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsCzechia" />
```

- In the **HardwareStation.Extension.config** configuration file, add the following line to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.EFRSample" />
```

2. Make the following changes in the **BuildTools\Customization.settings** package customization configuration file.

- Add the following lines to include the CRT extensions in the deployable packages.

```
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.EFRSample.d11" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.d11" />
<ISV_HardwareStation_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.d11" />
```

- Add the following line to include the Hardware station extension in the deployable packages.

```
<ISV_HardwareStation_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.EFRSample" />
```

3. Start the MSBuild Command Prompt for Visual Studio utility, and run **msbuild** under the Retail SDK folder to create deployable packages.

4. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).

5. Complete all the required setup tasks that are described in the [Set up Commerce for Czech Republic](#) section.

## Design of extensions

### Commerce runtime extension design

The purpose of the extension that is a fiscal document provider is to generate service-specific documents and handle responses from the fiscal registration service.

The CRT extension is **Runtime.Extensions.DocumentProvider.EFRSample**.

For more details about the design of the fiscal integration solution, see [Fiscal registration process and fiscal integration samples for fiscal devices](#).

#### Request handler

There is a single **DocumentProviderEFRFiscalCZE** request handler for document provider, which is used to generate fiscal documents for the fiscal registration service.

This handler is inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the connector document provider name that is specified in Headquarters.

The connector supports the following requests.

- **GetFiscalDocumentDocumentProviderRequest** – This request contains information about what document should be generated. It returns a service-specific document that should be registered in the fiscal registration service.
- **GetSupportedRegistrableEventsDocumentProviderRequest** – This request returns the list of events to subscribe to. Currently, the following events are supported: sales, customer account deposits and customer order deposits.
- **GetFiscalRegisterResponseToSaveDocumentProviderRequest** – This request returns the response from the fiscal registration service. This response is serialized to form a string so that it's ready to be saved.

#### Configuration

The **DocumentProviderFiscalEFRSampleCzech** configuration file is located in the **Configuration** folder of the extension project. The purpose of this file is to enable settings for the document provider to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added.

- VAT rates mapping
- Default VAT group
- Deposit VAT group

### Hardware station extension design

The purpose of the extension that is a fiscal connector is to communicate with the fiscal registration service.

The Hardware station extension is **HardwareStation.Extension.EFRSample**. The Hardware station extension uses the HTTP protocol to submit documents that the CRT extension generates to the fiscal registration service. It also handles the responses that are received from the fiscal registration service.

#### Request handler

The **EFRHandler** request handler is the entry point for handling requests to the fiscal registration service.

The handler is inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the fiscal connector name that is specified in Headquarters.

The connector supports the following requests.

- **SubmitDocumentFiscalDeviceRequest** – This request sends documents to the fiscal registration service and returns a response from it.
- **IsReadyFiscalDeviceRequest** – This request is used for a health check of the fiscal registration service.
- **InitializeFiscalDeviceRequest** – This request is used to initialize the fiscal registration service.

#### Configuration

The configuration file is located in the **Configuration** folder of the extension project. The purpose of the file is to enable settings for the fiscal connector to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added.

- **Endpoint address** – The URL of the fiscal registration service.
- **Timeout** – The amount of time, in milliseconds, that the driver will wait for a response from the fiscal registration service.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Fiscal printer integration sample for Poland

2/18/2021 • 11 minutes to read • [Edit Online](#)

## Introduction

The Dynamics 365 Commerce functionality for Poland includes a sample integration of the point of sale (POS) with a fiscal printer. The sample extends the [fiscal integration functionality](#) and supports the POSNET THERMAL HD 2.02 protocol for fiscal printers from [Posnet Polska S.A.](#) The sample enables communication with a fiscal printer that is connected via a COM port by using a native software driver. It was implemented and tested by using a software emulator that Posnet provided for the Posnet Thermal HD FV EJ fiscal printer. The sample is provided in the form of source code and is part of the Retail software development kit (SDK).

Microsoft doesn't release any hardware, software, or documentation from Posnet. For information about how to get the fiscal printer and operate it, contact [Posnet Polska S.A.](#)

## Scenarios

The following scenarios are covered by the fiscal printer integration sample for Poland:

- Sales scenarios:
  - Print a fiscal receipt for cash-and-carry sales and returns.
  - Capture a response from the fiscal printer, and store it in the channel database.
  - Taxes:
    - Map to the fiscal printer's tax codes (departments).
    - Transfer mapped tax data to the fiscal printer.
  - Payments:
    - Map to the fiscal printer's methods of payment.
    - Print payments on a fiscal receipt.
    - Print change information.
  - Print line discounts.
  - Gift cards:
    - Exclude an issued/re-charged gift card line from a fiscal receipt for a sale.
    - Print a payment that uses a gift card as a regular method of payment.
  - Print fiscal receipts for customer order operations:
    - A fiscal receipt isn't printed for a customer order deposit.
    - Print a fiscal receipt for carry-out lines of a hybrid customer order.
    - Print a fiscal receipt for the pickup operation for a customer order.
    - Print a fiscal receipt for a return order.
  - Print the [customer information](#) that is specified for a sales transaction on a fiscal receipt. An example of this information is the customer's VAT number.
- End of day statements (fiscal X and fiscal Z reports).
- Error handling, such as the following options:
  - Retry fiscal registration if a retry is possible, such as if the fiscal printer isn't connected, isn't ready, or isn't responding, the printer is out of paper, or there is a paper jam.
  - Postpone fiscal registration.
  - Skip fiscal registration, or mark the transaction as registered, and include info codes to capture the reason for the failure and additional information.
  - Check the availability of the fiscal printer before a new sales transaction is opened or a sales transaction is finalized.

### Default data mapping

The following default data mapping is included in the fiscal document provider configuration that is provided as part of the fiscal integration sample:

- Value-added tax (VAT) rates mapping:

*0 : 23.00 ; 1 : 8.00 ; 2 : 5.00 ; 3 : 0.00*

- Tender type mapping:

*0 : 0 ; 1 : 0 ; 2 : 2 ; 3 : 2 ; 4 : 0 ; 5 : 0 ; 6 : 0 ; 7 : 2 ; 8 : 0*

### Gift cards

The fiscal printer integration sample implements the following rules that are related to gift cards:

- Exclude sales lines that are related to the *Issue gift card* and *Add to gift card* operations from the fiscal receipt.
- Don't print a fiscal receipt if it consists only of gift card lines.
- Deduct the total amount of gift cards that are issued or re-charged in a transaction from payment lines of the fiscal receipt.
- Save calculated adjustments of payment lines in the channel database with a reference to a corresponding fiscal transaction.
- Payment by gift card is considered a regular payment.

#### Customer deposits and customer order deposits

The fiscal printer integration sample implements the following rules that are related to customer deposits and customer order deposits:

- Don't print a fiscal receipt if a transaction is a customer deposit.
- Don't print a fiscal receipt if a transaction contains only a customer order deposit or a customer order deposit refund.
- Print the amount of the previously paid deposit on a fiscal receipt for a customer order pickup operation.
- Deduct the customer order deposit amount from payment lines when a hybrid customer order is created.
- Save calculated adjustments of payment lines in the channel database with a reference to a fiscal transaction for a hybrid customer order.

#### Limitations of the sample

- The fiscal printer supports only scenarios where sales tax is included in the price. Therefore, the **Price include sales tax** option must be set to **Yes** for both stores and customers.
- Daily reports (fiscal X and fiscal Z) are printed by using the embedded *Shift report* format.
- Printing a bar code on fiscal receipts is considered a potential customization, because this feature isn't supported in the embedded formats and can be implemented only by using the customizable **Super-format** report.
- The fiscal printer doesn't support mixed transactions. The **Prohibit mixing sales and returns in one receipt** option should be set to **Yes** in POS functionality profiles.

## Set up Commerce for Poland

### Configure fiscal integration

Complete the fiscal integration setup steps as described in [Set up the fiscal integration for Commerce channels](#):

- [Set up a fiscal registration process](#). Note also the settings for the fiscal registration process that are [specific to this fiscal printer integration sample](#).
- [Set error handling settings](#).
- [Set up fiscal X/Z reports from the POS](#).
- [Enable manual execution of postponed fiscal registration](#).

### Enable extensions

#### Commerce runtime extension components

The Commerce runtime (CRT) extension components are included in the Retail SDK. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`, under

`RetailSdk\SampleExtensions\CommerceRuntime`.

1. Find the `Runtime.Extensions.DocumentProvider.PosnetSample` project, and build it.
2. In the `Extensions.DocumentProvider.PosnetSample\bin\Debug` folder, find the `Contoso.Commerce.Runtime.Extensions.DocumentProvider.PosnetSample.dll` assembly file.
3. Copy the assembly file to the CRT extension folder:
  - **Commerce Scale Unit:** Copy the assembly to the `\bin\ext` folder under the Microsoft Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extensions configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension's configuration file. Add `source="assembly"` and `value="Contoso.Commerce.Runtime.Extensions.DocumentProvider.PosnetSample"`.
6. Restart the Commerce service.
  - **Commerce Scale Unit:** Restart the Commerce service site from IIS Manager.
  - **Client broker:** End the `dllhost.exe` process in Task Manager, and then restart Modern POS.

### Hardware station extension components

The Hardware station extension components are included in the Retail SDK. To complete the following procedures, open the Hardware Station solution, `HardwareStationSamples.sln`, under `RetailSdk\SampleExtensions\HardwareStation`.

1. Find the `Extension.PosnetThermalFVFiscalPrinterSample` project, and build it.
2. In the `Extension.PosnetThermalFVFiscalPrinterSample\bin\Debug` folder, find the `Contoso.Commerce.HardwareStation.PosnetThermalFVFiscalPrinterSample.dll` assembly file.
3. Copy the files to a deployed Hardware station machine:
  - **Remote Hardware station:** Copy the files to the `bin` folder under the IIS Hardware station site location. Copy the printer driver libraries (`libposcmbth.dll`, `libcmbth_serial.dll`, and `cmbth_pl.Ing`).
4. Find the configuration file for the Hardware station's extensions. The file is named `HardwareStation.Extension.config`:
  - **Remote Hardware station:** The file is located under the IIS Hardware station site location.
5. Add the following section to the `composition` section of the config file.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.PosnetThermalFVFiscalPrinterSample" />
```

6. Restart the Hardware station service:
  - **Remote Hardware station:** Restart the Hardware station site from IIS Manager.

### Set up the registration process

To enable the registration process, follow these steps to set up Headquarters. For more details, see [Set up a fiscal registration process](#).

1. Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Fiscal Connectors**. Import the configuration from `RetailSdk\SampleExtensions\HardwareStation\Extension.Posnet.ThermalDeviceSample\Configuration\ConnectorConnectorPos`
2. Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Fiscal Document providers**. Import the configuration from `RetailSdk\SampleExtensions\CommerceRuntime\Extension.DocumentProvider.PosnetSample\Configuration\DocumentProvide`
3. Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Connector Technical profiles**. Create a new profile, and select the loaded connector from the earlier step. Update connection settings if an update is required.
4. Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Connector Functional profiles**. Create a new profile, and select the loaded connector and document provider from the earlier steps. Update data mapping settings, if an update is required.
5. Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Connector Functional group**. Create a new group, and select the connector functional profile from the earlier step.
6. Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Registration process**. Create a new process, and select the connector functional group from the earlier step.
7. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. Open the functionality profile that is linked to the store where the registration process should be activated. On the **Fiscal registration process** FastTab, select the registration process that was created earlier.
8. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**. Open the hardware profile that is linked to the Hardware station that the fiscal printer will be connected to. On the **Fiscal peripherals** FastTab, select the connector technical profile.
9. Open the distribution schedule (**Retail and Commerce > Retail and Commerce IT > Distribution schedule**), and select jobs `1070` and `1090` to transfer data to the channel database.

### Production environment

Follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Complete the steps that are described in the [Enable extensions](#) section earlier in this topic.
2. Make the following changes in the package configuration files under the `RetailSdk\Assets` folder:
  - In the `commerceruntime.ext.config` and `CommerceRuntime.MPOSOOffline.Ext.config` configuration files, add the following line to the `composition` section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.Extensions.DocumentProvider.PosnetSample" />
```

- In the `HardwareStation.Extension.config` configuration file, add the following line to the `composition` section.

```
<add source="assembly"
value="Contoso.Commerce.HardwareStation.PosnetThermalFVFiscalPrinterSample" />
```

3. Make the following changes in the **BuildTools\Customization.settings** package customization configuration file:

- Add the following line to include the CRT extension in the deployable packages.

```
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.Extensions.DocumentProvider.PosnetSample.dll"/>
```

- Add the following line to include the Hardware station extension in the deployable packages.

```
<ISV_HardwareStation_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.PosnetThermalFVFiscalPrinterSample.dll"/>
```

4. Start the MSBuild Command Prompt for Visual Studio utility, and run **msbuild** under the Retail SDK folder to create deployable packages.
5. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).

## Design of extensions

### Commerce runtime extension design

The purpose of the extension that is a fiscal document provider is to generate printer-specific documents and handle responses from the fiscal printer.

The Commerce runtime extension is **Runtime.Extensions.DocumentProvider.PosnetSample**. This extension generates a set of printer-specific commands in JavaScript Object Notation (JSON) format that are defined by POSNET specification 19-3678.

For more details about the design of the fiscal integration solution, see [Fiscal registration process and fiscal integration samples for fiscal devices](#).

#### Request handler

The **DocumentProviderPosnetProtocol** request handler is the entry point for the request to generate documents from the fiscal printer.

The handler is inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the connector document provider name that is specified in Headquarters.

The connector supports the following requests:

- **GetFiscalDocumentDocumentProviderRequest** – This request contains information about what document should be generated. It returns a printer-specific document that should be registered in the fiscal printer.
- **GetSupportedRegistrableEventsDocumentProviderRequest** – This request returns the list of events to subscribe to. Currently, the following events are supported: sales, printing X report, and printing Z report.

#### Configuration

The configuration file is found in the **Configuration** folder of the extension project. The purpose of the file is to enable settings for the document provider to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added:

- VAT rates mapping
- Tender type mapping
- Deposit payment type

### Hardware station extension design

The purpose of the extension that is a fiscal connector is to communicate with the fiscal printer.

The Hardware station extension is **HardwareStation.Extension.PosnetThermalFVFiscalPrinterSample**. This extension calls the functions of the POSNET driver to submit commands that the Commerce runtime extension generates to the fiscal printer. It also handles device errors.

#### Request handler

The **FiscalPrinterHandler** request handler is the entry point for handling the request to the fiscal peripheral device.

The handler is inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the fiscal connector name that is specified in Headquarters.

The connector supports the following requests:

- **SubmitDocumentFiscalDeviceRequest** – This request sends documents to printers and returns the response from the fiscal printer.
- **IsReadyFiscalDeviceRequest** – This request is used for a health check of the device.
- **InitializeFiscalDeviceRequest** – This request is used for printer initialization.

#### **Configuration**

The configuration file is located in the **Configuration** folder of the extension project. The purpose of the file is to enable settings for the connector to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added:

- **Connection string** – This string describes the details of the connection to the device in a format that is supported by the driver. For details, see the POSNET driver documentation.
- **Date and time synchronization** – This setting specifies whether the date and time of the printer must be synced with the connected Hardware station.
- **Device timeout** – The amount of time, in milliseconds, that the driver will wait for a response from the device. For details, see the POSNET driver documentation.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customer information management for Poland

2/18/2021 • 7 minutes to read • [Edit Online](#)

## Introduction

This topic describes how you can handle customer information, such as the customer's value-added tax (VAT) number, in Retail point of sale (POS) for Poland.

You can specify the customer's VAT number when you create or edit a customer master record in POS. You can also specify a VAT number for a sales transaction by copying it from the transaction customer or entering it manually. The customer information can then be printed on both regular and fiscal receipts, and it can be used for invoicing purposes.

### NOTE

This functionality is available in version 10.0.7 and later.

## Setup

You must complete the following configuration to use this functionality:

- Set up a registration type for the VAT number.
- Add the **Add customer information** operation to screen layouts.
- Activate the inquiry for customer information.
- Set up receipt formats.
- Configure channel components.

### Set up a registration type for the VAT number

Before VAT numbers can be specified in POS, you must create an appropriate registration type for the VAT number and link it to the **VAT ID** registration category. For more information about how to work with registration types and registration IDs, see [Registration IDs](#).

### WARNING

If a registration type isn't created or isn't linked to the **VAT ID** registration category, an error will be generated in POS when VAT number is filled in for a customer address.

### Add the Add customer information operation to screen layouts

The **Add customer information** operation can be used to add customer information, such as the VAT number, to a sales transaction. This information can be copied from the customer that is specified for the transaction, or it can be manually entered.

On the **Button grids** page, select the button grid where the operation should appear, and open the Button grid designer. Add a new button, and then, in the **Action** field, select **Add customer information**. For more information about how to work with screen layouts and button grids, see [Screen layouts for the point of sale \(POS\)](#).

### Activate the inquiry for customer information

If the customer information isn't specified for a sales transaction, an inquiry for that information can be

triggered automatically after the transaction is finalized. This approach is an alternative to the **Add customer information** operation.

To activate the inquiry for customer information, set the **Enable inquiry of customer information in sales transactions** option to **Yes** in the **Tax parameters** section on the **Functions** FastTab of the **POS functionality profiles** page.

### Set up receipt formats

You can configure receipt formats so that the customer's VAT number is printed on receipts.

#### NOTE

The default company of the user who creates the receipt setup should be the same legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

On the **Language text** page, on the **POS** tab, add the following record for the label of the custom field for receipt formats. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the following table are just examples. You can change them to meet your requirements. However, the **Text ID** value that you use must be unique, and it must be equal to or more than 900001.

LANGUAGE ID	TEXT ID	TEXT
en-US	900001	VAT number

On the **Custom fields** page, add the following record for the custom field for receipt formats. Note that the **Caption text ID** value must correspond to the **Text ID** value that you specified on the **Language text** page.

NAME	TYPE	CAPTION TEXT ID
FISCALCUSTOMER_VATID_PL	Receipt	900001

In the Receipt format designer, add the custom field to the appropriate receipt section for every receipt format that is required. For more information about how to work with receipt formats, see [Receipt templates and printing](#).

### Configure channel components

To make the functionality that is specific to Poland available, you must configure extensions for channel components. For more information, see the [Deployment guidelines](#) section later in this topic.

## Example scenarios

The following example scenarios show how to work with customer information in POS for Poland.

#### Scenario 1: Make a sale to an anonymous customer

1. Sign in to POS.
2. Add items to the cart.
3. Select **Add customer information**, and then select **Enter manually**.
4. Enter the customer's VAT number, and then select **OK**.
5. Register payments for the transaction, and then finalize the transaction.
6. Verify that the printed receipt contains the customer's VAT number.

#### Scenario 2: Make a sale to a new named customer

1. Sign in to POS.
2. Add items to the cart.
3. Select **Add customer**, and then select **New**.
4. Specify the new customer's attributes.
5. Select **Create a new address**. Then specify the new customer's contact information and an address.
6. In the **VAT number** field, enter the customer's VAT number.
7. Save the customer record and the customer address record, and add the customer to the transaction.
8. Register payments for the transaction, and then finalize the transaction.
9. Because the inquiry for customer information has been activated, but customer information hasn't been added to the transaction, the **Enter customer information** dialog box is opened. Select **Yes**, and then select **Copy from transaction customer**.
10. Verify the customer's VAT number, and then select **OK**.
11. Verify that the printed receipt contains the customer's VAT number.

#### NOTE

If you must specify a different customer for the transaction, you must clear the customer information and then copy it again after the new customer is added.

### Scenario 3: Change the customer information for a sale to a named customer

1. Sign in to POS.
2. Add items to the cart.
3. Select **Add customer**, and then select a customer account to add it to the transaction.
4. Select **Add customer information**, and then select **Copy from transaction customer**.
5. Verify the customer's VAT number, and then select **OK**.
6. Select **Add customer information**, and then select **Clear** to clear the customer information from the transaction.
7. Select **Add customer information**, and then select **Enter manually**.
8. Specify the customer's VAT number, and then select **OK**.
9. Register payments for the transaction, and then finalize the transaction.
10. Verify that the printed receipt contains the customer's VAT number.

## Deployment guidelines

This section provides deployment guidance for enabling customer information management in the localization of Dynamics 365 Commerce for Poland.

#### NOTE

Some steps in these procedures vary, depending on the product version you're using. For more information, see [What's new or changed in Dynamics 365 for Retail](#).

If you want to enable the integration of POS with fiscal printers for Poland, and specifically if you want to print customer VAT numbers on fiscal receipts, you must deploy the [fiscal printer integration sample for Poland](#).

### Update customizations

Follow these steps to update customizations.

- [Retail 10.0.7 and later](#)
- [Retail 10.0.12 and later](#)



If any of your customizations include request handlers for the `SaveCartRequest` or `CreateSalesOrderServiceRequest` requests:

1. Find the request handler for `SaveCartRequest`.
2. Find the line of code that runs the original request handler.
3. Add the following lines before calling the original request handler:

```
using Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdPoland.Services;  
  
...  
  
new TaxRegistrationIdFiscalCustomerService().Execute(request);
```

4. Find the request handler for `CreateSalesOrderServiceRequest`.
5. Find the line of code that runs the original request handler.
6. Replace it with the following code:

```
using Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdPoland.Services;  
  
...  
  
return new TaxRegistrationIdFiscalCustomerService().Execute(request);
```

## Update a development environment

Follow these steps to update a development environment.

### CRT extension components

1. Find the extension configuration file for the Commerce runtime (CRT):
  - **Commerce Scale Unit:** Find the `CommerceRuntime.Ext.config` file in the `bin\ext` folder under the Microsoft Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Find the `CommerceRuntime.MPOSOffline.Ext.config` file under the local CRT client broker location.
2. Register the CRT extension in the extension configuration file.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdPoland" />
```

### WARNING

Do not edit the `CommerceRuntime.config` and `CommerceRuntime.MPOSOffline.config` files. These files aren't intended for any customizations.

### Modern POS extension components

Follow these steps to make the `TaxRegistrationId.PL` extension available.

1. Open the solution at `RetailSdk\POS\ModernPOS.sln`.
2. In `POS.Extensions\extensions.json`, turn on the extension.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/TaxRegistrationId.PL"
    }
  ]
}
```

3. Build the solution.
4. Open Modern POS, and test the functionality.

#### Cloud POS extension components

Follow these steps to make the TaxRegistrationId.PL extension available.

1. Open the solution at RetailSdk\POS\CloudPOS.sln.
2. In POS.Extensions\extensions.json, turn on the extension.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/TaxRegistrationId.PL"
    }
  ]
}
```

3. Build the solution.
4. Open Cloud POS, and test the functionality.

#### Update a production environment

Follow these steps to create deployable packages that contain Commerce components, and to apply the packages in a production environment.

1. In the CommerceRuntime.Ext.config and CommerceRuntime.MPOSOffline.Ext.config configuration files under the RetailSdk\Assets folder, add the following lines to the composition section.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdPoland" />
```

2. Turn on the TaxRegistrationId.PL POS extension.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/TaxRegistrationId.PL"
    }
  ]
}
```

3. Run **msbuild** for the whole Retail software development kit (SDK) to create deployable packages.
4. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Retail SDK packaging](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cash register functionality for France

2/18/2021 • 16 minutes to read • [Edit Online](#)

This topic provides an overview of the cash register functionality that is available for France in Dynamics 365 Commerce. It also provides guidelines for setting up the functionality.

The functionality consists of the following parts:

- Common point-of-sale (POS) features that are available to customers in all countries or regions. Examples include an option to register various events in the POS audit log.
- France-specific features, such as digital signatures for sales transactions.

## Overview

### Common POS features

To learn about POS features that are available to customers in all countries or regions, see [Commerce home page](#).

The following POS localization features that are available to customers in all countries or regions can now be used specifically for France:

- **Register additional events in the POS audit event log.** If the **Audit** option in the POS functionality profile is set to **Yes**, the following events are registered in the POS audit event log:
  - Sign-in
  - Sign-out
  - Printing a copy of a receipt
  - Starting offline mode
  - Ending offline mode
  - Cleanup of transactions from the channel database

### France-specific POS features

The following France-specific POS features are enabled when the **ISO code** field in the POS functionality profile is set to **FR**.

#### Digital signing overview

The following types of data (transactions and events) are digitally signed in POS:

- Sales transactions
- Copies of receipts
- Closed shift/Z reports
- Audit events

The signature is created and recorded in the channel database at the same time that the transaction is finalized or the event is registered. The data that is signed is a text string that consists of several data fields. These fields vary, depending on the type of data. The general signing process consists of the following steps:

1. Based on the type of data, select the next sequential number for signing purposes.
2. Extract the data fields that must be signed from the record that is being signed.
3. Build a string that consists of a comma-separated list of the data fields.
4. Add the previous signature for the same type of data.

5. Calculate a hash code of the string by using the SHA-x algorithm.
6. Encrypt the resulting string by using a digital certificate.
7. Do the base64url transformation for the resulting string.
8. Store the string that is used for signing, the sequential number, the signature, and the thumbprint of the certificate in a fiscal response record that is linked to the transaction or event.
9. Transfer the fiscal response to the enterprise resource planning (ERP) system in Headquarters, together with the transaction or event.

#### **Digital signing of sales transactions**

Only transactions for cash sales are signed. Here are some examples of transactions that are excluded from the signing process:

- Prepayments (customer account deposits)
- Quotations
- Prepayments for sales orders (customer order deposits)
- Issuing a gift card and adding funds to a gift card
- Non-sales transactions (float entry, tender removal, and so on)

The data that is signed for a sales transaction is a text string that consists of the following data fields:

1. The total amount of sales lines. The amount includes tax per tax rate.
2. The total amount of sales. The amount includes tax.
3. The date and time of the transaction, in the format YYYYMMDDHHMMSS.
4. The register number.
5. The sequential number of the signed sales transaction.
6. The type of sales transaction.
7. A value (Y/N) that indicates whether the transaction is the first signed sales transaction for the register.
8. The previous signature for the same register. A blank value is used for the first signed sales transaction.

You can view the transaction signature, together with the transaction data that was used to generate it, on the **Fiscal transactions** FastTab of the **Store transactions** page in Headquarters.

#### **Digital signing of receipt copies**

When a copy of a receipt is printed, the event is registered in the POS audit event log. Only copies of receipts for signed sales transactions are signed. The data that is signed for a receipt copy event is a text string that consists of the following data fields:

1. The receipt number of the original sales transaction.
2. The type of transaction for the original sales transaction.
3. The number of the receipt copy for this sales transaction.
4. The staff ID of the operator who is printing the receipt copy.
5. The date and time of the receipt copy event, in the format YYYYMMDDHHMMSS.
6. The sequential number of the signed receipt copy event.
7. A value (Y/N) that indicates whether the transaction is the first signed receipt copy event for the register.
8. The previous signature for the same register. A blank value is used for the first signed receipt copy event.

You can view the signature of the receipt copy, together with the event data that was used to generate it, on the **Signature** tab of the **Audit events** page in Headquarters.

#### **Digital signing of closed shifts**

When a shift is closed, the event is registered in the POS audit event log. The data that is signed for a shift closing event is a text string that consists of the following data fields:

1. The total amount of sales. The amount includes tax per tax rate.

2. The total amount of sales. The amount includes tax.
3. The date and time of the shift closing event, in the format YYYYMMDDHHMMSS.
4. The sequential number of the shift closing event.
5. A value (Y/N) that indicates whether the transaction is the first signed shift closing event for the register.
6. The previous signature for the same register. A blank value is used for the first signed shift closing event.

You can view the signature of a closed shift, together with the shift data that was used to generate it, on the **Signature** tab of the **Shifts** page in Headquarters.

#### **Digital signing of events**

The data that is signed for an event other than a receipt copy or shift closing event is a text string that consists of the following data fields:

1. The sequential number of the signed event.
2. A predefined event code.
3. A description of the event.
4. The date and time of the event.
5. The staff ID of the operator who raised the event.
6. The register number.
7. A value (Y/N) that indicates whether the transaction is the first signed event for the register.
8. The previous signature for the same register. A blank value is used for the first signed event.

You can view the event signature, together with the event data that was used to generate it, on the **Signature** tab of the **Audit events** page in Headquarters.

#### **Receipts**

Receipts for France can include additional information that was implemented by using custom fields:

- **Transaction type** – You can add a field to a receipt format layout to identify the type of transaction. For example, a sales receipt will include the text "Sales."
- **Sequential number of signed sales transaction** – A receipt can include the sequential number of a signed sales transaction. This number is used to associate the printed receipt with a digital signature in the database.
- **Extract from digital signature** – A receipt can include an extract from the digital signature. This extract is used to confirm that the transaction is signed. It consists of a concatenation of the third, seventh, thirteenth, and nineteenth symbols of the signature.
- **Reprint number** – An original receipt or a receipt copy can include the number of the receipt copy. For an original receipt, the value is **0** (zero).
- **Line count** – A receipt can include the number of printed item lines on the receipt.
- **Sales totals** – Custom fields for receipt totals exclude non-sales amounts from the total transaction amounts. Non-sales amounts include amounts for the following operations:
  - Prepayments (customer account deposits)
  - Prepayments for sales orders (customer order deposits)
  - Issuing a gift card
  - Adding funds to a gift card
- **Certification data** – A receipt can include the category and number of the certificate of compliance that an authorized body issued per the NF 525 certification requirements.
- **Build number** – A receipt can include the POS build number.

### Restricting the duration of shifts

There is an option to enforce daily shift closing in POS. A shift can't last longer than the time that is specified in the **Shift closing time** field. Several minutes before that time, the operator will start to receive warnings that the shift must be closed. The number of minutes is determined by the value of the **Shift closing interval (minutes)** field.

### X and Z reports

The information that is included on X and Z reports is based on French requirements:

- **Total sales** for the shift. This information includes amounts only for cash sales transactions. Prepayments and operations for issuing a gift card are excluded.
- **Total returns** for the shift.
- **Cumulative grand total**. This amount is calculated as the cumulative grand total amount of the previous shift plus the total sales amount of this shift minus the absolute value of the total returns amount of this shift.
- **Cumulative perpetual grand total**. This amount is calculated as the cumulative perpetual grand total amount of the previous shift plus the total sales amount of this shift plus the absolute value of the total returns amount of this shift.
- Value-added tax (VAT) amounts per tax rate.

The totals are also stored in the closed shift record and transferred to Headquarters.

### Period grand total journal

Period grand total journals summarize sales totals per store and fiscal period.

Period grand total journals are maintained on the **Period grand total journal** page. To create a new journal, you must specify a store. If previous journals exist for the store, the next fiscal period after the last closed journal for the store is automatically used as the new journal period. If previous journals do not exist, you can specify the end date of the journal. In this case, the fiscal period that includes the specified date is used as the journal period.

The journal can then be calculated. Shifts that were closed during the journal period are selected, and totals are calculated for those shifts. You can view the journal's tax totals per sales tax code. You can also view the shifts that are included in the journal.

After the journal is calculated, it can be closed. A closed journal can't be modified, and another journal can't be created for a previous period, the same period, or an intersecting period. However, the last closed journal for a store can be canceled. In that case, another journal can be created for the same store and period.

A closed journal is digitally signed. You can view the journal signature, together with the journal data that was used to generate it, on the **Signature details** tab of the **Period grand total journal** page in Headquarters.

### Archive

An archive is an XML file that can be exported from a Period grand total journal that has been closed. It includes the totals for the closed period, and also includes detailed data about sales transactions and events. The exported file is digitally signed, and the signature is contained in a separate file.

The archive format is implemented by using [Electronic reporting \(ER\)](#).

## Setting up Commerce for France

This section describes the Commerce settings that are specific to and recommended for France. For more information, see [Commerce home page](#).

To use the France-specific functionality, you must complete these tasks:

- Set the **Country/region** field to **FRA** (France) in the primary address of the legal entity.
- Set the **ISO code** field to **FR** (France) in the POS functionality profile of every store that is located in France.

You must also specify the following settings for France. Note that you must run appropriate distribution jobs after you complete the setup.

### Set up the legal entity

You must make the following changes on the **Legal entities** page. These settings are used in the archive format.

- On the **Bank account information** FastTab, in the **Routing number** field, specify the VAT identifier of the organization.
- On the **Statutory reporting** FastTab, in the **NAF code** field, specify the Nomenclature des Activités Françaises (NAF) code of the organization.
- On the **Tax registration** FastTab, in the **Tax registration number** field, specify the Système d'identification du répertoire des établissements (SIRET) number of the organization.

### Set up VAT per French requirements

You must create sales tax codes, sales tax groups, and item sales tax groups. You must also set up sales tax information for products and services. For more information about how to set up and use sales tax, see [Sales tax overview](#).

You must also specify sales tax groups and enable the **Prices include sales tax** option for stores that are located in France.

### Set up functionality profiles

You must enable auditing by setting the **Audit** option to **Yes**.

To enforce daily shift closing, you must make the following changes:

- Set the **Enforce daily shift closing** option to **Yes**.
- Set the **Shift closing time** and **Shift closing interval (minutes)** fields.

### Configure custom fields so that they can be used in receipt formats for sales receipts

You can configure the language text and custom fields that are used in the POS receipt formats. The default company of the user who creates the receipt setup should be the same legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

On the **Language text** page, add the following records for the labels of the custom fields for receipt layouts. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the table are just examples. You can change them to meet to your requirements. However, the **Text ID** values that you use must be unique, and they must be equal to or higher than 900001.

LANGUAGE ID	TEXT ID	TEXT ID
en-US	900001	Transaction type
en-US	900002	Sequential number
en-US	900003	Digital signature
en-US	900004	Reprint number
en-US	900005	Sales tax amount
en-US	900006	Sales total



LANGUAGE ID	TEXT ID	TEXT ID
en-US	900007	Sales total tax
en-US	900008	Sales total including tax
en-US	900009	Build number
en-US	900010	Certification category
en-US	900011	Certificate number
en-US	900012	Line count

On the **Custom fields** page, add the following records for the custom fields for receipt layouts. Note that **Caption text ID** values must correspond to the **Text ID** values that you specified on the **Language text** page.

NAME	TYPE	CAPTION TEXT ID
TRANSACTIONTYPE	Receipt	900001
SEQUENTIALNUMBER	Receipt	900002
DIGITALSIGNATURE	Receipt	900003
REPRINTNUMBER	Receipt	900004
SALESTAXAMOUNT	Receipt	900005
SALESTOTAL	Receipt	900006
SALESTOTALTAX	Receipt	900007
SALESTOTALINCLUDETAX	Receipt	900008
BUILDNUMBER	Receipt	900009
CERTIFICATIONCATEGORY	Receipt	900010
CERTIFICATENUMBER	Receipt	900011
LINECOUNT	Receipt	900012

### Configure receipt formats

For every required receipt format, change the value of the **Print behavior** field to **Always print**.

In the Receipt format designer, add the following custom fields to the appropriate receipt sections. Note that field names correspond to the language texts that you defined in the previous section.

- **Header:** Add the following field:
  - **Transaction type** – This field identifies the type of receipt.
- **Lines:** We recommend that you add the following standard fields:

- **Unit price with tax**
- **Total price with tax**
- **Tax ID**
- **Footer:** Add the following fields:
  - **Sales total** – This field prints the receipt's total cash sale amount. The amount excludes tax. Prepayments and gift card operations are excluded.
  - **Sales total tax** – This field prints the receipt's total tax amount for cash sales. Prepayments and gift card operations are excluded.
  - **Sales total including tax** – This field prints the receipt's total cash sale amount. The amount includes tax. Prepayments and gift card operations are excluded.
  - **Tax ID** – This standard field enables a sales tax summary to be printed per sales tax code. The field must be added to a new line.
  - **Sales tax amount** – This field prints the receipt's tax amount for cash sales per sales tax code. Prepayments and gift card operations are excluded. The field must be added to the same line as the **Tax ID** field.
  - **Sequential number** – This field prints the sequential number of a signed sales transaction.
  - **Digital signature** – This field prints the extract from the digital signature.
  - **Reprint number** – This field prints the number of a receipt copy.
  - **Build number** – This field prints the POS build number.
  - **Certification category** – This field prints the category of the certificate of compliance that an authorized body issued per the NF 525 certification requirements.
  - **Certificate number** – This field prints the number of the certificate of compliance that an authorized body issued per the NF 525 certification requirements.
  - **Line count** – This field prints the number of printed item lines on a receipt.
  - **Text** – Add a text field, and specify the VAT identifier of the organization.
  - **Text** – Add a text field, and specify the NAF code of the organization.
  - **Text** – Add a text field, and specify the SIRET number of the organization.
  - **Store name** – This standard field prints the name of the store.
  - **Store address** – This standard field prints the address of the store.

For more information about how to work with receipt formats, see [Set up and design receipt formats](#).

### **Configure the digital signature parameters for Headquarters**

To digitally sign Period grand total journals and archives, you must set up digital signature parameters. The signing is done by using a digital certificate that is stored in Microsoft Azure Key Vault storage. The following steps must be completed before you can use a certificate that is stored in Key Vault storage:

- The Key Vault storage must be created. We recommend that you deploy the storage in the same geographical region as the Commerce Scale Unit.
- The certificate must be uploaded to the Key Vault storage.
- The Application Object Server (AOS) application must be authorized to read secrets from the Key Vault storage.

For more information about how to work with Key Vault, see [Get started with Azure Key Vault](#).

Then, on the **Key Vault parameters** page, you must specify the parameters for accessing the Key Vault storage:

- **Name and Description** – The name and description of the Key Vault storage.
- **Key Vault URL** – The URL of the Key Vault storage.
- **Key Vault client** – An interactive client ID of the Azure Active Directory (Azure AD) application that is associated with the Key Vault storage for authentication purposes. This client should have access to read

secrets from the storage.

- **Key Vault secret key** – A secret key that is associated with the Azure AD application that is used for authentication in the Key Vault storage.
- **Name, Description, and Secret reference** – The name, description, and secret reference of the certificate.

Finally, on the **Commerce parameters** page, you must specify the parameters for digital signatures:

- **Certificate** – Select the certificate that you configured in the previous step.
- **Hash function** – Specify one of the cryptographic hash algorithms that are supported by Microsoft .NET, such as **SHA1**.
- **Encoding** – Specify the encoding of the signed data, such as **UTF-8**.

### Configure the archive export format

You can download the ER configuration for the archive from Microsoft Dynamics Lifecycle Services (LCS). For more information, see [Import electronic reporting configurations](#). You must download the following versions, or later versions, of the configurations:

- **Retail channel data.version.2** data model
- **Archiving DMM.version.2.3** data model mapping
- **Retail data archive FR .version.2.5** format

After you import the configurations, on the **Commerce parameters** page, on the **Electronic documents** tab, in the **Retail data archive export format** field, select the **Retail data archive FR .version.2.5** format.

### Renitalize Commerce components

#### NOTE

You only need to complete the steps of this section if you are updating an existing environment.

To enable audit events, you must reinitialize the Commerce Extensible enumerations. To enable transmitting France-specific data from POS to HQ, you must reinitialize the Commerce Scheduler.

On the **General** FastTab of the **Commerce parameters** page, click **Initialize**. For more information, see [Initialize seed data in new Retail environments](#)

There is an option to separately configure the scheduler. Click **Commerce scheduler** > **Initialize commerce scheduler**. On the **Initialize Commerce scheduler** page, click **OK**.

### Configure channel components

To enable France-specific functionality, you must configure extensions for channel components. For more information, see the [deployment guidelines](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Deployment guidelines for cash registers for France

2/18/2021 • 20 minutes to read • [Edit Online](#)

This topic is a deployment guide that shows how to enable the Dynamics 365 Commerce localization for France. The localization consists of several extensions of components. For example, the extensions let you print custom fields on receipts, register additional audit events, sales transactions, and payment transactions in Point of Sale (POS), digitally sign sales transactions, and print X and Z reports in local formats. For more information about the localization for France, see [Cash register functionality for France](#).

This localization is part of the Retail software development kit (SDK). For information about how to install and use the SDK, see the [Retail software development kit \(SDK\) architecture](#).

This localization consists of extensions for the Commerce runtime (CRT), Retail Server, and POS. To run this sample, you must modify and build the CRT, Retail Server, and POS projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Microsoft Visual Studio Online (VSO), where no files have been changed yet.

## NOTE

In Commerce 10.0.8 and above, Retail Server is known as Commerce Scale Unit. Because this topic applies to multiple previous versions of the app, *Retail Server* is used throughout the topic.

## Storing a certificate for digital signing in Azure Key Vault

The digital signature extension uses a certificate that is installed in the local certificate storage of the machine where Retail Server is deployed. The thumbprint of the certificate must be specified in the configuration file (see the [SequentialSignatureRegister component](#) section later in this topic). Depending on the implementation topology, the certificate might have to be stored in [Microsoft Azure Key Vault storage](#). The localization for France contains a code sample that shows how to override the signing flow and sign sales transactions by using a certificate that is stored in Azure Key Vault storage.

### Prerequisites

The following steps must be completed before you can use a certificate that is stored in Azure Key Vault storage:

- The Azure Key Vault storage must be created. We recommend that you deploy the storage in the same geographical region as the Retail Server.
- The certificate must be uploaded to the storage.
- The Retail Server application must be authorized to read secrets from the storage.

For more information about how to work with Azure Key Vault, see [Get started with Azure Key Vault](#).

### Using the sample

The `DigitalSignatureKeyVaultSample` project contains sample code that uses a certificate that is stored in Azure Key Vault storage. To use the sample in a production environment, you must implement logic so that the following parameters can be specified in the `HashAndSignData` method of the `CertificateSignatureServiceRequestHandler` class:

- **Azure Key Vault URL** – The URL of the Azure Key Vault storage.

```
settings.Add(WellKnownKeyVaultSettings.KeyVaultUrl, "Set your Azure Key Vault URL here");
```

- **Client ID** – An interactive client ID of the Azure Active Directory (Azure AD) application that is associated with the Azure Key Vault storage for authentication purposes. This client should have access to read secrets from the Azure Key Vault storage.

```
settings.Add(WellKnownKeyVaultSettings.KeyVaultInteractiveClientId, "Set the client ID here");
```

- **Client secret** – A secret key that is associated with the Azure AD application that is used for authentication in the Azure Key Vault storage.

```
// Secret key value should be encrypted and stored in a safe place.
settings.Add(WellKnownKeyVaultSettings.KeyVaultClientSecretKey, "Set the secret key here");
```

- **Secret reference** – A secret reference to the certificate.

```
SecretCertificate secretCertificate = settingsHelper.SecretProvider.GetSecret("vault:///{{Specify the secret reference}}") as SecretCertificate;
```

To override the signing flow, follow these steps.

1. Build the **DigitalSignatureKeyVaultSample** project, and copy the **Contoso.Commerce.Runtime.DigitalSignatureKeyVaultSample.dll** assembly to the **bin\ext Retail Server** folder.
2. Update the **commerceRuntime.ext.config** file by adding the following line to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DataSignatureKeyVaultSample" />
```

#### NOTE

The thumbprint of the certificate that is used for digital signing should be specified in the configuration file of the **SequentialSignatureRegister** assembly, even if the certificate is stored in Azure Key Vault storage. For more information, see the [SequentialSignatureRegister component](#) section later in this topic.

### Using certificate profiles in Commerce channels

In Commerce version 10.0.15 and later, you can use the [User-defined certificate profiles for retail stores](#) feature that supports failover to offline when Key Vault or Headquarters are not available. The feature extends the [Manage secrets for retail channels](#) feature.

To apply the new feature in the CRT extension, follow these steps.

1. Create a new CRT extension project (C# class library project type). Use the sample templates from the Retail software development kit (SDK) (RetailSDK\SampleExtensions\CommerceRuntime).
2. Add custom handler for **CertificateSignatureServiceRequest** in the **SequentialSignatureRegister** project.
3. To read a secret call, **GetUserDefinedSecretCertificateServiceRequest** using a constructor with **profileId** parameter. That will start the functionality working with settings from Certificate profiles. Based on the settings, the certificate will be retrieved either from Azure Key Vault or local machine storage.

```
GetUserDefinedSecretCertificateServiceRequest getUserDefinedSecretCertificateServiceRequest = new
GetUserDefinedSecretCertificateServiceRequest(profileId: "ProfileId", secretName: null, thumbprint: null,
expirationInterval: null); GetUserDefinedSecretCertificateServiceResponse
```

```
getUserDefinedSecretCertificateServiceResponse =  
request.RequestContext.Execute(getUserDefinedSecretCertificateServiceRequest);  
  
X509Certificate2 Certificate = getUserDefinedSecretCertificateServiceResponse.Certificate;
```

4. When the certificate is retrieved, proceed with data signing.
5. Build the CRT extension project.
6. Copy the output class library and paste it into ...RetailServer\webroot\bin\Ext for manual testing.
7. In the CommerceRuntime.Ext.config file, update the extension composition section with the custom library information.

## Specifying application attributes that will be printed on receipts

You can use custom fields to print the following application attributes on receipts:

- **Build number** – The software version of the POS application. By default, the value should equal the POS build number that Microsoft assigned to the POS application.
- **Certificate category** and **Certificate number** – The category and number of the certificate of compliance that an accredited body issues for the application. By default, the values equal the category and the number of the certificate that is granted to Microsoft:
  - Microsoft Dynamics 365 for Commerce:
    - **Certificate category:** C
    - **Certificate number:** 18/0202
  - Microsoft Dynamics 365 for Commerce:
    - **Certificate category:** B
    - **Certificate number:** 18/0203

### NOTE

By default, the certificate category and number that are assigned are printed. If you're implementing Commerce, you must override the certificate category and number.

If you customize the POS application, and your customizations affect the compliance of the application, you might have to request a new certificate of compliance from an accredited body. In this case, you must override the build number, and the certificate category and number. Otherwise, the default values for the certificate category and number will be printed, but you must still specify the POS build number that Microsoft assigned to the POS application.

### Overriding the build number

The software version/build number and publisher are specified in `RetailSDK\BuildTools\Customization.settings`.

```
<CustomVersion Condition="'$(CustomVersion)' == ''>1.0.0.1</CustomVersion>  
<CustomName Condition="'$(CustomName)' == ''>Contoso Retail Customization</CustomName>  
<CustomDescription Condition="'$(CustomDescription)' == ''>Contoso Retail Customization</CustomDescription>  
<CustomPublisher Condition="'$(CustomPublisher)' == ''>CN=Contoso Ltd.</CustomPublisher>  
<CustomPublisherDisplayName Condition="'$(CustomPublisherDisplayName)' == ''>Contoso Ltd.  
</CustomPublisherDisplayName>  
<CustomCopyright Condition="'$(CustomCopyright)' == ''>Copyright © 2016</CustomCopyright>
```

### Overriding the certificate category and number

The certificate category and number are specified in

`RetailSDK\SampleExtensions\CommerceRuntime\Extensions.ReceiptsFrance\GetSalesTransactionCustomReceiptFieldService`.

```
/// <summary>
/// Certification category.
/// </summary>
private const string CertificationCategory = "C";

/// <summary>
/// Certificate number.
/// </summary>
private const string CertificateNumber = "18/0202";
```

#### NOTE

You must also override the certificate category and number if you're implementing Commerce. In this case, use the certificate category and number that are provided in the [Specifying application attributes that will be printed on receipts](#) section earlier in this topic.

## Development environment

Follow these steps to set up a development environment so that you can test and extend the localization functionality.

### CRT extension components

The CRT extension components are included in the CRT samples. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`, under `RetailSdk\SampleExtensions\CommerceRuntime`.

#### CommonFrance component

1. Find the `Runtime.Extensions.CommonFrance` project, and build it.
2. In the `Extensions.CommonFrance\bin\Debug` folder, find the `Contoso.Commerce.Runtime.CommonFrance.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Retail Server:** Copy the assembly to the `\bin\ext` folder under the Microsoft Internet Information Services (IIS) Retail Server site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.CommonFrance" />
```

#### ReceiptsFrance component

1. Find the `Runtime.Extensions.ReceiptsFrance` project, and build it.
2. In the `Extensions.ReceiptsFrance\bin\Debug` folder, find the

**Contoso.Commerce.Runtime.ReceiptsFrance.dll** assembly file.

3. Copy the assembly file to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

4. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.ReceiptsFrance" />
```

#### **SalesPaymentTransExt component**

1. Find the `Runtime.Extensions.SalesPaymentTransExt` project, and build it.

2. In the `Extensions.SalesPaymentTransExt\bin\Debug` folder, find the `Contoso.Commerce.Runtime.SalesPaymentTransExt.dll` assembly file.

3. Copy the assembly file to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

4. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SalesPaymentTransExt" />
```

#### **SalesPaymentTransExtFrance component**

1. Find the `Runtime.Extensions.SalesPaymentTransExtFrance` project, and build it.

2. In the `Extensions.SalesPaymentTransExtFrance\bin\Debug` folder, find the `Contoso.Commerce.Runtime.SalesPaymentTransExtFrance.dll` assembly file.

3. Copy the assembly file to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

4. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and



it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SalesPaymentTransExtFrance" />
```

#### SequentialSignatureFrance component

1. Find the `Runtime.Extensions.SequentialSignatureFrance` project, and build it.
2. In the `Extensions.SequentialSignatureFrance\bin\Debug` folder, find the `Contoso.Commerce.Runtime.SequentialSignatureFrance.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Retail Server:** The file is named `commercerruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SequentialSignatureFrance" />
```

#### SequentialSignatureRegister component

1. Find the `Runtime.Extensions.SequentialSignatureRegister` project.
2. Modify the `App.config` file by specifying the thumbprint, store location, and store name for the certificate that should be used to sign sales transactions. The `certificateThumbprint` property is the only mandatory property. The value must be a string that is 40 characters long in upper case and that doesn't include any delimiters. For more information, see [How to retrieve the thumbprint of a certificate](#).

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="SequentialSignatureRegister"
type="Contoso.Commerce.Runtime.SequentialSignatureRegister.Configuration.SequentialSignatureRegisterC
onfigSection, Contoso.Commerce.Runtime.SequentialSignatureRegister"/>
  </configSections>
  <SequentialSignatureRegister certificateThumbprint="insert key certificateThumbprint here"
certificateStoreLocation="LocalMachine" certificateStoreName="My"/>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.1"/>
  </startup>
</configuration>
```

3. Build the project.
4. In the `Extensions.SequentialSignatureRegister\bin\Debug` folder, find the following files:
  - The `Contoso.Commerce.Runtime.SequentialSignatureRegister.dll` assembly file
  - The `Contoso.Commerce.Runtime.SequentialSignatureRegister.dll.config` configuration file
5. Copy the files to the CRT extension folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

6. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOffline.Ext.config`, and it's under the local CRT client broker location.

7. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SequentialSignatureRegister" />
```

#### **SequentialSignatureRegister.Contracts component**

1. Find the `Runtime.Extensions.SequentialSignatureRegister.Contracts` project, and build it.

2. In the `Extensions.SequentialSignatureRegister.Contracts\bin\Debug` folder, find the `Contoso.Commerce.Runtime.SequentialSignatureRegister.Contracts.dll` assembly file.

3. Copy the assembly file to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

#### **XZReportsFrance component**

1. Find the `Runtime.Extensions.XZReportsFrance` project, and build it.

2. In the `Extensions.XZReportsFrance\bin\Debug` folder, find the `Contoso.Commerce.Runtime.XZReportsFrance.dll` assembly file.

3. Copy the assembly file to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

4. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOffline.Ext.config`, and it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.XZReportsFrance" />
```

#### **RestrictingShiftDuration component**

- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

1. Find the `Runtime.Extensions.RestrictingShiftDuration` project, and build it.

2. In the `Extensions.RestrictingShiftDuration\bin\Debug` folder, find the

Contoso.Commerce.Runtime.RestrictingShiftDuration.dll assembly file.

3. Copy the assembly file to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

4. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commercerruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.RestrictingShiftDuration" />
```

### Retail Server extension components

#### SalesTransactionSignature Retail Server sample component

1. In the

`RetailSDK\SampleExtensions\RetailServer\RetailServer.Extensions.SalesTransactionSignatureSample` folder, find the `RetailServer.Extensions.SalesTransactionSignatureSample` project, and build it.

2. In the `RetailServer\Extensions.SalesTransactionSignatureSample\bin\Debug` folder, find the `Contoso.RetailServer.SalesTransactionSignatureSample.dll` assembly file.

3. Copy the assembly file to the `\bin\ext` folder under the IIS Retail Server site location.

4. Find the configuration file for Retail Server. The file is named `web.config`, and it's in the root folder under the IIS Retail Server site location.

5. Register the Retail Server extensions in the `extensionComposition` section of the configuration file.

```
<add source="assembly" value="Contoso.RetailServer.SalesTransactionSignatureSample" />
```

### Proxy extension component

You must complete the following procedure to enable the extensions in offline mode for Modern POS.

#### SalesTransactionSignature Retail proxy sample component

1. In the

`RetailSDK\SampleExtensions\RetailProxy\RetailProxy.Extensions.SalesTransactionSignatureSample` folder, find the `RetailServer.Extensions.SalesTransactionSignatureSample` project, and build it.

2. In the `RetailProxy\RetailProxy.Extensions.SalesTransactionSignatureSample\bin\Debug` folder, find the `Contoso.Commerce.RetailProxy.SalesTransactionSignatureSample` assembly file.

3. Copy the assembly files to the `\ext` folder under the local CRT client broker location.

4. Register the proxy change in the extensions configuration file. The file is named `RetailProxy.MPOSOOffline.ext.config`, and it's under the local CRT client broker location.

```
<add source="assembly" value="Contoso.Commerce.RetailProxy.SalesTransactionSignatureSample" />
```

## Modern POS extension components

1. Open the solution at `RetailSdk\POS\ModernPOS.sln`, and make sure that it can be compiled without errors. Additionally, make sure that you can run Modern POS from Microsoft Visual Studio by using the **Run** command.

### NOTE

Modern POS must not be customized. You must enable User Account Control (UAC), and you must uninstall previously installed instances of Modern POS as required.

2. Include the following existing source code folders in the `Pos.Extensions` project.

- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)
- SalesTransactionSignatureSample
- SequentialSignature
- AuditEventSignatureSample
- RestrictingShiftDuration
- SalesTransBuildNumberSample

### NOTE

To view all files in the project folder, not just the files that are included in the project, select the **Show All Files** button in Solution Explorer. If this button isn't available, make sure that you selected the project. The icons of files and folders that aren't currently part of the project have a dotted outline. Right-click the folder to include in the project, and then select **Include in Project**.

3. Enable the extensions to be compiled by removing the following folders from the exclude list in `tsconfig.json`:

- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)
- SalesTransactionSignatureSample
- SequentialSignature
- AuditEventSignatureSample
- RestrictingShiftDuration
- SalesTransBuildNumberSample

4. Enable the extensions to be loaded by adding the following lines in `extensions.json`:

- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```
{
  "baseUrl": "SalesTransactionSignatureSample"
},
{
  "baseUrl": "SequentialSignature"
},
{
  "baseUrl": "AuditEventSignatureSample"
},
{
  "baseUrl": "RestrictingShiftDuration"
},
{
  "baseUrl": "SalesTransBuildNumberSample"
}
```

#### NOTE

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the readme.md file in the **Pos.Extensions** project.

5. Rebuild the solution.
6. Run Modern POS in the debugger, and test the functionality.

#### Cloud POS extension components

1. Open the solution at **RetailSdk\POS\CloudPOS.sln**, and make sure that it can be compiled without errors.
2. Include the following existing source code folders in the **Pos.Extensions** project:
  - [Retail 7.3.2 and later](#)
  - [Retail 7.3.5 and later](#)
  - [Retail 8.1.1 and later](#)
  - SalesTransactionSignatureSample
  - SequentialSignature
  - AuditEventSignatureSample
  - RestrictingShiftDuration
  - SalesTransBuildNumberSample

#### NOTE

To view all files in the project folder, not just the files that are included in the project, select the **Show All Files** button in Solution Explorer. If this button isn't available, make sure that you selected the project. The icons of files and folders that aren't currently part of the project have a dotted outline. Right-click the folder to include in the project, and then select **Include in Project**.

3. Enable the extensions to be compiled by removing the following folders from the exclude list in **tsconfig.json**:
  - [Retail 7.3.2 and later](#)
  - [Retail 7.3.5 and later](#)
  - [Retail 8.1.1 and later](#)
  - SalesTransactionSignatureSample

- SequentialSignature
- AuditEventSignatureSample
- RestrictingShiftDuration
- SalesTransBuildNumberSample

4. Enable the extensions to be loaded by adding the following lines in `extensions.json`:

- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```
{
  "baseUrl": "SalesTransactionSignatureSample"
},
{
  "baseUrl": "SequentialSignature"
},
{
  "baseUrl": "AuditEventSignatureSample"
},
{
  "baseUrl": "RestrictingShiftDuration"
},
{
  "baseUrl": "SalesTransBuildNumberSample"
}
```

#### NOTE

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the `readme.md` file in the `Pos.Extensions` project.

5. Rebuild the solution.

6. Run the solution by using the `Run` command and following the steps in the Retail SDK handbook.

7. Test the functionality.

#### Set up required parameters in Headquarters

For more information, see [Cash register functionality for France](#).

## Production environment

Follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Complete the steps in the [Cloud POS extension components](#) or [Modern POS extension components](#) section earlier in this topic.
2. Make the following changes in the package configuration files under the `RetailSdk\Assets` folder:
  - a. In the `commerceruntime.ext.config` and `CommerceRuntime.MPOSOOffline.Ext.config` configuration files, add the following lines to the `composition` section:
    - [Retail 7.3.2 and later](#)
    - [Retail 7.3.5 and later](#)
    - [Retail 8.1.1 and later](#)

```
<add source="assembly" value="Contoso.Commerce.Runtime.CommonFrance" />
<add source="assembly" value="Contoso.Commerce.Runtime.ReceiptsFrance" />
<add source="assembly" value="Contoso.Commerce.Runtime.RestrictingShiftDuration" />
<add source="assembly" value="Contoso.Commerce.Runtime.SalesPaymentTransExt" />
<add source="assembly" value="Contoso.Commerce.Runtime.SalesPaymentTransExtFrance" />
<add source="assembly" value="Contoso.Commerce.Runtime.SequentialSignatureFrance" />
<add source="assembly" value="Contoso.Commerce.Runtime.SequentialSignatureRegister" />
<add source="assembly" value="Contoso.Commerce.Runtime.XZReportsFrance" />
```

To use a certificate that is stored in Azure Key Vault storage for digital signing, add the following line.

#### NOTE

Before you add this line, complete the steps in the [Storing a certificate for digital signing in Azure Key Vault](#) section earlier in this topic.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DataSignatureKeyVaultSample" />
```

- b. In the `RetailProxy.MPOSOOffline.ext.config` configuration file, add the following lines to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.RetailProxy.SalesTransactionSignatureSample" />
```

3. Make the following changes in the **Customization.settings** package customization configuration file:

- a. Add the following lines to the **ItemGroup** section to include the Commerce proxy extension in the deployable packages.

```
<ISV_RetailProxy_CustomizableFile
  Include="$(SdkReferencesPath)\Contoso.Commerce.RetailProxy.SalesTransactionSignatureSample.dll
" />
```

- b. Add the following lines to the **ItemGroup** section to include the CRT extensions in the deployable packages:

- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```

<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.CommonFrance.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.ReceiptsFrance.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.RestrictingShiftDuration.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SalesPaymentTransExt.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SalesPaymentTransExtFrance.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SequentialSignatureFrance.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SequentialSignatureRegister.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SequentialSignatureRegister.dll.config"
/>
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SequentialSignatureRegister.Contracts.d
ll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.XZReportsFrance.dll" />

```

To use a certificate that is stored in Azure Key Vault storage for digital signing, add the following line.

#### NOTE

Before you add this line, complete the steps in the [Storing a certificate for digital signing in Azure Key Vault](#) section, earlier in this topic.

```

<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DataSignatureKeyVaultSample.dll" />

```

- c. Add the following lines to the **ItemGroup** section to include the Retail Server extension in the deployable packages.

```

<ISV_RetailServer_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.RetailServer.SalesTransactionSignatureSample.dll" />

```

4. Update the Retail Server configuration file. In **RetailSDK\Packages\RetailServer\Code\web.config**, add the following lines to the **extensionComposition** section.

```

<add source="assembly" value="Contoso.RetailServer.SalesTransactionSignatureSample" />

```

5. Modify the certificate's configuration file by specifying the thumbprint, store location, and store name for the certificate that should be used to sign sales transactions. Then copy the configuration file to the **References** folder. The file is named **Contoso.Commerce.Runtime.SequentialSignatureRegister.dll.config**, and it's under **Extensions.SequentialSignatureRegister\bin\Debug**.
6. Override the build number and the category and number of the certificate of compliance, as required. For more information, see the instructions in the [Specifying application attributes that will be printed on receipts](#) section earlier in this topic.
7. Start the MSBuild Command Prompt for Visual Studio utility, and run **msbuild** under the Retail SDK



folder to create deployable packages.

8. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).

### **Enable the digital signature in offline mode for Modern POS**

To enable the digital signature in offline mode for Modern POS, you must follow these steps after you activate Modern POS on a new device.

1. Sign in to POS.
2. On the **Database connection status** page, make sure that the offline database is fully synchronized. When the value of the **Pending downloads** field is **0** (zero), the database is fully synchronized.
3. Sign out of POS.
4. Wait a while for the offline database to be fully synchronized.
5. Sign in to POS.
6. On the **Database connection status** page, make sure that the offline database is fully synchronized. When the value of the **Pending transactions in offline database** field is **0** (zero), the database is fully synchronized.
7. Restart Modern POS.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Fiscal registration service integration sample for Germany

2/18/2021 • 21 minutes to read • [Edit Online](#)

## Introduction

To meet local fiscal requirements for cash registers in Germany, the Microsoft Dynamics 365 Commerce functionality for Germany includes a sample integration of the point of sale (POS) with an external fiscal registration service. The sample extends the [fiscal integration functionality](#). It's based on the [EFR \(Electronic Fiscal Register\)](#) solution from [EFSTA](#) and enables communication with the EFR service via the HTTPS protocol. The EFR service should be hosted on either the Retail Hardware station or a separate computer that can be connected to from the Hardware station. The sample is provided in the form of source code and is part of the Retail software development kit (SDK).

Microsoft doesn't release any hardware, software, or documentation from EFSTA. For information about how to get the EFR solution and operate it, contact [EFSTA](#).

## Scenarios

The following scenarios are covered by the fiscal registration service integration sample for Germany.

### Sales operations

- **Registration of cash-and-carry sales and returns in the fiscal registration service:**

Registration of sales operations includes the following steps:

1. Registration of the transaction start

The start of each transaction is registered in a technical security element (TSE) that is connected to the EFR service. As a result of registration a TSE assigns a transaction ID (TID).

2. Registration of the transaction end

When a transaction is concluded at the POS, it's registered by using the same TID that was assigned during registration of the transaction start. At that moment, detailed transaction data is sent to the fiscal registration service. This data includes sales line information, and information about discounts, payments, and taxes.

3. Capturing a response from the fiscal registration service

Security data is received from a TSE as a part of a response and is saved in the transaction in the channel database. The security data consist of the following information: - TID - Date and time of the transaction start - Date and time of the transaction end - Signature counter - Check value - Serial number of the TSE

- **Registration of customer orders in the fiscal registration service:** The registration process is the same as the process for cash-and-carry sales and returns.
- **Registration of operations that involve gift cards and deposits:** The registration process is the same as the process for cash-and-carry sales and returns.

### Notifying users about fiscal registration failures

There are two ways that the fiscal registration service can notify users about failures that occurred during the fiscal registration:

- Print additional information from the response in the **Info message** field on receipts.

- Show notifications from the fiscal service as user messages at the POS.

#### NOTE

This notification mechanism requires that the **Show fiscal registration notifications** parameter on the **Connector technical profiles** page be turned on.

#### Printing receipts

Receipt printing is mandatory in Germany. All receipts must contain at least the following information:

- Name and address of the company
- Information about goods, including their prices and quantities
- Information about payments that were received
- Information about taxes, including total amounts per tax rate
- Security data:
  - TID
  - Date and time of the transaction start
  - Date and time of the transaction end
  - Signature counter
  - Check value
  - Serial number of the TSE
- Informational message

#### NOTE

A QR code can also be printed on receipts. Although the QR code is optional, it's highly recommended. For more information about how to get QR code as a part of a response from the fiscal registration service, see the "EFR Guide [DE]" document that is published on the [EFSTA documentation](#) website.

The **Info message** field on receipts shows a notification from the fiscal registration service. For example, if a signature device is broken, special text can be printed on a receipt.

#### Voided, suspended, and recalled transactions

- A voided transaction is registered as a request to terminate a transaction in the fiscal registration service.
- A suspended transaction is registered as a request to terminate a transaction in the fiscal registration service.
- Recall of a suspended transaction is registered as the start of a new transaction in the fiscal registration service.

#### Non-sales transactions and shift closing

The following non-sales transactions are registered as non-fiscal operations in the fiscal registration service by using the **NFS** tag:

- Declare start amount
- Float entry
- Tender removal
- Safe drop
- Bank drop
- Income accounts
- Expense accounts

The **Close shift** operation is also registered as a non-fiscal operation in the fiscal registration service by using the **NFS** tag.

### Data export and audit

All transactions must be signed by a TSE to ensure their integrity, authenticity, and completeness, and to help prevent manipulation of recorded data.

#### WARNING

Only a certified TSE can be used. For information about the types and models of TSEs that are supported in the EFR solution, see the "EFR Guide [DE]" document that is published on the [EFSTA documentation](#) website. For information about how to choose and obtain a TSE, contact [EFSTA](#).

Regulations in Germany require support for the DSFinV-K export. The DSFinV-K export can be triggered in the EFR solution. For more information about the DSFinV-K export, see the "EFR Guide [DE]" document that is published on the [EFSTA documentation](#) web-site.

### Default data mapping

The following default data mapping is included in the fiscal document provider configuration that is provided as part of the fiscal integration sample.

- **Tender type mapping:** 1: 0; 2: 1; 3: 3; 4: 8; 5: 2; 6: 0; 7: 7; 8: 6; 9: 0; 10: 8; 11: 1

#### NOTE

In each pair of values that is separated by a semicolon (;), the first number refers to a payment method that is set up for the store. The second number refers to a corresponding payment group in the EFR service, as represented by the **PayG** attribute.

- **Value-added tax (VAT) rates mapping:** A: 19.00; B: 7.00; C: 10.70; D: 5.50; E: 0.00

#### NOTE

In each pair of values that is separated by a semicolon (;), the letter refers to a tax group in the EFR service, as represented by the **TaxG** attribute. The number refers to the tax percentage.

- **Tax group for gift cards and deposits:** G
- **Tax group for VAT exempt:** F

#### WARNING

Tax settings in the default data mapping are responsible for matching tax settings in the system and tax groups in the EFR service. Tax groups can be printed on receipts only if the **Code for printing** field is set on the **Sales tax codes** page.

### Limitations of the sample

The fiscal registration service supports only scenarios where sales tax is included in prices. Therefore, the **Prices include sales tax** option must be set to **Yes** for both stores and customers.

The fiscal service doesn't support situations where more than one sales tax code is applied to the same transaction line.

The fiscal integration framework doesn't support sales quotations. Therefore, those operations aren't registered in the fiscal service.

# Set up Commerce for Germany

This section describes the Commerce settings that are specific to and recommended for Germany. For more setup information, see [Commerce home page](#).

To use the functionality that is specific to Germany, you must specify the following settings.

- In the primary address of the legal entity, set the **Country/region** field to **DEU** (Germany).
- In the POS functionality profile of every store that is located in Austria, set the **ISO code** field to **DE** (Germany).

You must also specify the following settings for Germany. Be sure to run appropriate distribution jobs after you complete the setup.

## Set up VAT per German requirements

You must create sales tax codes, sales tax groups, and item sales tax groups. You must also set up sales tax information for products and services. For more information about how to set up and use sales tax features, see [Sales tax overview](#).

On sales receipts, you can print an abbreviated code for a sales tax code (for example, "A" or "B"). To make this functionality available, set the **Code for printing** field on the **Sales tax codes** page.

## Set up stores

On the **All stores** page, update the store details. Specifically, set the following parameters:

- In the **Sales tax group** field, specify the sales tax group that should be used for sales to the default customer.
- Set the **Prices include sales tax** option to **Yes**.
- Set the **Name** field to the company name. This change helps ensure that the company name appears on a sales receipt. Alternatively, you can add the company name to the sales receipt layout as free-form text.
- Set the **Tax identification number (TIN)** field to the company identification number. This change helps ensure that the company identification number appears on a sales receipt. Alternatively, you can add the company identification number to the sales receipt layout as free-form text.

## Set up functionality profiles

Set up POS functionality profiles. On the **Receipt numbering** FastTab, set up receipt numbering by creating or updating records for the **Sale**, **Sales order**, and **Return** receipt transaction types.

## Configure custom fields so that they can be used in receipt formats for sales receipts

You can configure the language text and custom fields that are used in the POS receipt formats. The default company of the user who creates the receipt setup should be the same legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

On the **Language text** page, add the following records for the labels of the custom fields for receipt layouts. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the table are just examples. You can change them to meet to your requirements. However, the **Text ID** values that you use must be unique, and they must be equal to or more than 900001.

Add the following POS labels to the **POS** section of the **Language text** page.

LANGUAGE ID	TEXT ID	TEXT
en-US	900001	QR code

LANGUAGE ID	TEXT ID	TEXT
en-US	900002	Transaction ID
en-US	900003	Tax Retail Print Code
en-US	900004	Tax Amount (sales)
en-US	900005	Tax Basis (sales)
en-US	900006	Transaction start date time
en-US	900007	Transaction end date time
en-US	900008	Serial number of the security element
en-US	900009	Signature counter
en-US	900010	Check value
en-US	900011	Info message

On the **Custom fields** page, add the following records for the custom fields for receipt layouts. Note that the **Caption text ID** values must correspond to the **Text ID** values that you specified on the **Language text** page.

NAME	TYPE	CAPTION TEXT ID
QRCODE_DE	Receipt	900001
TRANSACTIONID_DE	Receipt	900002
RETAILPRINTCODE_DE	Receipt	900003
SALESTAXAMOUNT_DE	Receipt	900004
SALESTAXBASIS_DE	Receipt	900005
TRANSACTIONSTARTDATETIME_DE	Receipt	900006
TRANSACTIONENDDATETIME_DE	Receipt	900007
SECURITYELEMENTSERIALNUMBER_DE	Receipt	900008
SIGNCOUNTER_DE	Receipt	900009
SIGN_DE	Receipt	900010
INFOMESSAGE_DE	Receipt	900011

### Configure receipt formats

For every receipt format that is required, change the value of the **Print behavior** field to **Always print**.

In the Receipt format designer, add the following custom fields to the appropriate receipt sections. Note that field names correspond to the language texts that you defined in the previous section.

- **Header:** Add the following fields:
  - **Store name** and **Tax Identification Number** fields, which are used to print the company name and identification number on receipts. Alternatively, you can add the company name and identification number to the layout as free-form text.
  - **Store address**, **Date**, **Time 24H**, **Receipt Number**, and **Register number** fields.
- **Lines:** Add the following fields:
  - **Item name** field
  - **Qty** field
  - **Total price with tax** field
  - **Tax Retail Print Code** field, which is used to print the abbreviated code that corresponds to the sales tax code that applies to the item
- **Footer:** Add the following fields:
  - Payment fields, so that the payment amounts for each payment method are printed. For example, add the **Tender name** and **Tender amount** fields to one line of the layout.
  - Fields in the **Tax break down** field group. The fields in this field group must be printed on a separate line.
    - **Tax Id** field, which is a standard field that enables a sales tax summary to be printed for each sales tax code. The field must be added to a new line.
    - **Tax Percentage** field, which is a standard field that is used to print the effective tax rate for the sales tax code.
    - **Tax Basis (sales)** field, which is used to print the receipt's total cash sale amount for the sales tax code. Prepayments and gift card operations are excluded.
    - **Tax Amount (sales)** field, which is used to print the receipt's tax amount for cash sales for the sales tax code.
    - **Tax Retail Print Code** field, which is used to print the abbreviated code that corresponds to the sales tax code.
  - Fields that contain secured transaction data that is returned by the fiscal registration service:
    - **Transaction ID** field, which identifies the number of the cash transaction in the fiscal registration service
    - **Transaction start date time** field
    - **Transaction end date time** field
    - **Serial number of the security element** field
    - **Signature counter** field
    - **Check value** field
    - **QR Code** field, which is used to print the reference to the registered cash transaction in the form of a QR code
  - **Info message** field, so that notification messages from the fiscal registration service can be shown on receipts. For example, if a signature device is broken, special text can be printed on a receipt.

For more information about how to work with receipt formats, see [Set up and design receipt formats](#).

### **Configure fiscal integration**

Set up the fiscal integration by completing the following steps that are described in [Set up the fiscal integration](#)

for Commerce channels:

1. [Set up a fiscal registration process](#). Also note the settings for the fiscal registration process that are [specific to this fiscal registration service integration sample](#).
2. [Set error handling settings](#).

#### WARNING

The error handling capabilities of the fiscal integration framework might be not fully aligned with the local fiscal regulations.

- We recommend that you leave the **Continue on error** option on the **Fiscal registration process** page turned off, because all transactions must be correctly registered, even if the first attempt at fiscal registration wasn't successful.
- Before you turn on the **Skip or Mark as registered** option on the **Fiscal registration process** page, you should discuss these changes to the fiscal registration process with your tax consultant or the local tax office.

3. [Enable manual execution of postponed fiscal registration](#).

## Deployment guidelines for cash registers for Germany

The fiscal registration service integration sample for Germany is part of the Retail SDK. For more information about how to install and use the Retail SDK, see the [Retail software development kit \(SDK\) architecture](#).

This sample consists of extensions for the Commerce runtime (CRT) and Hardware station. To run this sample, you must modify and build the CRT and Hardware station projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Azure DevOps, where no files have been changed yet.

Follow these steps to set up a development environment so that you can test and extend the sample.

### Enable CRT extensions

The CRT extension components are included in the CRT samples. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`, under `RetailSdk\SampleExtensions\CommerceRuntime`.

#### DocumentProvider.EFRSample component

1. Find the `Runtime.Extensions.DocumentProvider.EFRSample` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.EFRSample\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.EFRSample.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit:** Copy the assembly to the `\bin\ext` folder under the Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file:

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.EFRSample" />
```



#### DocumentProvider.DataModelEFR component

1. Find the `Runtime.Extensions.DocumentProvider.DataModelEFR` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.DataModelEFR\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit:** Copy the assembly to the `\bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
```

#### Update the extension configuration file

1. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
2. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsGermany" />
```

#### Enable Hardware station extensions

The Hardware station extension components are included in the Hardware station samples. To complete the following procedures, open the solution `HardwareStationSamples.sln.sln` under `RetailSdk\SampleExtensions\HardwareStation`.

#### EFRSample component

1. Find the `HardwareStation.Extension.EFRSample` project, and build it.
2. In the `Extension.EFRSample\bin\Debug` folder, find following assembly files:
  - `Contoso.Commerce.HardwareStation.EFRSample.dll`
  - `Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll`
3. Copy the assembly files to the Hardware station extensions folder:
  - **Shared hardware station:** Copy the files to the `bin` folder under the IIS Hardware station site location.
  - **Dedicated hardware station on Modern POS:** Copy the files to the Modern POS client broker location.
4. Find the extension configuration file for the Hardware station's extensions. The file is named `HardwareStation.Extension.config`.

- **Shared hardware station:** The file is located under the IIS Hardware station site location.
- **Dedicated hardware station on Modern POS:** The file is located under the Modern POS client broker location.

5. Add the following line to the **composition** section of the configuration file.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.EFRSample.dll" />
```

### Set up the registration process

To enable the registration process, follow these steps to set up Commerce Headquarters. For more details, see [Set up the fiscal integration for Commerce channels](#).

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**. On the **General** tab, set the **Enable fiscal integration** option to **Yes**.
2. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connectors**, and load the connector configuration. The file location is **RetailSdk\SampleExtensions\HardwareStation\Extension.EFRSample\Configuration\ConnectorEFRSample.xml**.
3. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal document providers**, and load the document provider configurations. The configuration file is **RetailSdk\SampleExtensions\CommerceRuntime\Extensions.DocumentProvider.EFRSample\Configuration\DocumentProviderFiscalEFRSampleGermany.xml**.
4. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector functional profiles**. Create a new connector functional profile, and select the document provider and the connector that you loaded earlier. Update the data mapping settings as required.

#### NOTE

By default, the **Include customer data** option is set to **Yes**. If you don't want customer information such as names and addresses to be sent to the fiscal registration service, you can change the setting to **No**.

5. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector technical profiles**. Create a new connector technical profile, and select the connector that you loaded earlier. Update the connection settings as required.

#### WARNING

By default, the **Show fiscal registration notifications** parameter is turned on. We recommend that you leave it turned on, because the fiscal registration service sends notifications about some specific errors that might occur during fiscal registration (for example, a transaction wasn't signed at the moment of registration).

6. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connector groups**. Create a new fiscal connector group for the connector functional profile that you created earlier.
7. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal registration processes**. Create a new fiscal registration process, create a fiscal registration process step, and select the fiscal connector group that you created earlier.
8. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. Select a functionality profile that is linked to the store where the registration process should be activated. On the **Fiscal registration process** FastTab, select the fiscal registration process that you created earlier.

9. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**. Select a hardware profile that is linked to the Hardware station that the fiscal printer will be connected to. On the **Fiscal peripherals** FastTab, select the connector technical profile that you created earlier.
10. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**, and run jobs 1070 and 1090 to transfer data to the channel database.

### Production environment

In the previous procedure, you enabled the extensions that are components of the fiscal registration service integration sample. In addition, you must follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Make the following changes in the package configuration files under the **RetailSdk\Assets** folder:

- In the **commerceruntime.ext.config** and **CommerceRuntime.MPOSOOffline.Ext.config** configuration files, add the following lines to the **composition** section.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsGermany" />
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.EFRSample" />
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
```

- In the **HardwareStation.Extension.config** configuration file, add the following lines to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.EFRSample" />
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR" />
```

2. Make the following changes in the **BuildTools\Customization.settings** package customization configuration file:

- Add the following lines to include the CRT extensions in the deployable packages.

```
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.EFRSample.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll" />
```

- Add the following lines to include the Hardware station extensions in the deployable packages.

```
<ISV_HardwareStation_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.EFRSample.dll" />
<ISV_HardwareStation_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.DataModelEFR.dll" />
```

3. Start the MSBuild Command Prompt for Visual Studio utility, and run **msbuild** under the Retail SDK folder to create deployable packages.
4. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).
5. Complete all the required setup tasks that are described in the [Set up Commerce for Germany](#) section earlier in this topic.

# Design of extensions

## CRT extension design

The purpose of the extension that is a fiscal document provider is to generate service-specific documents and handle responses from the fiscal registration service.

The CRT extension is `Runtime.Extensions.DocumentProvider.EFRSample`. For more details about the design of the fiscal integration solution, see [Overview of fiscal integration for Commerce channels](#).

### Request handler

There is one request handler for the document provider, `DocumentProviderEFRFiscalDEU`. This handler is used to generate fiscal documents for the fiscal registration service. It's inherited from the `INamedRequestHandler` interface. The `HandlerName` method is responsible for returning the name of the handler. The handler name should match the connector document provider name that is specified in Commerce Headquarters.

The connector supports the following requests:

- **GetFiscalDocumentDocumentProviderRequest** – This request contains information about what document should be generated. It returns a service-specific document that should be registered in the fiscal registration service.
- **GetFiscalTransactionExtendedDataDocumentProviderRequest** – This request returns the response together with extended data.

### Configuration

The `DocumentProviderFiscalEFRSampleGermany` configuration file is located in the **Configuration** folder of the extension project. The purpose of this file is to enable settings for the document provider to be configured from Commerce Headquarters. The file format is aligned with the requirements for fiscal integration configuration.

The following settings are added:

- **VAT rates mapping** – The mapping of tax percentage values that are set up for the sales tax codes to values of the `TaxG` (tax group) attribute in requests that are sent to the fiscal service.
- **Tax group for gift cards and deposits** – The value of the `TaxG` attribute in requests that are sent to the fiscal service, based on operations that involve gift cards or deposits.
- **Tender type mapping** – The mapping of payment methods to values of the `PayG` (payment group) attribute in requests that are sent to the fiscal service.
- **Tax group for VAT exempt** – The value of the `TaxG` attribute in requests that are sent to the fiscal service, based on operations that are exempt from tax obligations.
- **Include customer data** – If this parameter is turned on, requests to the fiscal service will contain customer information such as names and addresses in cases where a customer is added to a transaction.

## Hardware station extension design

The purpose of the extension that is a fiscal connector is to communicate with the fiscal registration service.

The Hardware station extension is `HardwareStation.Extension.EFRSample`. It uses the HTTP protocol to submit documents that the CRT extension generates to the fiscal registration service. It also handles the responses that are received from the fiscal registration service.

### Request handler

The `EFRHandler` request handler is the entry point for handling requests to the fiscal registration service. This handler is inherited from the `INamedRequestHandler` interface. The `HandlerName` method is responsible for returning the name of the handler. The handler name should match the fiscal connector name that is specified in Commerce Headquarters.

The connector supports the following requests:

- **SubmitDocumentFiscalDeviceRequest** – This request sends documents to the fiscal registration service and returns a response from it.
- **IsReadyFiscalDeviceRequest** – This request is used for a health check of the fiscal registration service.
- **InitializeFiscalDeviceRequest** – This request is used to initialize the fiscal registration service.

#### **Configuration**

The configuration file is located in the **Configuration** folder of the extension project. The purpose of the file is to enable settings for the fiscal connector to be configured from Commerce Headquarters. The file format is aligned with the requirements for fiscal integration configuration.

The following settings are added:

- **Endpoint address** – The URL of the fiscal registration service.
- **Timeout** – The amount of time, in milliseconds (ms), that the driver will wait for a response from the fiscal registration service.
- **Show fiscal registration notifications** – If this parameter is turned on, notifications from the fiscal service will be shown as user messages at the POS.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Goods and Services Tax (GST) integration for cash registers for India

2/18/2021 • 12 minutes to read • [Edit Online](#)

This topic provides a walkthrough of the features that are related to Goods and Services Tax (GST). It also highlights the effect of GST on various types of commerce business transactions, and shows the accounting and posting of transactions where the receipt is printed at the point of sale (POS).

## Prerequisites

- Set up GST for India. For more information, see [India Goods and Services Tax \(GST\)](#).
- Configure Commerce channel components. To enable India-specific functionality, you must configure extensions for channel components. For more information, see the [deployment guidelines](#).

## India tax entities for Commerce

The following table shows the navigation paths for the India tax entities in Commerce.

INDIA TAX ENTITIES	NAVIGATION PATH IN COMMERCE
Business verticals	Retail and Commerce > Channel setup > Sales taxes > Business verticals
Enterprise tax registration numbers	Retail and Commerce > Channel setup > Sales taxes > Enterprise tax registration numbers
GST reference number sequence group	Retail and Commerce > Channel setup > Sales taxes > GST reference number sequence group
HSN codes	Retail and Commerce > Channel setup > Sales taxes > HSN codes
Service accounting codes	Retail and Commerce > Channel setup > Sales taxes > Service accounting codes
Maintain setoff hierarchy profiles	Retail and Commerce > Channel setup > Sales taxes > Maintain setoff hierarchy profiles
VAT schedules	Retail and Commerce > Channel setup > Sales taxes > VAT schedules
Tax setup	Retail and Commerce > Channel setup > Sales taxes > Tax configuration > Tax setup

### NOTE

The navigation paths for the India tax entities in Commerce differ from the navigations paths in Finance. For information about the navigation paths in Finance, see [India Goods and Services Tax \(GST\)](#).

# Validate tax information for the store

The tax information for the store comes from the selected warehouse. This warehouse is defined in the warehouse master. The configured tax information from the store is printed on the POS receipt. It's also updated on the sales order at the headquarters for the financial postings.

Follow these steps to view the tax information for a store.

1. Go to **Retail and Commerce > Channels > Stores > All stores**.
2. Select a store.
3. Select the **Tax information** FastTab.

## Configure language texts and custom fields

You can configure the language text and custom fields that are used in the POS receipt formats. The default company of the user who creates the receipt setup should be the same as the legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

### Set up the POS language text

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profile > Language text**.
2. On the **POS** tab, on the **POS language text** FastTab, select the language ID for the text. The language should match the user's preferred language.
3. In the **Text ID** field, enter a unique ID that is equal to or more than **900001**.
4. In the **Text** field, enter the language text.

Language text

POS

Commerce Scale Unit

Setup POS language text for user interface fields and POS messages.

POS language text

+ Add Remove

Filter

Language ID	Text ID ↑	Text
en-US	900001	GST TIN No
en-US	900002	Company PAN No
en-US	900003	VAT TIN No
en-US	900004	SAC
en-US	900005	HSN
en-US	900006	Tax Component

### Create custom fields

When you create custom fields, the value of the **Caption text ID** field must match the value that you entered for the **Text ID** field on the **Language text** page.

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profile > Custom fields**.
2. Enter a name for the field.
3. Select the field type.
4. In the **Caption text ID** field, enter the **Text ID** value for one of the language texts on the **Language text**

page.

### Custom fields

Filter

Name	Type	Caption text ID ↑
GSTINNO_IN	Receipt	900001
COMPANYPANNO_IN	Receipt	900002
VATTINNO_IN	Receipt	900003
SACCODE_IN	Receipt	900004
HSNCODE_IN	Receipt	900005
TAXCOMPONENT_IN	Receipt	900006

## Create the receipt format

You can use Receipt format designer to add custom fields to the appropriate receipt sections. For more information, see [Set up and design receipt formats](#).

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profile > Receipt formats**.
2. Select a receipt format for the **Receipt** receipt type, and make the required changes.

Receipt format design - Receipt format ID: IN01

Save

Form Information Form Id IN01 Form name Receipt Form description Customers receipt

Header

EFT Time 24H

Entry Source Code

Euro Auth Code

Expiry date

GST TIN No

HSN

Invoice account number

Invoice Comment

Invoice customer address

Header

Lines

Footer

Object information Align Left Fill character Prefix Characters 15 Variable Font type Regular Font size Re... Select image Delete

## Update receipt profiles

After you create a receipt format, you can assign that format to a receipt profile.

Follow these steps to update a receipt profile.

1. Go to **Retail and Commerce > Setup > POS > Receipt profile**.
2. Select the receipt profile to update.
3. Select **Edit**.

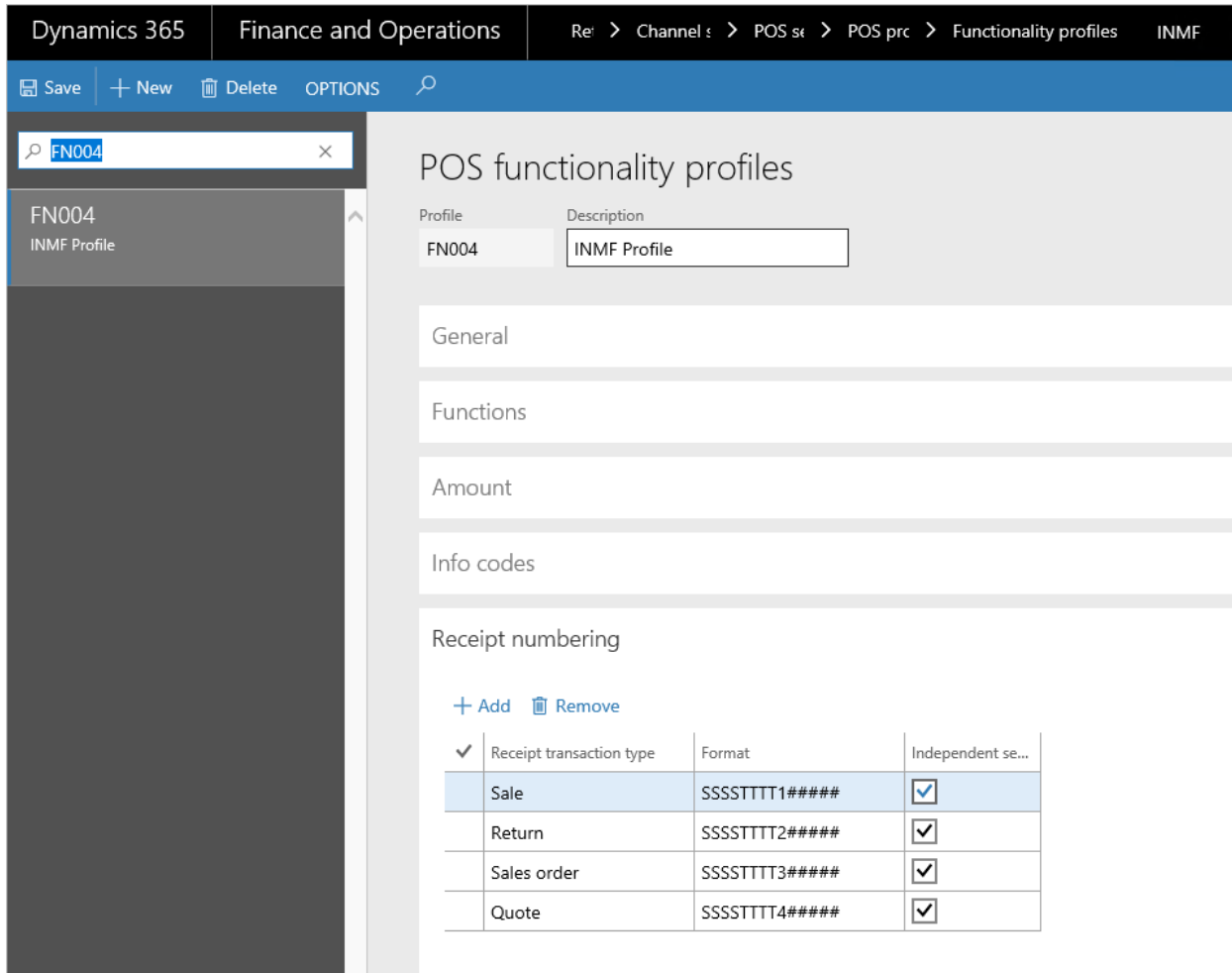


4. For each receipt type in the list, select a receipt format.

## Update the POS invoice number

You can reconcile the POS receipt number with the invoice number for customer transactions. If you set the **Update POS invoice number** option to **Yes** on the **Posting** tab of the **Commerce parameters** page, the POS receipt number is entered in the **Transaction ID** field for corresponding sales orders.

You can set the **Update POS invoice number** option to **Yes** only if the existing receipt number format includes both the store number and the terminal number. The following illustration shows a POS functionality profile where the receipt numbering includes the store number and the terminal number.



The screenshot shows the Dynamics 365 interface for 'POS functionality profiles'. The breadcrumb navigation is 'Retail > Channel > POS st > POS prc > Functionality profiles > INMF'. The page title is 'POS functionality profiles'. The profile name is 'FN004' and the description is 'INMF Profile'. The 'Receipt numbering' section contains a table with the following data:

Receipt transaction type	Format	Independent se...
Sale	SSSSTTTT1#####	<input checked="" type="checkbox"/>
Return	SSSSTTTT2#####	<input checked="" type="checkbox"/>
Sales order	SSSSTTTT3#####	<input checked="" type="checkbox"/>
Quote	SSSSTTTT4#####	<input checked="" type="checkbox"/>

## Run a distribution schedule

To synchronize Tax Engine (GTE) data from headquarter to the POS database, you must add a job to the **Distribution schedule** page.

Follow these steps to verify that the job exists and to run the job.

1. Go to **Retail and Commerce > Periodic > Data distribution > Distribution schedule**.
2. Verify that a new job, **1180**, has been added for **Generic tax engine**.
3. Run all the jobs (**9999**).

## Example scenarios

The following example scenarios walk you through sample transactions for India:

- Scenario 1: Sell to a registered customer
- Scenario 2: Sell taxable goods to a consumer

- Scenario 3: Sell taxable goods to an anonymous customer where GST is price-inclusive
- Scenario 4: Sell an exempted good
- Scenario 5: Return the transaction that has GST

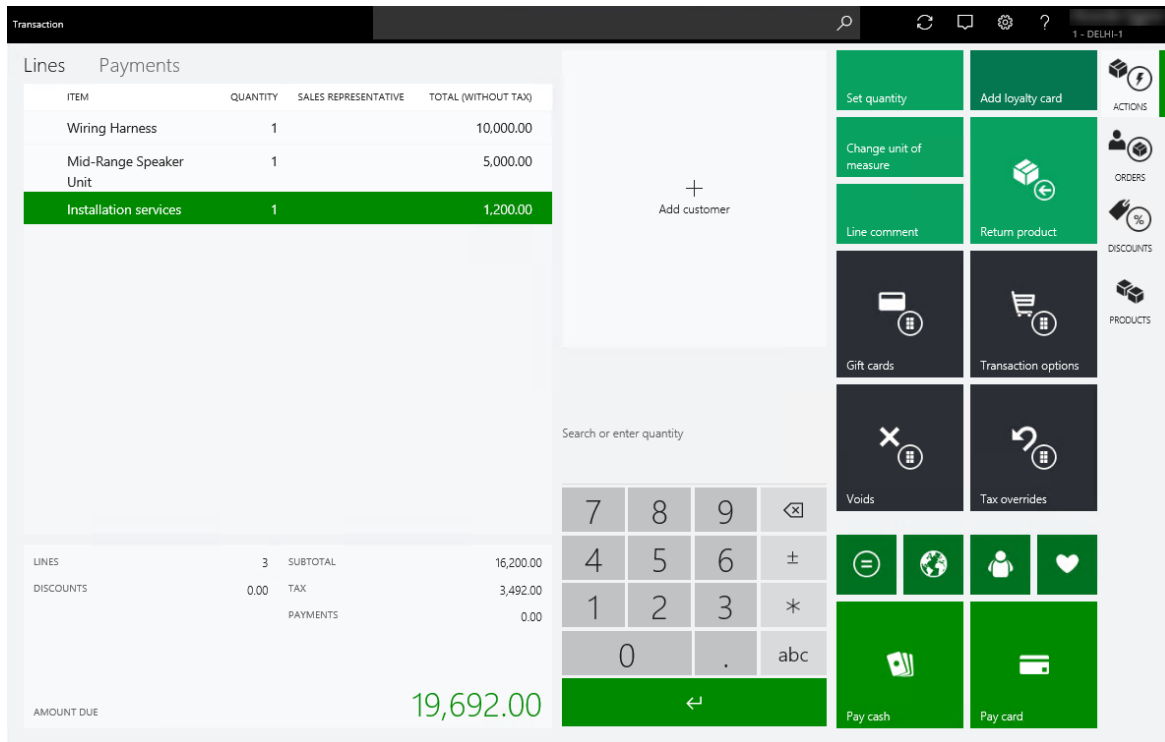
**Scenario 1: Sell to a registered customer**

Sales to a registered customer are known as *business-to-business* (B2B) sales. If the store location and the place of supply (that is, the customer address) are in the same state, the transaction is an *intrastate sale*, and Central GST (CGST) and State GST (SGST) must be paid. If the store location and the place of supply are in different states, the transaction is an *interstate sale*, and Integrated GST (IGST) must be paid.

**NOTE**

For these transactions, the fields for the customer address and the registration number are required.

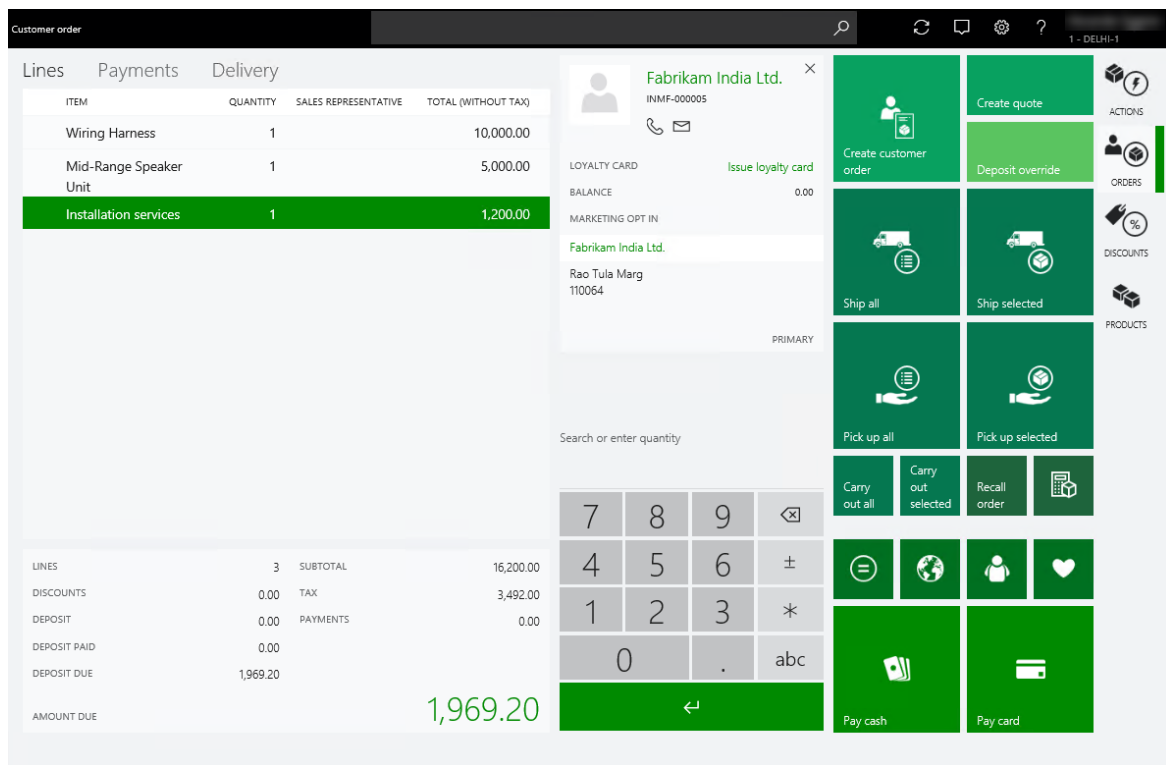
1. Sign in to the POS.
2. Enter the items, and then select **Enter**.



3. Validate the GST calculations. Consider the rate that is defined in the tax setup.

ITEM/SERVICE	UNIT PRICE	TAX RATES	CGST	SGST
M0001	10,000.00	CGST 12%, SGST 11%	1,200.00	1,100.00
M0002	5,000.00	CGST 10%, SGST 10%	500.00	500.00
S0001	1,200.00	CGST 11%, SGST 5%	132.00	60.00
	16,200.00		1,832.00	1,660.00
Total amount				19,692.00

4. Select **Orders** > **Create customer order**.
5. Select **Add customer**, and select the customer account.
6. Select **Pick up all**.
7. Select the store and the pick-up date.
8. Select **OK**.



**NOTE**

In this example, the state of the store location is Delhi, and the state of the customer address is also Delhi. Because the state is the same, intrastate GST is calculated.

9. Select **Exact** to process the deposit payment.
10. Validate the receipt:
  - a. Select **Show journal**.
  - b. Select the transactions.
  - c. Select **Receipt**.

Transaction:	INMF-000167	Date:	25-01-20
Terminal:	DELHI-1	Time:	18:03
Employee:	000160		
Customer No.:	INMF-000005		
Customer name:	Fabrikam India Ltd.		
Company PAN No	AAACC1478C		
GSTIN No.	07AEVPG9380C1Z7		
TIN No.:			
Item name	Qty	Price	Total
-----	-----	-----	-----
Wiring Harness	1.00	10,000.00	10,000.00
HSN/SAC	70119010		
Mid-Range Speaker Unit	1.00	5,000.00	5,000.00
HSN/SAC	50020010		
Installation services	1.00	1,200.00	1,200.00
HSN/SAC	999210		
-----	-----	-----	-----
Subtotal			16,200.00
Tax			3,492.00
Total			19,692.00
Cash			1,969.20
=====			
<B: ELHIHI-13_000017>			
Tax components	Tax code	Tax per	Tax amount
CGST	AA	12.00	1,200.00
	AB	10.00	500.00
	AC	11.00	132.00
SGST	BA	11.00	1,100.00
	BB	10.00	500.00
	BC	5.00	60.00

11. Validate the sales order and tax document in Headquarters:

- a. Go to **Retail and Commerce > Customers > All sales orders.**
- b. Select the sales order.
- c. On the Action Pane, on the **Sell** tab, in the **Tax** group, select **Tax document.**

Apply adjustment Recalculate Tax applicability Tax document JSON OPTIONS							
Header - SalesTable SalesId: INMF-000167, Sale							
Lines - SalesLine LineNum: 1.00, ItemId: M000							
GST							
CGST							
SGST							
Lines - SalesLine LineNum: 2.00, ItemId: M000							
GST							
CGST							
SGST							
Lines - SalesLine LineNum: 3.00, ItemId: S000							
GST							
CGST							
SGST							
Tax document							
Sales order							
Tax details							
Tax details Adjustment							
Tax type	Tax component	Base Amount (tr...	Base Amount (a...	Rate	Tax Amount (tra...	Tax Amount (ac...	
CGST	CGST	10,000.00	10,000.00	12.00000	1,200.00	1,200.00	
CGST	CGST	5,000.00	5,000.00	10.00000	500.00	500.00	
CGST	CGST	1,200.00	1,200.00	11.00000	132.00	132.00	
SGST	SGST	10,000.00	10,000.00	11.00000	1,100.00	1,100.00	
SGST	SGST	5,000.00	5,000.00	10.00000	500.00	500.00	
SGST	SGST	1,200.00	1,200.00	5.00000	60.00	60.00	

12. Recall and process the customer order:

- a. Sign in to the POS.
- b. Select **Recall order.**
- c. Search for and select the order.
- d. Select **Picking and packing > Pickup**
- e. Select **Select all** and then **Pickup.**

Customer order pickup

Lines Payments

ITEM	QUANTITY	SALES REPRESENTATIVE	TOTAL (WITHOUT TAX)
Wiring Harness	1		10,000.00
Mid-Range Speaker Unit	1		5,000.00
Installation services	1		1,200.00

Fabrikam India Ltd. INMF-000005

LOYALTY CARD Issue loyalty card

BALANCE 0.00

MARKETING OPT IN

Fabrikam India Ltd.  
Rao Tula Marg  
110064

PRIMARY

Search or enter quantity

7	8	9	⌫
4	5	6	±
1	2	3	*
0	.	abc	

←	→	↶	↷
⊞	🌐	👤	❤️
👉	👉	👉	👉
👉	👉	👉	👉

LINES	3	SUBTOTAL	16,200.00
DISCOUNTS	0.00	TAX	3,492.00
DEPOSIT	1,969.20	PAYMENTS	0.00
DEPOSIT PAID	1,969.20		
DEPOSIT DUE	17,722.80		
AMOUNT DUE			17,722.80

Pay cash Pay card

13. Select Exact to process the payment.

14. Validate the receipt.

```

Transaction: INMF-000167      Date: 25-01-20
Terminal:    DELHI-1         Time: 18:59
Employee:    000160
Customer No.: INMF-000005
Customer name: Fabrikam India Ltd.
Company PAN No AAACC1478C

GSTIN No.    07AEVPG9380C127

TIN No. :

Item name          Qty      Price      Total
-----
wiring Harness    1.00     10,000.00  10,000.00
HSN/SAC           70119010

Mid-Range Speaker Unit 1.00     5,000.00   5,000.00
HSN/SAC           50020010

Installation services 1.00     1,200.00   1,200.00
HSN/SAC           999210

-----
Subtotal          16,200.00
Tax                3,492.00
Total             19,692.00

Cash              17,722.80

=====
<B: ELHIHI-13_000018>

Tax components Tax code  Tax per Tax amount
CGST           AA           12.00     1,200.00
               AB           10.00     500.00
               AC           11.00     132.00
SGST           BA           11.00     1,100.00
               BB           10.00     500.00
               BC           5.00      60.00
  
```

15. Validate the voucher transactions:

- a. Go to **Retail and Commerce > Customers > All sales orders**.
- b. Select the sales order.
- c. On the Action Pane, on the **Invoice** tab, select **Invoice journals**.
- d. Select **Voucher**.

LEDGER ACCOUNT NAME	DEBIT AMOUNT (RS.)	CREDIT AMOUNT (RS.)
Customer account	19,692.00	
CGST payable account		1,832.00
SGST payable account		1,660.00
Sales account		16,200.00

- e. Select **Tax document**.
- f. Verify that the receipt number is updated as the transaction ID.

The screenshot displays the 'Tax document' window for a sales invoice. The 'Transaction Id' is 'IHI-13\_000018'. The 'Tax details' table is as follows:

Tax type	Tax component	Base Amount (tr...)	Base Amount (a...)	Rate	Tax Amount (tra...)	Tax Amount (ac...)
GST	CGST	10,000.00	10,000.00	12.00000	1,200.00	1,200.00
GST	CGST	5,000.00	5,000.00	10.00000	500.00	500.00
GST	CGST	1,200.00	1,200.00	11.00000	132.00	132.00
GST	SGST	10,000.00	10,000.00	11.00000	1,100.00	1,100.00
GST	SGST	5,000.00	5,000.00	10.00000	500.00	500.00
GST	SGST	1,200.00	1,200.00	5.00000	60.00	60.00

### Scenario 2: Sell taxable goods to a consumer

When you sell to unregistered customers, the sales are referred to as *business-to-consumer* (B2C) sales. Tax is calculated in the same manner for B2B and B2C sales.

1. Sign in to the POS.
2. Enter an item, and then select **Enter**.
3. Select **Add customer**, and select the customer account.

Transaction

Lines Payments

ITEM	QUANTITY	SALES REPRESENTATIVE	TOTAL (WITHOUT TAX)
Wiring Harness	1		10,000.00

Wingtip Toys India Lt...  
INMF-000001

LOYALTY CARD Issue loyalty card

BALANCE 0.00

MARKETING OPT IN

Wingtip Toys India Ltd.  
Ashoka Pillar road  
Bangalore, Bangalore - 560030

PRIMARY

Search or enter quantity

7	8	9	⊗
4	5	6	±
1	2	3	*
0	.	abc	

AMOUNT DUE 12,500.00

Pay cash Pay card

Create quote  
Create customer order  
Deposit override  
Ship all Ship selected  
Pick up all Pick up selected  
Carry out all Carry out selected Recall order  
Pay cash Pay card

#### NOTE

In this example, the state of the store location is Delhi, but the state of the customer address is Bengaluru (Bangalore). Because the states differ, interstate GST is computed.

4. Select **Exact** to process the payment.
5. Validate the receipt:
  - a. Select **Show journal**.
  - b. Select the transactions.
  - c. Select **Receipt**.

Transaction:	DELHI-DELHI-1-116	Date:	25-01-20
Terminal:	DELHI-1	Time:	19:15
Employee:	000160		
Customer No.:	INMF-000001		
Customer name:	Wingtip Toys India Ltd.		
Company PAN No	AAACC1478C		
GSTIN No.	07AEVPG9380C1Z7		
TIN No.:			
Item name	Qty	Price	Total
-----	-----	-----	-----
Wiring Harness	1.00	10,000.00	10,000.00
HSN/SAC	70119010		
-----	-----	-----	-----
Subtotal			10,000.00
Tax			2,500.00
Total			12,500.00
Cash			12,500.00
=====			
<B: ELHIHI-11_000021>			
Tax components	Tax code	Tax per	Tax amount
IGST	AA	25.00	2,500.00

6. Validate the sales invoice in Headquarters:

- a. Go to **Retail and Commerce > Retail and Commerce IT > Data distribution.**
- b. Run job **P-0001 (Channel transactions).**
- c. Close the page.

7. Post the statement:

- a. Go to **Retail and Commerce > Channels > Stores > Open statements.**
- b. Create a statement.
- c. Select **Calculate statement** and then **Post statement.**

8. Validate the voucher transactions:

- a. Go to **Retail and Commerce > Customers > All sales orders.**
- b. Select the sales invoice.
- c. Select **Sales order lines > Tax information.**
- d. On the appropriate tabs, verify the location (store address) and the customer address.



**Tax information**

Location: Retail

Address: M.G.Road, Delhi, Delhi - 110041, Delhi, Delhi, IND

Tax information: GST\_wH

Prices include sales tax: No

Exempt: No

---

**GST**

GSTIN/GDI/UID: 07AEVPG9380C1Z7

HSN codes: 70119010

Service accounting codes:

ITC Category: Input

Is scrap: No

---

**VAT**

Sales tax:

Withholding tax:

Excise:

Service:

---

**Customer tax information**

Location: Wingtip Toys India Ltd.

Address: Ashoka Pillar road, Bangalore, Bangalore - 560030

Tax information:

- e. Select **OK**.
- f. On the Action Pane, on the **Invoice** tab, select **Invoice journals**.
- g. Select **Voucher**.

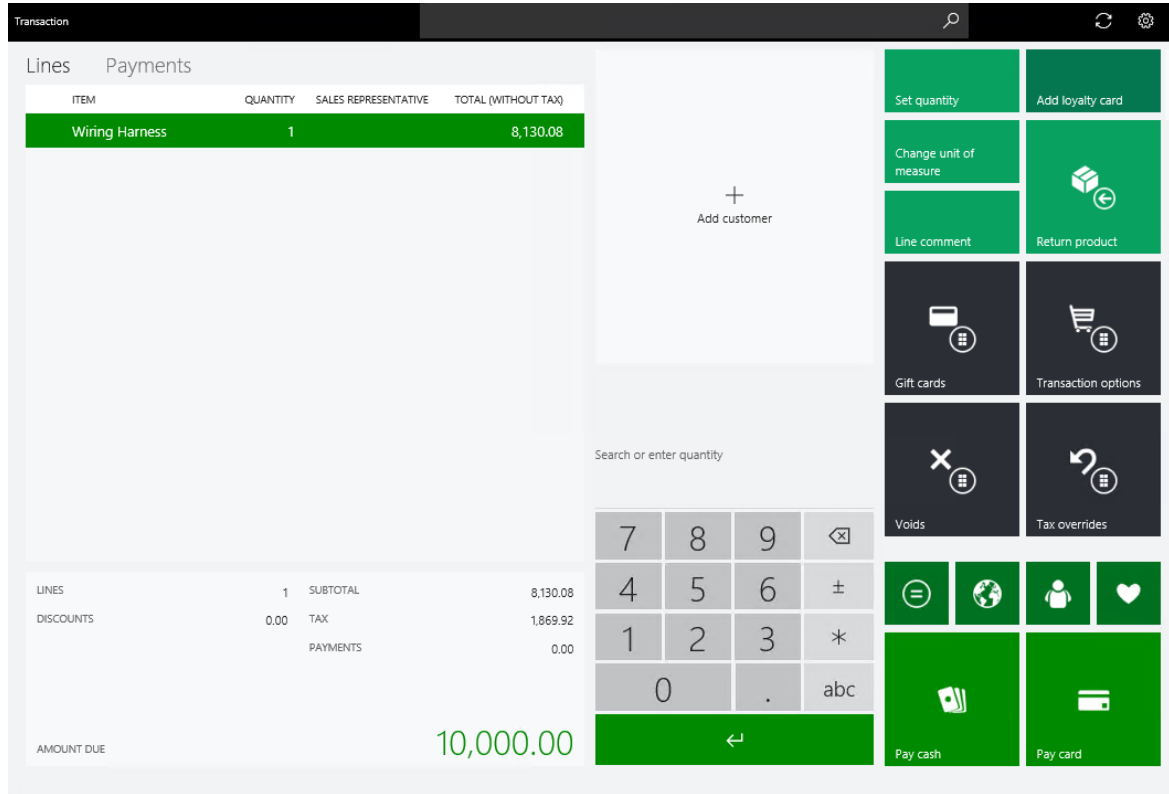
LEDGER ACCOUNT NAME	DEBIT AMOUNT (RS.)	CREDIT AMOUNT (RS.)
Customer account	12,500.00	
IGST payable account		2,500.00
Sales account		10,000.00

- h. Select **Tax document**.
- i. Verify that the receipt number is updated as the transaction ID.

**Scenario 3: Sell taxable goods to an anonymous customer where GST is price-inclusive**

1. Define price-inclusiveness at the store:
  - a. Go to **Retail and Commerce > Channels > Stores > All stores**.
  - b. Select a store.
  - c. Set the **Prices include sales tax** option to **Yes**.
2. Run the distribution schedule:
  - a. Go to **Retail and Commerce > Retail and Commerce IT > Data distribution**.
  - b. Run the job to update the changes in the POS database.

- c. Close the page.
3. Enter a transaction:
  - a. Sign in to the POS.
  - b. Enter an item, and then select **Enter**. For this example, use an item that has the following values:
    - **Taxable value:** 10,000.00
    - **CGST:** 12 percent
    - **SGST:** 11 percent



4. Select **Exact** to process the payment.
5. Validate the receipt.

Transaction:	DELHI-DELHI-1-123	Date:	25-01-20
Terminal:	DELHI-1	Time:	19:50
Employee:	000160		
Customer No.:			
Customer name:			
Company PAN No	AAACC1478C		
GSTIN No.	07AEVPG9380C1Z7		
TIN No.:			
Item name	Qty	Price	Total
-----			
wiring Harness	1.00	10,000.00	10,000.00
HSN/SAC	70119010		
-----			
Subtotal			8,130.08
Tax			1,869.92
Total			10,000.00
Cash			10,000.00
=====			
<B: ELHIHI-11_000023>			
Tax components	Tax code	Tax per	Tax amount
CGST	AA	12.00	975.61
SGST	BA	11.00	894.31

6. Validate the sales invoice in Headquarters:

- Go to **Retail and Commerce > Retail and Commerce IT > Data distribution.**
- Run job **P-0001 (Channel transactions).**
- Close the page.

7. Post the statement:

- Go to **Retail and Commerce > Channels > Stores > Open statements.**
- Create a statement.
- Select **Calculate statement** and then **Post statement.**

8. Validate the voucher transactions:

- Go to **Retail and Commerce > Customers > All sales orders.**
- Select the sales invoice.
- On the Action Pane, on the **Invoice** tab, select **Invoice journals.**
- Select **Voucher.**

LEDGER ACCOUNT NAME	DEBIT AMOUNT (RS.)	CREDIT AMOUNT (RS.)
Customer account	10,000.00	
CGST payable account		975.61

LEDGER ACCOUNT NAME	DEBIT AMOUNT (RS.)	CREDIT AMOUNT (RS.)
SGST payable account		894.31
Sales account		8,130.08

e. Select **Tax document**.

f. Verify that the transaction ID is updated according to the GST number sequence that is defined in the GST reference number sequence group.

Tax type	Tax component	Base Amount (tr...	Base Amount (a...	Rate	Tax Amount (tra...	Tax Amount (ac...
GST	CGST	8,130.08	8,130.08	12.00000	975.61	975.61
GST	SGST	8,130.08	8,130.08	11.00000	894.31	894.31

#### Scenario 4: Sell an exempted good

1. Sign in to the POS.
2. Enter an exempted item.

ITEM	QUANTITY	SALES REPRESENTATIVE	TOTAL (WITHOUT TAX)
Enclosure	1		4,000.00

LINES	1	SUBTOTAL	4,000.00
DISCOUNTS	0.00	TAX	0.00
		PAYMENTS	0.00
AMOUNT DUE			4,000.00

3. Select **Exact** to process the payment.

4. Validate the receipt.

Transaction:	DELHI-DELHI-1-117	Date:	25-01-20
Terminal:	DELHI-1	Time:	19:28
Employee:	000160		
Customer No.:			
Customer name:			
Company PAN No	AAACC1478C		
GSTIN No.	07AEVPG9380C1Z7		
TIN No.:			
Item name	Qty	Price	Total
-----	-----	-----	-----
Enclosure	1.00	4,000.00	4,000.00
HSN/SAC	25163209		
-----	-----	-----	-----
Subtotal			4,000.00
Total			4,000.00
Cash			4,000.00
=====			
<B: ELHIHI-11_000022>			
Tax components	Tax code	Tax per	Tax amount

5. Validate the sales invoice in Headquarters:

- a. Go to **Retail and Commerce > Retail and Commerce IT > Data distribution.**
- b. Run job P-0001 (Channel transactions).
- c. Close the page.

6. Post the statement:

- a. Go to **Retail and Commerce > Channels > Stores > Open statements.**
- b. Create a statement.
- c. Select **Calculate statement** and then **Post statement.**

7. Validate the voucher transactions:

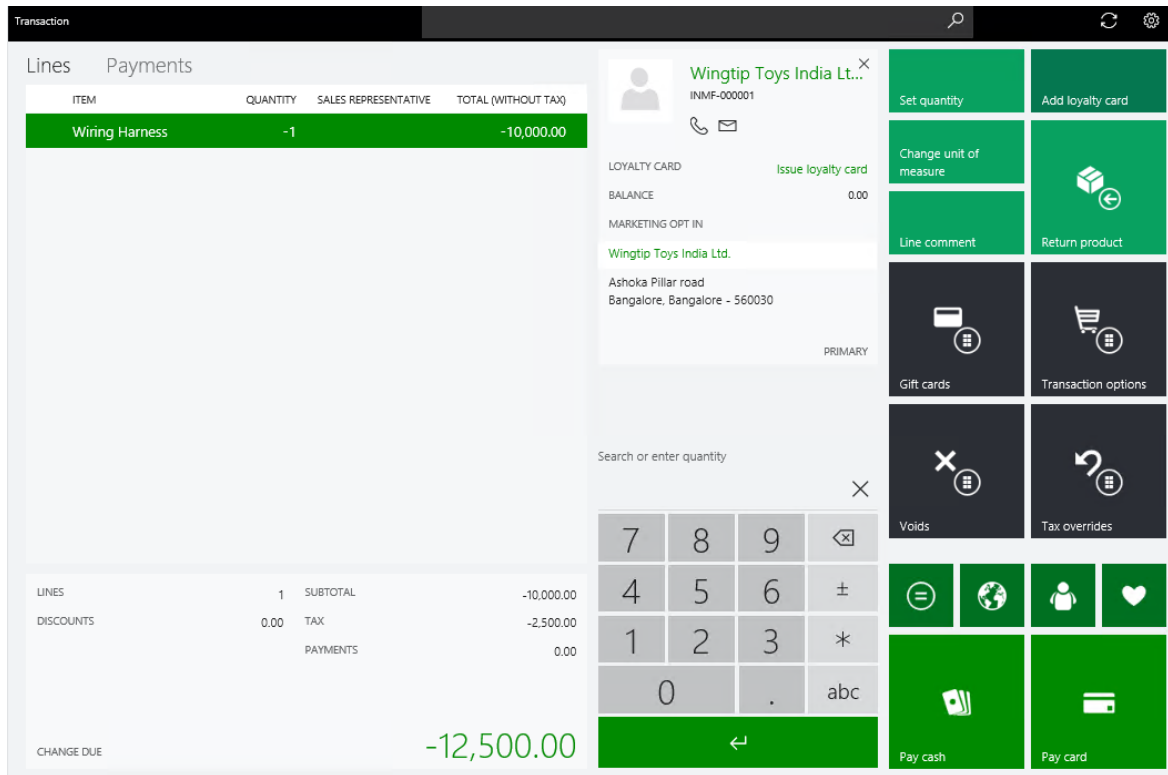
- a. Go to **Retail and Commerce > Customers > All sales orders.**
- b. Select the sales invoice.
- c. On the Action Pane, on the **Invoice** tab, select **Invoice journals.**
- d. Select **Voucher.**

LEDGER ACCOUNT NAME	DEBIT AMOUNT (RS.)	CREDIT AMOUNT (RS.)
Customer account	12,000.00	
Sales - Finished Goods		12,000.00

- e. Select **Tax document.**
- f. Verify that the **Exempt** option is set to **Yes.**

**Scenario 5: Return the transaction that has GST:**

1. Sign in to the POS.
2. Select **Show journal**.
3. Select the transaction, and then select **Return**.
4. Select **Select all** and then **Return**.
5. Verify that the GST calculation is done correctly, based on the selected original transactions that must be returned.



6. Select **Exact**.
7. Validate the receipt.

Transaction:	DELHI-DELHI-1-124	Date:	25-01-20
Terminal:	DELHI-1	Time:	20:00
Employee:	000160		
Customer No.:	INMF-000001		
Customer name:	Wingtip Toys India Ltd.		
Company PAN No	AAACC1478C		
GSTIN No.	07AEVPG9380C1Z7		
TIN No.:			
Item name	Qty	Price	Total
-----			
Wiring Harness	-1.00	10,000.00	-10,000.00
HSN/SAC	70119010		
-----			
Subtotal			-10,000.00
Tax			-2,500.00
Total			-12,500.00
change back (Cash)			-12,500.00
=====			
<B: ELHIHI-12_000006>			
Tax components	Tax code	Tax per	Tax amount
IGST	AA	25.00	-2,500.00

8. Validate the sales invoice in Headquarters:

- a. Go to **Retail and Commerce > Retail and Commerce IT > Data distribution.**
- b. Run job **P-0001 (Channel transactions).**
- c. Close the page.

9. Post the statement:

- a. Go to **Retail and Commerce > Channels > Stores > Open statements.**
- b. Create a statement.
- c. Select **Calculate statement** and then **Post statement.**

10. Validate the voucher transactions:

- a. Go to **Retail and Commerce > Customers > All sales orders.**
- b. Select the sales invoice.
- c. On the Action Pane, on the **Invoice** tab, select **Invoice journals.**
- d. Select **Voucher.**

LEDGER ACCOUNT NAME	DEBIT AMOUNT (RS.)	CREDIT AMOUNT (RS.)
Customer account		12,500.00
IGST payable account	2,500.00	
Sales account	10,000.00	

e. Select **Tax document.**

f. Verify that the return receipt number is updated as the transaction ID.

# Update credit notes with references to original invoices

## NOTE

This functionality is available with Application update 10.0.3 and later.

In order to be correctly reflected in the GSTR reporting, sales credit notes should contain references to original sales invoices. When store transactions are posted through statements, it is not always possible to establish this reference for return transactions. You can use the **Update credit notes with references to original invoices** procedure to update the **Original GST transaction ID** link in credit notes so that the link correctly references the related original sales invoice. The procedure is located on the **Retail and Commerce > Retail and Commerce IT > POS posting** menu.

It is also recommended that you enable the **Do not aggregate returns** parameter on the **Commerce parameters** page. In this case, each return transaction will be posted as a separate sale order when posting a statement. This option is only available if the transaction aggregation is enabled.

# Manage customer registration numbers from POS

## NOTE

This functionality is available in Application update 10.0.6 and later.

You can specify customer registration numbers, such as GSTIN, VAT number (TIN), and PAN number, when creating or editing a customer master record and a customer address record in POS. The customer registration numbers may be printed in receipts or used for searching customers in POS.

## Configure printing customer registration numbers in receipts

To enable printing customer registration numbers in receipts, follow the procedure outlined in the [Configure language texts and custom fields](#) section. Add language texts and custom fields with the following names:

- TAXREGISTRATIONGST\_IN for the GST registration number;
- TAXREGISTRATIONTIN\_IN for the VAT registration number;
- TAXREGISTRATIONPAN\_IN for the PAN number.

Add the custom fields to receipt profiles.

## Enable searching customers by tax registration numbers in POS

To enable searching customers by tax registration numbers in POS, on the **POS search criteria** tab of the **Commerce parameters** page, add a record on the **Customer search criteria** fast-tab and select **Tax registration number** in the **Customer search criteria** drop-down list. Select the **Display as shortcut** checkbox while keeping the **Can be refined** checkbox clear. Run the 1110 job on the **Distribution schedules** page.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Deployment guidelines for cash registers for India

2/18/2021 • 11 minutes to read • [Edit Online](#)

This topic is a deployment guide that shows how to enable the requirements for Goods and Services Tax (GST) in the Dynamics 365 Commerce app's localization for India. For more information about the localization for India, see [Goods and Services Tax \(GST\) integration for cash registers for India](#).

This sample is part of the Retail software development kit (SDK). For information about how to install and use the SDK, see the [Retail software development kit \(SDK\) architecture](#).

This sample consists of extensions for the Commerce runtime (CRT). To run this sample, you must modify and build the CRT projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Microsoft Visual Studio Online (VSO), where no files have been changed yet.

## NOTE

Some steps in the procedures in this topic differ, depending on the version of the app that you're using. For more information, see [What's new or changed in Dynamics 365 Retail](#).

## Prerequisites

Make sure that the Visual C++ Redistributable Packages are present on the machine that you're running Goods and Services Tax (GST) calculations on. For Cloud POS, and for Modern POS in online mode, this machine is Retail Server (Retail Server is known as Commerce Scale Unit in Commerce 10.0.8 and above). For Modern POS in offline mode, it's the Modern POS machine itself. To get the packages, see [Download the Visual C++ Redistributable Packages](#).

## Development environment

Follow these steps to set up a development environment so that you can test and extend the sample.

### The CRT extension components

The CRT extension components are included in the CRT samples. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`. You can find this solution under `RetailSdk\SampleExtensions\CommerceRuntime`.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

1. Find the `Runtime.Extensions.GenericTaxEngine` project, and build it.

2. Find the following files:

- In the `Extensions.GenericTaxEngine\bin\Debug` folder:
  - `Contoso.Commerce.Runtime.Extensions.GenericTaxEngine.dll`
- In the `Reference\Newtonsoft.Json\9.0.0.0` folder:

- Newtonsoft.Json.dll
- In the **Reference\TaxEngine** folder:
  - Microsoft.Dynamics365.Tax.Core.dll
  - Microsoft.Dynamics365.Tax.DataAccessor.dll
  - Microsoft.Dynamics365.Tax.DataAccessFramework.dll
  - Microsoft.Dynamics365.Tax.DataModel.dll
  - Microsoft.Dynamics365.Tax.Metadata.dll
  - Microsoft.Dynamics365.LocalizationFramework.dll
  - Microsoft.Dynamics365.LocalizationFrameworkCore.dll
  - Microsoft.Dynamics365.ElectronicReportingMapping.dll
  - Microsoft.Dynamics365.XppSupportLayer.dll
- Find the following folders in the **Reference\Z3** folder:
  - x86
  - x64

3. Copy the 11 assembly files, and both x64 and x86 folders to the CRT extensions folder:

- **Retail Server:** Copy the assemblies to the `\bin\ext` folder under the Microsoft Internet Information Services (IIS) Retail server site location.
- **Local CRT on Modern POS:** Copy the assemblies to the `\ext` folder under the local CRT client broker location.

4. Find the extensions configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOffline.Ext.config`, and it's under the local CRT client broker location.

5. Register the CRT change in the extensions configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.Extensions.GenericTaxEngine" />
```

#### **WARNING**

Do not edit the `commerceruntime.config` and `CommerceRuntime.MPOSOffline.config` files. These files aren't intended for any customizations.

### **The extension components**

1. Open `web.config` in the root folder under the IIS Retail Server site location. Note that Retail Server is known as Commerce Scale Unit in Commerce 10.0.8 and above.
2. Register the extensions in the `extensionComposition` section of the configuration file.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

**NOTE**

This step doesn't apply to this version.

**The ClientBroker extension components**

1. Open `RetailProxy.MPOSOffline.ext.config` under the local CRT client broker location.
2. Register the Proxy extensions in the `extensionComposition` section of the configuration file.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

**NOTE**

This step doesn't apply to this version.

**The Modern POS extension components**

Enable the Tax Registration Id extension.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

**NOTE**

This step doesn't apply to this version.

**The Cloud POS extension components**

Enable the Tax Registration Id extension.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

**NOTE**

This step doesn't apply to this version.

**Set up required parameters in Headquarters**

For more information, see [Goods and Services Tax \(GST\) integration for cash registers for India](#).

## Production environment

Follow these steps to create deployable packages that contain components, and to apply the packages in a

production environment.

1. In the `commerceruntime.ext.config` and `CommerceRuntime.MPOSOffline.Ext.config` configuration files under the `RetailSdk\Assets` folder, add the following lines to the `composition` section.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

```
<add source="assembly" value="Contoso.Commerce.Runtime.Extensions.GenericTaxEngine" />
```

2. In the `RetailProxy.MPOSOffline.ext.config` configuration file under the `RetailSdk\Assets` folder, add the following lines to the `composition` section.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

**NOTE**

This step doesn't apply to this version.

3. Update the web configuration file. In the `RetailSDK\Packages\RetailServer\Code\web.config` file, add the following lines to the `extensionComposition` section.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

**NOTE**

This step doesn't apply to this version.

4. In the `Customization.settings` package customization configuration file under the `RetailSdk\BuildTools` folder, add the following lines to the `ItemGroup` section to include the CRT extensions in deployable packages.

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

```

<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.Extensions.GenericTaxEngine.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.Tax.Core.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.Tax.Metadata.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.Tax.DataAccessor.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.Tax.DataAccessFramework.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.Tax.DataModel.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.LocalizationFramework.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.LocalizationFrameworkCore.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.ElectronicReportingMapping.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\TaxEngine\Microsoft.Dynamics365.XppSupportLayer.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Newtonsoft.Json\9.0.0\Newtonsoft.Json.dll" />

```

5. Modify the following files to include the Z3 libraries in deployable packages:

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)
  
- Packages\ModernPOS.Sdk\Sdk.ModernPOSSetup.csproj
- Packages\ModernPOSOffline.Sdk\Sdk.ModernPOSSetupOffline.csproj
- Packages\RetailServer\Sdk.RetailServerSetup.proj

Add the following lines to the **ItemGroup** section.

```

<_bin_ext_Z3_x86_File Include="..\..\References\Z3\x86\*.*)" />
<_bin_ext_Z3_x64_File Include="..\..\References\Z3\x64\*.*)" />

```

For **Sdk.ModernPOSSetup.csproj** and **Sdk.ModernPOSSetupOffline.csproj** also add the following lines to the **<Target Name="CopyPackageFiles">** section.

```

<Copy SourceFiles="@(_bin_ext_Z3_x86_File)"
DestinationFolder="$(OutputPath)content.folder\CustomizedFiles\ClientBroker\ext\x86"
SkipUnchangedFiles="true" />
<Copy SourceFiles="@(_bin_ext_Z3_x64_File)"
DestinationFolder="$(OutputPath)content.folder\CustomizedFiles\ClientBroker\ext\x64"
SkipUnchangedFiles="true" />

```

For **Sdk.RetailServerSetup.proj** also add the following lines to the **<Target Name="CopyPackageFiles">** section.

```
<Copy SourceFiles="@(_bin_ext_Z3_x86_File)"
DestinationFolder="$(OutputPath)content.folder\RetailServer\Code\bin\ext\x86"
SkipUnchangedFiles="true" />
<Copy SourceFiles="@(_bin_ext_Z3_x64_File)"
DestinationFolder="$(OutputPath)content.folder\RetailServer\Code\bin\ext\x64"
SkipUnchangedFiles="true" />
```

6. Enable the Tax Registration Id POS extension:

- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 8.1.3 and later](#)
- [Retail 10.0 and later](#)
- [Retail 10.0.6 and later](#)

**NOTE**

This step doesn't apply to this version.

7. Run **msbuild** for the whole Retail SDK to create deployable packages.

8. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create retail deployable packages](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Support for upgrade and N-1 for India

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the steps needed to set up and use Phased Rollout (N-1) commerce components for India. The upgrade procedure and the workflow for N-1 are basically the same as for a general Dynamics 365 Commerce environment. For general information about N-1 installation and usage, see [Upgrade and N-1 support for Retail](#).

In addition, the following steps are important for upgrade:

- Both Commerce Headquarters and AX 2012 Retail components must support Goods and Services Tax (GST) calculation. See the [Prerequisites](#) section for details.
- Downgrade of the GST configuration is required to use the configuration prepared for Commerce at the AX 2012 channel side.
- Distribution schedule includes additional upload and download jobs required to synchronize GST configuration and tax calculation results between Headquarters and AX 2012 channels.

## Prerequisites

- Update AX 2013 R3 retail components to GST Update 2, [KB4058327](#), or higher. For details, see [India GST Update 2 - Release Notes](#).
- Set up basic N-1 environment. For more information, refer to [Phased Rollout \(N-1\) installation, configuration, and cutover guide](#).
- Set up Commerce GST for India in Headquarters according to the [Goods and Services Tax \(GST\) integration for cash registers for India](#) guide.

## Periodic procedure to downgrade tax configuration

GST configuration data differs between AX 2012 and Commerce versions. A special periodic procedure should be run to transform data. To run this procedure, do the following:

1. Sign in to Headquarters, and go to **Retail and Commerce > Retail and Commerce IT > Process tax configuration from N-1**.
2. Click **OK**.

Each time tax configuration changes are made and finalized the above operation should be run before sending the data to the AX 2012 channel.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Fiscal printer integration sample for Italy

2/18/2021 • 11 minutes to read • [Edit Online](#)

## Introduction

The Commerce functionality for Italy includes a sample integration of the point of sale (POS) with a fiscal printer. The sample extends the [fiscal integration functionality](#) so that it works with [Epson FP-90III Series](#) printers from Epson, and it enables communication with a fiscal printer in the web server mode via the EpsonFPMate web-service using Fiscal ePOS-Print API. The sample supports the Registratore Telematico (RT) mode only. The sample is provided in the form of source code and is part of the Retail software development kit (SDK).

Microsoft doesn't release any hardware, software, or documentation from Epson. For information about how to get the fiscal printer and operate it, contact [Epson Italia S.p.A.](#)

## Scenarios

The following scenarios are covered by the fiscal printer integration sample for Italy:

- Sales scenarios:
  - Print a fiscal receipt for cash-and-carry sales and returns.
  - Capture a response from the fiscal printer, and store it in the channel database.
  - Taxes:
    - Map to the fiscal printer's tax codes (departments).
    - Transfer mapped tax data to the fiscal printer.
    - Print taxes on a fiscal receipt.
  - Payments:
    - Map to the fiscal printer's methods of payment.
    - Print payments on a fiscal receipt.
    - Print change information.
  - Print line discounts.
  - Gift cards:
    - Exclude an issued/re-charged gift card line from a fiscal receipt for a sale.
    - Print a payment that uses a gift card as a regular method of payment.
  - Print fiscal receipts for customer order operations:
    - A fiscal receipt isn't printed for a customer order deposit.
    - Print a fiscal receipt for carry-out lines of a hybrid customer order.
    - Print a fiscal receipt for the pickup operation for a customer order.
    - Print a fiscal receipt for a return order.
  - Print a bar code for the receipt number on a fiscal receipt.
  - Print the [customer information](#) that is specified for a sales transaction on a fiscal receipt. An example of this information is the customer's lottery code.
- End of day statements (fiscal X and fiscal Z reports).
- Error handling, such as the following options:
  - Retry fiscal registration if a retry is possible, such as if the fiscal printer isn't connected, isn't ready or isn't responding, the printer is out of paper, or there is a paper jam.
  - Postpone fiscal registration.
  - Skip fiscal registration, or mark the transaction as registered, and include info codes to capture the reason for the failure and additional information.
  - Check the availability of the fiscal printer before a new sales transaction is opened or a sales transaction is finalized.

### Default data mapping

The following default data mapping is included in the fiscal document provider configuration that is provided as part of the fiscal integration sample:

- Value-added tax (VAT) rates mapping:  
*1 : 22.00 ; 2 : 10.00 ; 3 : 4.00 ; 4 : 0.00*
- Tender type mapping:  
*1 : 0 ; 2 : 1 ; 3 : 2 ; 4 : 2 ; 5 : 0 ; 6 : 0 ; 7 : 0 ; 8 : 2 ; 9 : 0 ; 10 : 2 ; 11 : 1*

### Gift cards

The fiscal printer integration sample implements the following rules that are related to gift cards:

- Exclude sales lines that are related to the *Issue gift card* and *Add to gift card* operations from the fiscal receipt.



- Don't print a fiscal receipt if it consists only of gift card lines.
- Deduct the total amount of gift cards that are issued or re-charged in a transaction from payment lines of the fiscal receipt.
- Save calculated adjustments of payment lines in the channel database with a reference to a corresponding fiscal transaction.
- Payment by gift card is considered a regular payment.

#### Customer deposits and customer order deposits

The fiscal printer integration sample implements the following rules that are related to customer deposits and customer order deposits:

- Don't print a fiscal receipt if a transaction is a customer deposit.
- Don't print a fiscal receipt if a transaction contains only a customer order deposit or a customer order deposit refund.
- Print the amount of the previously paid deposit on a fiscal receipt for a customer order pickup operation.
- Deduct the customer order deposit amount from payment lines when a hybrid customer order is created.
- Save calculated adjustments of payment lines in the channel database with a reference to a fiscal transaction for a hybrid customer order.

#### Limitations of the sample

- The fiscal printer supports only scenarios where sales tax is included in the price. Therefore, the **Price include sales tax** option must be set to **Yes** for both stores and customers.
- Daily reports (fiscal X and fiscal Z) are printed by using the format that is embedded in the fiscal printer's firmware.
- The fiscal printer doesn't support mixed transactions. The **Prohibit mixing sales and returns in one receipt** option should be set to **Yes** in POS functionality profiles.
- The sample supports integration only with a fiscal printer that is working in the RT (Registratore Telematico) mode.

## Set up Commerce for Italy

### Configure fiscal integration

Complete the fiscal integration setup steps as described in [Set up the fiscal integration for Commerce channels](#):

- [Set up a fiscal registration process](#). Note also the settings for the fiscal registration process that are [specific to this fiscal printer integration sample](#).
- [Set up fiscal texts for discounts](#).
- [Set error handling settings](#).
- [Set up fiscal X/Z reports from the POS](#).
- [Enable manual execution of postponed fiscal registration](#).
- [Set up the functionality for management of customer information in POS](#)

### Enable extensions

#### Commerce runtime extension components

The Commerce runtime extension components are included in the Retail SDK. To complete the following procedures, open the CRT solution, **CommerceRuntimeSamples.sln**, under **RetailSdk\SampleExtensions\CommerceRuntime**.

1. Find the **Runtime.Extensions.DocumentProvider.EpsonFP90IIISample** project, and build it.
2. In the **Extensions.DocumentProvider.EpsonFP90IIISample\bin\Debug** folder, find the **Contoso.Commerce.Runtime.DocumentProvider.EpsonFP90IIISample.dll** assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit**: Copy the assembly to the **\bin\ext** folder under the Microsoft Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS**: Copy the assembly to the **\ext** folder under the local CRT client broker location.
4. Find the extensions configuration file for CRT:
  - **Commerce Scale Uni**: The file is named **commercerruntime.ext.config**, and it's in the **bin\ext** folder under the IIS Commerce Scale Uni site location.
  - **Local CRT on Modern POS**: The file is named **CommerceRuntime.MPOSOOffline.Ext.config**, and it's under the local CRT client broker location.
5. Register the CRT change in the extensions configuration file. Add **source="assembly"** and **value="Contoso.Commerce.Runtime.DocumentProvider.EpsonFP90IIISample"**.
6. Restart the Commerce Scale Unit:
  - **Commerce Scale Uni**: Restart the Commerce Scale Uni site from IIS Manager.
  - **Client broker**: End the **dllhost.exe** process in Task Manager, and then restart Modern POS.

#### Hardware station extension components

The Hardware station extension components are included in the Retail SDK. To complete the following procedures, open the Hardware Station solution, **HardwareStationSamples.sln**, under **RetailSdk\SampleExtensions\HardwareStation**.

1. Find the **HardwareStation.Extensions.EpsonFP90IIIFiscalDeviceSample** project, and build it.

- In the `Extensions.EpsonFP90IIIFiscalDeviceSample\bin\Debug` folder, find the `Contoso.Commerce.HardwareStation.EpsonFP90IIIFiscalDeviceSample.dll` assembly file.
- Copy the files to a deployed Hardware station machine:
  - Remote Hardware station:** Copy the files to the `bin` folder under the IIS Hardware station site location.
  - Local Hardware station:** Copy the files to the Modern POS client broker location.
- Find the configuration file for the Hardware station's extensions. The file is named `HardwareStation.Extension.config`:
  - Remote Hardware station:** The file is located under the IIS Hardware station site location.
  - Local Hardware station:** The file is located under the Modern POS client broker location.
- Add the following section to the `composition` section of the config file.

```
<add source="assembly"
value="Contoso.Commerce.HardwareStation.Extension.EpsonFP90IIIFiscalDeviceSample" />
```

- Restart the Hardware station service:
  - Remote Hardware station:** Restart the Hardware station site from IIS Manager.
  - Local Hardware station:** End the `dllhost.exe` process in Task Manager, and then restart Modern POS.

### Set up the registration process

To enable the registration process, follow these steps to set up Headquarters. For more details, see [Set up a fiscal registration process](#).

- Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Fiscal Connectors**. Import the configuration from `RetailSdk\SampleExtensions\HardwareStation\Extension.EpsonFP90IIIFiscalDeviceSample\Configuration\ConnectorEpsonFP90IIISar`
- Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Fiscal Document providers**. Import the configuration from `RetailSdk\SampleExtensions\CommerceRuntime\Extension.DocumentProvider.EpsonFP90IIISample\Configuration\DocumentProvide`
- Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Connector Technical profiles**. Create a new profile, and select the loaded connector from the earlier step. Update the connection settings if an update is required.
- Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Connector Functional profiles**. Create a new profile, and select the loaded connector and document provider from the earlier steps. Update data mapping settings if an update is required.
- Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Connector Functional group**. Create a new group, and select the connector functional profile from the earlier step.
- Go to **Retail and Commerce > Channel Setup > Fiscal Integration > Registration process**. Create a new process, and select the connector functional group from the earlier step.
- Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. Open the functionality profile that is linked to the store where the registration process should be activated. On the **Fiscal registration process** FastTab, select the registration process that was created earlier.
- Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**. Open the hardware profile that is linked to the Hardware station that the fiscal printer will be connected to. On the **Fiscal peripherals** FastTab, select the connector technical profile.
- Open the distribution schedule (**Retail and Commerce > Retail and Commerce IT > Distribution schedule**), and select jobs **1070** and **1090** to transfer data to the channel database.

### Production environment

Follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

- Complete the steps that are described in the [Enable extensions](#) section earlier in this topic.
- Make the following changes in the package configuration files under the `RetailSdk\Assets` folder:

- In the `commercerruntime.ext.config` and `CommerceRuntime.MPOSOOffline.Ext.config` configuration files, add the following line to the `composition` section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.EpsonFP90IIISample" />
```

- In the `HardwareStation.Extension.config` configuration file, add the following line to the `composition` section.

```
<add source="assembly"
value="Contoso.Commerce.HardwareStation.Extension.EpsonFP90IIIFiscalDeviceSample" />
```

- Make the following changes in the `BuildTools\Customization.settings` package customization configuration file:
  - Add the following line to include the CRT extension in the deployable packages.

```
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.EpsonFP90IIISample.dll
"/>
```

- Add the following line to include the Hardware station extension in the deployable packages.

```
<ISV_HardwareStation_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.EpsonFP90IIIFiscalDeviceSample.
dll"/>
```

4. Start the MSBuild Command Prompt for Visual Studio utility, and run **msbuild** under the Retail SDK folder to create deployable packages.
5. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).

## Design of extensions

### Commerce runtime extension design

The purpose of the extension that is a fiscal document provider is to generate printer-specific documents and handle responses from the fiscal printer.

The Commerce runtime extension is **Runtime.Extensions.DocumentProvider.EpsonFP90IIISample**.

For more details about the design of the fiscal integration solution, see [Fiscal registration process and fiscal integration samples for fiscal devices](#).

#### Request handler

The **DocumentProviderEpsonFP90III** request handler is the entry point for the request to generate documents from the fiscal printer.

The handler is inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the connector document provider name that is specified in Headquarters.

The connector supports the following requests:

- **GetFiscalDocumentDocumentProviderRequest** – This request contains information about what document should be generated. It returns a printer-specific document that should be registered in the fiscal printer.
- **GetSupportedRegistrableEventsDocumentProviderRequest** – This request returns the list of events to subscribe to. Currently, the following events are supported: sales, printing X report, and printing Z report.

#### Configuration

The configuration file is located in the **Configuration** folder of the extension project. The purpose of the file is to enable settings for the document provider to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added:

- VAT codes mapping
- VAT rates mapping
- Tender type mapping
- Barcode type for receipt number
- Deposit payment type

### Hardware station extension design

The purpose of the extension that is a fiscal connector is to communicate with the fiscal printer.

The Hardware station extension is **HardwareStation.Extension.EpsonFP90IIIFiscalDeviceSample**. This extension uses the HTTP protocol to submit documents that the Commerce runtime extension generates to the fiscal printer. It also handles the responses that are received from the fiscal printer.

#### Request handler

The **EpsonFP90IIISample** request handler is the entry point for handling request to the fiscal peripheral device.

The handler is inherited from the **INamedRequestHandler** interface. The **HandlerName** method is responsible for returning the name of the handler. The handler name should match the fiscal connector name that is specified in Headquarters.

The connector supports the following requests:

- **SubmitDocumentFiscalDeviceRequest** – This request sends documents to printers and returns the response from the fiscal printer.
- **IsReadyFiscalDeviceRequest** – This request is used for a health check of the device.
- **InitializeFiscalDeviceRequest** – This request is used for printer initialization.

#### Configuration

The configuration file is located in the **Configuration** folder of the extension project. The purpose of the file is to enable settings for the connector to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added:

- **Endpoint address** – The URL of the printer.

- **Date and time synchronization** – This setting specifies whether the date and time of the printer must be synced with the connected Hardware station.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customer information management for Italy

2/18/2021 • 8 minutes to read • [Edit Online](#)

## Introduction

This topic describes how you can handle customer information, such as the customer's lottery code, in the Commerce point of sale (POS) for Italy.

You can specify the customer information, such as the fiscal code or lottery code, when you create or edit a customer master record in POS. You can also specify the lottery code for a sales transaction by copying it from the transaction customer or entering it manually. The lottery code can then be printed on both regular and fiscal receipts, and it can be used for the national lottery. Personal fiscal codes can also be used to search for a customer in POS.

### NOTE

This functionality is available in version 10.0.7 and later.

## Setup

You must complete the following configuration to use this functionality:

- Set up a registration type for the lottery code.
- Add the **Add customer information** operation to screen layouts.
- Activate the inquiry for customer information.
- Set up receipt formats.
- Add a customer search criterion.
- Configure channel components.

### Set up a registration type for the lottery code

Before lottery codes can be specified in POS, you must create an appropriate registration type for the lottery code and link it to the **Lottery code** registration category. For more information about how to work with registration types and registration IDs, see [Registration IDs](#).

### WARNING

If a registration type isn't created or isn't linked to the **Lottery code** registration category, an error will be generated in POS when the lottery code is filled in for a customer address.

### Add the Add customer information operation to screen layouts

The **Add customer information** operation can be used to add customer information, such as the lottery code, to a sales transaction. This information can be copied from the customer that is specified for the transaction, or it can be manually entered.

On the **Button grids** page, select the button grid where the operation should appear, and open the Button grid designer. Add a new button, and then, in the **Action** field, select **Add customer information**. For more information about how to work with screen layouts and button grids, see [Screen layouts for the point of sale \(POS\)](#).

## Activate the inquiry for customer information

If the customer information isn't specified for a sales transaction, an inquiry for that information can be triggered automatically after the transaction is finalized. This approach is an alternative to the **Add customer information** operation.

To activate the inquiry for customer information, set the **Enable inquiry of customer information in sales transactions** option to **Yes** in the **Tax parameters** section on the **Functions** FastTab of the **POS functionality profiles** page.

## Set up receipt formats

You can configure receipt formats so that the customer's fiscal code and lottery code are printed on receipts.

### NOTE

The default company of the user who creates the receipt setup should be the same legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

On the **Language text** page, on the **POS** tab, add the following records for the labels of the custom fields for receipt formats. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the following table are just examples. You can change them to meet your requirements. However, the **Text ID** values that you use must be unique, and they must be equal to or more than 900001.

LANGUAGE ID	TEXT ID	TEXT
en-US	900001	Lottery code
en-US	900002	Fiscal code

On the **Custom fields** page, add the following records for the custom fields for receipt formats. Note that the **Caption text ID** values must correspond to the **Text ID** values that you specified on the **Language text** page.

NAME	TYPE	CAPTION TEXT ID
FISCALCUSTOMER_LOTTERYCODE_IT	Receipt	900001
CUSTOMER_FISCALCODE_IT	Receipt	900002

In the Receipt format designer, add the custom fields to the appropriate receipt section for every receipt format that is required. For more information about how to work with receipt formats, see [Receipt templates and printing](#).

## Add a customer search criterion

You can add a customer search criterion so that customers can be searched for in POS by their fiscal codes.

On the **Commerce parameters** page, on the **POS search criteria** tab, add a new customer search criterion. In the **Customer search criteria** field, select **Tax registration number**. Select the **Display as shortcut** check box, but leave the **Can be refined** check box cleared. Then, on the **Distribution schedules** page, run the **1110** job.

## Configure channel components

To make the functionality that is specific to Italy available, you must configure extensions for commerce channel components. For more information, see the [Deployment guidelines](#) section later in this topic.

# Example scenarios

The following example scenarios show how to work with customer information in POS for Italy.

## Scenario 1: Make a sale to an anonymous customer

1. Sign in to POS.
2. Add items to the cart.
3. Select **Add customer information**, and then select **Enter manually**.
4. Enter the customer's lottery code, and then select **OK**.
5. Register payments for the transaction, and then finalize the transaction.
6. Verify that the printed receipt contains the customer's lottery code.

## Scenario 2: Make a sale to a new named customer

1. Sign in to POS.
2. Add items to the cart.
3. Select **Add customer**, and then select **New**.
4. Specify the new customer's attributes. In the **Fiscal code** field, enter the customer's fiscal code.
5. Select **Create a new address**. Then specify the new customer's contact information and an address.
6. In the **Lottery code** field, enter the customer's lottery code.
7. Save the customer record and the customer address record, and add the customer to the transaction.
8. Register payments for the transaction, and then finalize the transaction.
9. Because the inquiry for customer information has been activated, but customer information hasn't been added to the transaction, the **Enter customer information** dialog box is opened. Select **Yes**, and then select **Copy from transaction customer**.
10. Verify the customer's lottery code, and then select **OK**.
11. Verify that the printed receipt contains the customer's lottery code.

### NOTE

If you must specify a different customer for the transaction, you must clear the customer information and then copy it again after the new customer is added.

## Scenario 3: Change the customer information for a sale to a named customer

1. Sign in to POS.
2. Add items to the cart.
3. Select **Add customer**, and then select a customer account to add it to the transaction.
4. Select **Add customer information**, and then select **Copy from transaction customer**.
5. Verify the customer's lottery code, and then select **OK**.
6. Select **Add customer information**, and then select **Clear** to clear the customer information from the transaction.
7. Select **Add customer information**, and then select **Enter manually**.
8. Specify the customer's lottery code, and then select **OK**.
9. Register payments for the transaction, and then finalize the transaction.
10. Verify that the printed receipt contains the customer's lottery code.

# Deployment guidelines

This section provides deployment guidance for enabling customer information management in the localization of Commerce for Italy.

## NOTE

Some steps in these procedures vary, depending on the version of Commerce that you're using. For more information, see [What's new or changed in Dynamics 365 for Retail](#).

If you want to enable the integration of POS with fiscal printers for Italy, and specifically if you want to print customer lottery codes on fiscal receipts, you must deploy the [fiscal printer integration sample for Italy](#).

## Update customizations

Follow these steps to update customizations.

- [Retail 10.0.7 and later](#)
- [Retail 10.0.12 and later](#)

If any of your customizations include request handlers for the `SaveCartRequest` or `CreateSalesOrderServiceRequest` requests:

1. Find the request handler for `SaveCartRequest`.
2. Find the line of code that runs the original request handler.
3. Add the following lines before calling the original request handler:

```
using Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdItaly.Services;  
...  
new TaxRegistrationIdFiscalCustomerService().Execute(request);
```

4. Find the request handler for `CreateSalesOrderServiceRequest`.
5. Find the line of code that runs the original request handler.
6. Replace it with the following code:

```
using Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdItaly.Services;  
...  
return new TaxRegistrationIdFiscalCustomerService().Execute(request);
```

## Update a development environment

Follow these steps to update a development environment.

### CRT extension components

1. Find the extension configuration file for the Commerce runtime (CRT):
  - **Commerce Scale Unit:** Find the `CommerceRuntime.Ext.config` file in the `bin\ext` folder under the Microsoft Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Find the `CommerceRuntime.MPOSOOffline.Ext.config` file under the local CRT client broker location.
2. Register the CRT extension in the extension configuration file.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdItaly" />
```



## WARNING

Do **not** edit the `CommerceRuntime.config` and `CommerceRuntime.MPOSOffline.config` files. These files aren't intended for any customizations.

### Modern POS extension components

Follow these steps to make the `TaxRegistrationId.IT` extension available.

1. Open the solution at `RetailSdk\POS\ModernPOS.sln`.
2. In `POS.Extensions\extensions.json`, turn on the extension.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/TaxRegistrationId.IT"
    }
  ]
}
```

3. Build the solution.
4. Open Modern POS, and test the functionality.

### Cloud POS extension components

Follow these steps to make the `TaxRegistrationId.IT` extension available.

1. Open the solution at `RetailSdk\POS\CloudPOS.sln`.
2. In `POS.Extensions\extensions.json`, turn on the extension.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/TaxRegistrationId.IT"
    }
  ]
}
```

3. Build the solution.
4. Open Cloud POS, and test the functionality.

### Update a production environment

Follow these steps to create deployable packages that contain Commerce components, and to apply the packages in a production environment.

1. In the `CommerceRuntime.Ext.config` and `CommerceRuntime.MPOSOffline.Ext.config` configuration files under the `RetailSdk\Assets` folder, add the following lines to the `composition` section.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.TaxRegistrationIdItaly" />
```

2. Turn on the `TaxRegistrationId.IT` POS extension.

```
{
  "extensionPackages": [
    {
      "baseUri": "Microsoft/TaxRegistrationId.IT"
    }
  ]
}
```

3. Run **msbuild** for the whole Retail software development kit (SDK) to create deployable packages.
4. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Retail SDK packaging](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Global CFDI electronic invoices for Mexico

2/18/2021 • 5 minutes to read • [Edit Online](#)

The Commerce functionality for Mexico supports the Comprobantes fiscales digitales por internet (CFDI) format for Mexican companies. For more information about CFDI electronic invoices, see [Electronic invoices \(CFDI\)](#).

When a company closes the daily process, it must issue a Global CFDI document to consolidate all receipts that were issued to the final consumers. This document includes the following information for each transaction that is registered during the period:

- A receipt number
- A corresponding amount

## Processing Global CFDI documents

The Global CFDI functionality lets you perform the following tasks:

- Create an electronic invoice, in Global CFDI format (layout 3.3), that is based on the posted commerce statement. For more information about the layout, see [CFDI layout version 3.3](#).
- For each electronic invoice generate a file in the .pdf or .xml format, and send it to the customer as an email attachment. After the Global CFDI electronic invoices are generated, they are verified and certified by a digital signature service provider (PAC) in the same way as other CFDI documents. For more information, see [Electronic invoices \(CFDI\)](#) and [Inquire and print an electronic invoice](#).

To generate and submit a Global CFDI electronic invoice, follow these steps.

1. As a preparatory step, on the **Retail** tab of the **Electronic invoice parameters** page (**Organization administration > Setup > EInvoice > Electronic invoice parameters**), you must specify the default parameters of the Global CFDI format.
2. Close the shift at the point of sale (POS).
3. Run the P-job in the distribution schedule to transfer transactions from the channel database to Headquarters.
4. Calculate and post a statement by following the steps in [Create, calculate, and post a statement for a retail store](#).
5. Run the **Post CFDI – Electronic invoices** periodic operation to create Global CFDI electronic invoices that are based on a posted statement. You can select a statement number for this periodic operation. If you don't select a statement number, the system creates Global CFDI electronic invoices for all posted statements that haven't yet been processed.

As a result of the **Post CFDI – Electronic invoices** periodic operation, two Global CFDI electronic invoices are created. One electronic invoice collects all receipts that are related to sales operations, and the other collects all receipts that are related to returns. For the electronic invoice that is related to returns, the **Return** attribute is set to **Yes**. You can view these electronic invoices on the **CFDI (electronic invoices)** page (**Retail and Commerce > Inquiries and reports > CFDI (electronic invoices)**).

All further workflows, such as communication with a service provider, generation of the .pdf and .xml files, and manual functions are the same as the workflows for CFDI Normal electronic invoices.

6. Run the **Export/import electronic invoice process** periodic operation to submit electronic invoices to

the PAC.

## Updates for the Global CFDI functionality

In Microsoft Dynamics 365 for Finance and Operations version 10.0.2 (May 2019), the Global CFDI functionality was extended to support new requirements that were introduced in the second revision of the *Global CFDI filling guide*. Starting in Microsoft Dynamics 365 for Finance and Operations version 10.0.2 (May 2019), you can perform the following tasks:

- At the customer's request, generate a separate CFDI Normal electronic invoice that is based on a sale or return operation that is registered at the POS.

In this case, the sale or return operation should be registered as a customer order. For more information about the functionality for customer orders, see [Customer orders in Modern POS \(MPOS\)](#).

- In a CFDI electronic invoice that is generated based on returns, specify universally unique identifiers (UUIDs) for CFDI electronic invoices that are related to the original sales operations.

Starting in Dynamics 365 for Finance and Operations version 10.0.2 (May 2019), the Global CFDI functionality also supports the following additional scenarios:

- [Item returns across multiple customer orders and invoices](#)
- Customer order returns that involve an exchange, when a customer invoice includes both lines that have positive amounts and lines that have negative amounts

### Showing related CFDI documents in a CFDI electronic invoice

To show the UUIDs of the original sales in CFDI electronic invoices for returns, turn on the **Specify related CFDI in returns** parameter on the **Retail** tab of the **Electronic invoice parameters** page (**Organization administration > Setup > EInvoice > Electronic invoice parameters**). After you turn on this parameter, both Global CFDI and CFDI Normal documents are generated that include the following information:

- A list of related electronic invoices.
- A status of **Draft**. (This status is a new status that was introduced in version 10.0.2). An electronic invoice is automatically set to this status if a UUID isn't specified for all the related electronic invoices. The **Export/import electronic invoice process** periodic procedure skips all electronic invoices that have a status of **Draft**.
- An XML file that is generated, where the **CFDI relacionados** section contains the list of related electronic invoices. You can view the list of related electronic invoices by selecting **Inquiries > Related e-invoices**. You can also manually add a new related electronic invoice on the same page.

The status of electronic invoices is automatically updated later, when all related electronic invoices get their UUIDs. To manually update the status of an electronic invoice, you can use the **Mark as ready** function. The status of the electronic invoice will be changed, and it will become available for export.

### Excluding customer orders from Global CFDI electronic invoices

To process customer orders as CFDI Normal electronic invoices in the same way that sales orders are processed in the **Accounts receivable** module, and to exclude those operations from the Global CFDI electronic invoices, turn on the **Exclude customer orders from CFDI Global** parameter on the **Retail** tab of the **Electronic invoice parameters** page (**Organization administration > Setup > EInvoice > Electronic invoice parameters**). After you turn on this parameter, a separate electronic invoice is created based on each sale or return operation that is posted via a customer order at the POS.

The following customer order operations can cause a CFDI Normal electronic invoice to be posted:

- Customer order pickup
- Customer order return

- Hybrid customer order (For more information about this scenario, see [Hybrid customer orders](#).)

When the **Exclude customer orders from CFDI Global** parameter is turned off, and no previous customer invoice was posted from the same customer order and processed as a separate electronic invoice (CFDI Normal), customer orders are included in Global CFDI electronic invoices. Otherwise, customer orders are processed as separate electronic invoices (CFDI Normal). Therefore, they are excluded from Global CFDI electronic invoices.

Customer order returns are processed in the same way (either CFDI Global or CFDI Normal) as the original sales operations.

### **Limitations**

Note the following limitations:

- All invoices from an original sales order are included in a return electronic invoice as the related CFDI documents.
- The scenario of a customer order return that involves an exchange is supported only for Global CFDI electronic invoices.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cash register functionality for Norway

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic provides an overview of the cash register functionality that is available for Norway in Dynamics 365 Commerce. It also provides guidelines for setting up the functionality. The functionality consists of the following parts:

- Common point-of-sale (POS) features that are available to customers in all countries or regions. Examples include an option that lets you prevent a copy of a receipt from being printed more than one time.
- Norway-specific features, such as digital signatures for sales transactions.

## Overview of cash register functionality for Norway

### Common POS features

To learn about POS features that are available to customers in all countries or regions, see [Help resources for Dynamics 365 Retail](#).

The following POS localization features that were previously implemented and made available to customers in all countries or regions can now be used specifically for Norway:

- **Print text fields on a receipt in a large font size.** You can use the **Font size** parameter in the Receipt format designer to specify that the large font size should be used for a field in the receipt format. (The large font size is approximately double the usual font size.) For example, you can use this parameter to print the "Copy" indicator on a copy of a receipt in large characters.
- **Register the printing of receipt copies in the POS audit event log.** You can use the **Audit** parameter in the POS functionality profile to enable copies of receipts to be printed and other POS audit events to be registered. The audit events are registered in the channel database and in Headquarters. You can view the audit events on the **Audit events** page.
- **Prevent a copy of a receipt from being printed more than one time.** When the **Audit** parameter in the POS functionality profile is enabled, the **Allow printing receipt copies** POS permission controls whether copies of receipts can be printed. There is also an option that lets you prevent a copy of a receipt from being printed more than one time.

Additionally, the following POS feature was implemented for Norway but made available to customers in all countries or regions:

- **Register additional events in the POS audit event log.** If the **Audit** parameter in the POS functionality profile is enabled, the following events are registered in the POS audit event log:
  - Price checks
  - Tax overrides
  - Corrections to line quantities
  - Clearing transactions from the channel database

### Norway-specific POS features

The following Norway-specific POS features are enabled when the **ISO code** parameter in the POS functionality profile is set to **No**.

#### Digital signing of sales transactions

Every sales transaction is digitally signed. The signature is created and recorded in the POS transaction journal at the same time that the transaction is finalized. The signature is also available in the journal that is exported for

audit purposes.

Only transactions for cash sales are signed. Here are some examples of transactions that are excluded from the signing process:

- Prepayments (customer account deposit)
- Prepayments for sales orders (customer order deposit)
- Issuing a gift card
- Non-sales transactions (float entry, tender removal, and so on)

The data that is signed is a text string that consists of the following data fields. The data fields are separated by semicolons.

1. Previous signature for the same POS (A zero [0] is used for the first transaction.)
2. Transaction date
3. Transaction time
4. Sequential signed transaction number
5. Transaction amount including tax
6. Transaction amount excluding tax

The digital signing process uses an RSA 1024-bit key that has a SHA-1 hash function (RSA-SHA1-1024). A certificate that is installed on Commerce Scale Unit is used for signing. The unique identifier of the certificate (footprint) is recorded together with the signature.

The signature is stored in the store database and the headquarters (HQ) database together with the transaction data. You can view the transaction signature, together with the transaction data that was used to generate it, on the **Fiscal transactions** FastTab of the **Store transactions** page.

#### **Receipts**

Receipts for Norway can include additional information that was implemented by using custom fields:

- **Receipt title** – You can add a field to a receipt format layout to identify the type of receipt. For example, a sales receipt will include the text "Sales receipt".
- **Signed transaction sequential number** – The sequential number of a signed transaction can appear on the receipt to associate a printed receipt with a digital signature in the database.
- **Receipt totals** – Custom fields for receipt totals exclude non-sales amounts from total transaction amounts. Non-sales amounts include amounts for the following operations:
  - Prepayments (customer account deposit)
  - Prepayments for sales orders (customer order deposit)
  - Issuing a gift card
  - Adding funds to a gift card

#### **X and Z reports**

The information that is included on X and Z reports is based on Norwegian requirements. For example, **Total cash sales** amounts include only amounts for cash sales transactions and exclude issue gift card operations and prepayments. Total cash sales are also listed per item group and payment method. In addition, cumulative **Grand total sales** and **Grand total returns** amounts are maintained and printed.

#### **SAF-T Cash Register audit file**

You can export the POS transaction journal in the predefined Standard Audit File - Tax (SAF-T) Cash Register format. The audit file includes information about the organization, relevant master data (such as item groups, items, and tax codes), cash sales transaction data together with signatures for those transactions, non-sales event data, and end-of-date report data.

The audit file can be exported for the following scenarios:

- Per store
- All stores
- Per terminal
- All terminals

You can also send a report from one legal entity on behalf of another legal entity. In this case, you must run the export from the operating legal entity and specify the reporting legal entity as the sender of the report.

The SAF-T Cash Register format is implemented at Headquarters by using [Electronic reporting](#).

## Setting up Commerce for Norway

This section describes the settings that are specific to and recommended for Norway. For more information, see [Help resources for Dynamics 365 Retail](#).

To use the Norway-specific functionality, you must complete these tasks:

- Set the **Country/region** field to **NOR** (Norway) in the primary address of the legal entity.
- Set the **ISO code** field to **NO** (Norway) in the POS functionality profile of every store that is located in Norway.

You must also specify the following settings for Norway.

### Set up the legal entity

Make sure that the name of the legal entity is specified. This name will be printed on X and Z reports.

Additionally, on the **Bank account information** FastTab, in the **Routing number** field, specify the organization number.

### Set up value-added tax (VAT) per Norwegian requirements

You must create sales tax codes, sales tax groups, and item sales tax groups. You must also set up sales tax information for products and services. For more information about how to set up and use sales tax, see [Sales tax overview](#).

You must also specify sales tax groups and enable the **Prices include sales tax** option for stores that are located in Norway.

### Set up functionality profiles

You must enable auditing and set up receipt numbering.

### Update POS permissions groups and individual permission settings for store workers

Set the **Allow printing receipt copy** permission to an appropriate value:

- **Allow always** – The operator can print a copy of a receipt multiple times.
- **Allow only once** – The operator can print a copy of a receipt only one time.
- **Allow only once, and only if HQ DB is available** – The operator can print a copy of a receipt only one time, and only if the HQ database is available through Commerce Data Exchange: Real-time Service, so that the system can verify that no copies of the receipt have previously been printed in any store.
- **Never** – The operator can't print a copy of a receipt.

### Configure custom fields so that they can be used in receipt formats for sales receipts

On the **Language text** page, add the following records for the labels of the custom fields for receipt layouts. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the table are just examples. You can change them to meet to your requirements.



LANGUAGE ID	TEXT	TEXT ID
en-US	Receipt title	900011
en-US	Is gift card	900012
en-US	Total (sales)	900013
en-US	Tax total (sales)	900014
en-US	Total with tax (sales)	900015
en-US	Tax amount (sales)	900016
en-US	Cash transaction ID	900017

On the **Custom fields** page, add the following records for the custom fields for receipt layouts. Note that **Caption text ID** values must correspond to the **Text ID** values that you specified on the **Language text** page.

NAME	TYPE	CAPTION TEXT ID
ReceiptTitle	Receipt	900011
IsGiftCard	Receipt	900012
SalesTotalExt	Receipt	900013
TaxTotalExt	Receipt	900014
TotalWithTaxExt	Receipt	900015
AmountPerTaxExt	Receipt	900016
CashTransactionSequentialNumber	Receipt	900017

### Configure receipt formats

For all required receipt formats, change the value of the **Print behavior** field to **Always print** for the receipt format.

In the Receipt format designer, add the following custom fields to the appropriate receipt sections. Note that field names correspond to the language texts that you defined in the previous section.

#### 1. Header:

- **Receipt title** – This field identifies the type of receipt.
- **Cash transaction ID** – This field prints the sequential number of the signed cash transaction.

#### 2. Lines:

- **Is gift card** – This field marks the receipt line as related to the Issue gift card or Add to gift card operation.

#### 3. Footer:

- **Total (sales)** – This field prints the receipt's total cash sale amount. The amount excludes tax. Prepayments and gift card operations are excluded.

- **Tax total (sales)** – This field prints the receipt's total tax amount for cash sales. Prepayments and gift card operations are excluded.
- **Total with tax (sales)** – This field prints the receipt's total cash sale amount. The amount includes tax. Prepayments and gift card operations are excluded.
- **Tax amount (sales)** – This field prints the receipt's tax amount for cash sales per tax code. Prepayments and gift card operations are excluded.

For more information about how to work with receipt formats, see [Set up and design receipt formats](#).

### **Configure the SAF-T Cash Register export format**

The SAF-T Cash Register configuration is available for download from Microsoft Dynamics Lifecycle Services (LCS). For more information, see [Import electronic reporting configurations](#). You must download the following configurations:

- **Retail channel data.version.1** – The data model configuration.
- **DMM Retail channel data.version.1.12** – The data model mapping configuration.
- **NO SAF T Cash Register.version.1.15** – The format configuration.

After you import the configurations, on the **Commerce parameters** page, on the **Electronic documents** tab, in the **SAF-T Cash register export format** field, select the name of the format configuration that was imported.

You must also map required master data to predefined SAF-T standard codes. For more information, see the SAF-T Cash register documentation that is provided by the Norwegian Tax Administration. To create the mapping, you must set the new **SAF-T Cash register code** field on the following pages:

- Item groups
- Payment methods
- Sales tax codes

### **Configure channel components**

To enable Norway-specific functionality, you must configure extensions for channel components. For more information, see the [deployment guidelines](#).

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Deployment guidelines for cash registers for Norway

2/18/2021 • 32 minutes to read • [Edit Online](#)

This topic is a deployment guide that shows how to enable the Dynamics 365 Commerce localization for Norway. The localization consists of several extensions of Commerce components. For example, the extensions let you print custom fields on receipts, register additional audit events, sales transactions, and payment transactions in Point of Sale (POS), digitally sign sales transactions, and print X and Z reports in local formats. For more information about the localization for Norway, see [Cash register functionality for Norway](#).

This sample is part of the Retail software development kit (SDK). For information about the SDK, see the [Retail software development kit \(SDK\) architecture](#).

This sample consists of extensions for the Commerce runtime (CRT), Retail Server, and POS. To run this sample, you must modify and build the CRT, Retail Server, and POS projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Microsoft Visual Studio Online (VSO), where no files have been changed yet.

## NOTE

In Commerce 10.0.8 and above, Retail Server is known as Commerce Scale Unit. Because this topic applies to multiple previous versions of the app, *Retail Server* is used throughout the topic.

## NOTE

Some steps in the procedures in this topic differ, depending on the version of Commerce that you're using. For more information, see [What's new or changed in Dynamics 365 Retail](#).

## Using certificate profiles in Commerce channels

In Commerce versions 10.0.15 and later, you can use the [User-defined certificate profiles for retail stores](#) feature that supports failover to offline when Key Vault or Commerce headquarters aren't available. The feature extends the [Manage secrets for retail channels](#) feature.

To apply this functionality in the CRT extension, follow these steps.

1. Create a new CRT extension project (C# class library project type). Use the sample templates from the Retail software development kit (SDK) (RetailSDK\SampleExtensions\CommerceRuntime).
2. Add custom handler for CertificateSignatureServiceRequest in the SequentialSignatureRegister project.
3. To read a secret call, GetUserDefinedSecretCertificateServiceRequest using a constructor with profileId parameter. That will start the functionality working with settings from Certificate profiles. Based on the settings, the certificate will be retrieved either from Azure Key Vault or local machine storage.

```
GetUserDefinedSecretCertificateServiceRequest getUserDefinedSecretCertificateServiceRequest = new
GetUserDefinedSecretCertificateServiceRequest(profileId: "ProfileId", secretName: null, thumbprint: null,
expirationInterval: null); GetUserDefinedSecretCertificateServiceResponse
getUserDefinedSecretCertificateServiceResponse =
request.RequestContext.Execute(getUserDefinedSecretCertificateServiceRequest);
```

```
X509Certificate2 Certificate = getUserDefinedSecretCertificateServiceResponse.Certificate;
```

4. When the certificate is retrieved, proceed with data signing.
5. Build the CRT extension project.
6. Copy the output class library and paste it into ...RetailServer\webroot\bin\Ext for manual testing.
7. In the CommerceRuntime.Ext.config file, update the extension composition section with the custom library information.

## Development environment

Complete these procedures to set up a development environment so that you can test and extend the sample.

### The CRT extension components

The CRT extension components are included in the CRT samples. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`, under `RetailSdk\SampleExtensions\CommerceRuntime`.

#### ReceiptsNorway component

1. Find the `Runtime.Extensions.ReceiptsNorway` project, and build it.
2. In the `Extensions.ReceiptsNorway\bin\Debug` folder, find the `Contoso.Commerce.Runtime.ReceiptsNorway.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Retail Server:** Copy the assembly to the `\bin\ext` folder under the Microsoft Internet Information Services (IIS) Retail Server site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.ReceiptsNorway" />
```

#### WARNING

Do **not** edit the `commerceruntime.config` and `CommerceRuntime.MPOSOOffline.config` files. These files aren't intended for any customizations.

#### SalesPaymentTransExt component

1. Find the `Runtime.Extensions.SalesPaymentTransExt` project, and build it.
2. In the `Extensions.SalesPaymentTransExt\bin\Debug` folder, find the `Contoso.Commerce.Runtime.SalesPaymentTransExt.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker

location.

4. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SalesPaymentTransExt" />
```

#### WARNING

Do not edit the `commerceruntime.config` and `CommerceRuntime.MPOSOOffline.config` files. These files aren't intended for any customizations.

#### XZReportsNorway component

1. Find the `Runtime.Extensions.XZReportsNorway` project, and build it.

2. In the `Extensions.XZReportsNorway\bin\Debug` folder, find the `Contoso.Commerce.Runtime.XZReportsNorway.dll` assembly file.

3. Copy the assembly file to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.

4. Find the extension configuration file for CRT:

- **Retail Server:** The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
- **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.XZReportsNorway" />
```

#### WARNING

Do not edit the `commerceruntime.config` and `CommerceRuntime.MPOSOOffline.config` files. These files aren't intended for any customizations.

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

#### RegisterAuditEvent sample component

1. Find the `Runtime.Extensions.RegisterAuditEventSample` project, and build it.

2. In the `Extensions.RegisterAuditEventSample\bin\Debug` folder, find the `Contoso.Commerce.Runtime.RegisterAuditEventSample.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Retail Server**: Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS**: Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Retail Server**: The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS**: The file is named `CommerceRuntime.MPOSOffline.Ext.config`, and it's under the local CRT client broker location.
5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.RegisterAuditEventSample" />
```

#### WARNING

Do not edit the `commerceruntime.config` and `CommerceRuntime.MPOSOffline.config` files. These files aren't intended for any customizations.

#### SalesTransactionSignature sample component

1. Find the `Runtime.Extensions.SalesTransactionSignatureSample` project.
2. Modify the `App.config` file by specifying the thumbprint, store location, and store name for the certificate that should be used to sign sales transactions.
3. Build the project.
4. In the `Extensions.SalesTransactionSignatureSample\bin\Debug` folder, find the following files:
  - The `Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll` assembly file
  - The `Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll.config` configuration file
5. Copy the files to the CRT extensions folder:
  - **Retail Server**: Copy the assembly to the `\bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS**: Copy the assembly to the `\ext` folder under the local CRT client broker location.
6. Find the extension configuration file for CRT:
  - **Retail Server**: The file is named `commerceruntime.ext.config`, and it's in the `bin\ext` folder under the IIS Retail Server site location.
  - **Local CRT on Modern POS**: The file is named `CommerceRuntime.MPOSOffline.Ext.config`, and it's under the local CRT client broker location.
7. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SalesTransactionSignatureSample" />
```

## WARNING

Do not edit the `commerceruntime.config` and `CommerceRuntime.MPOSOffline.config` files. These files aren't intended for any customizations.

### The Retail Server extension components

#### SalesTransactionSignature Retail Server sample component

1. In the `RetailSDK\SampleExtensions\RetailServer\RetailServer.Extensions.SalesTransactionSignatureSample` folder, find the `RetailServer.Extensions.SalesTransactionSignatureSample` project, and build it.
2. In the `RetailServer\Extensions.SalesTransactionSignatureSample\bin\Debug` folder, find the `Contoso.RetailServer.SalesTransactionSignatureSample.dll` assembly file.
3. Copy the assembly file to the Retail Server extensions folder.
  - [Application update 4](#)
  - [Application update 5 and later](#)
  - [Retail 7.3.1](#)
  - [Retail 7.3.2 and later](#)
  - [Retail 7.3.5 and later](#)
  - [Retail 8.1.1 and later](#)

The folder is the `\bin` folder under the IIS Retail Server site location.

4. Find the configuration file for Retail Server. The file is named `web.config`, and it's in the root folder under the IIS Retail Server site location.
5. Register the Retail Server extensions in the `extensionComposition` section of the configuration file.

```
<add source="assembly" value="Contoso.RetailServer.SalesTransactionSignatureSample" />
```

6. Register the dependencies of the Retail Server extensions.
  - [Application update 4](#)
  - [Application update 5 and later](#)
  - [Retail 7.3.1](#)
  - [Retail 7.3.2 and later](#)
  - [Retail 7.3.5 and later](#)
  - [Retail 8.1.1 and later](#)
1. In the `CommerceRuntime\Extensions.SalesTransactionSignatureSample\bin\Debug` folder, find the following files:
  - The `Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll` assembly file
  - The `Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll.config` configuration file
2. Copy the files to the `\bin` folder under the IIS Retail Server site location.
3. Register the CRT change in the extension configuration file for CRT. This file is named `commerceruntime.ext.config`, and it's in the `bin` folder under the IIS Retail Server site location.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SalesTransactionSignatureSample" />
```

## The Modern POS extension components

### Implement the proxy code for offline mode

This part is equivalent to the Retail Server controller, but it extends the local CRT that is used when the client isn't connected.

1. In the **customization.settings** file, change the **@(RetailServerLibraryPathForProxyGeneration)** section so that it uses the new Retail Server assembly for proxy generation.
2. Implement the following interface methods in the **StoreOperationsManager** class. For the first iteration, add the following code:

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```
public Task<bool> SalesTransactionSignatureServiceIsReady()
{
    throw new NotImplementedException();
}
public Task<FiscalTransaction> GetLastFiscalTransaction(string storeNumber, string terminalId)
{
    throw new NotImplementedException();
}
```

3. To regenerate the proxy code, build the **Proxies** folder from the command line by using the **msbuild /t:Rebuild** command.
4. Resolve the **Proxies.RetailProxy** project dependencies:
  - [Application update 4](#)
  - [Application update 5 and later](#)
  - [Retail 7.3.1](#)
  - [Retail 7.3.2 and later](#)
  - [Retail 7.3.5 and later](#)
  - [Retail 8.1.1 and later](#)

Open **RetailSDK\Proxies\RetailProxy\Proxies.RetailProxy.csproj**, add the **RetailSDK\SampleExtensions\CommerceRuntime\Extensions.SalesTransactionSignatureSample\CommerceRuntime.Extensions.SalesTransactionSignatureSample** project to the solution, and add a project reference to the **RetailProxy** project to reference **SalesTransactionSignatureSample**.

5. Adjust the interface methods in the **StoreOperationsManager** class:
  - [Application update 4](#)
  - [Application update 5 and later](#)
  - [Retail 7.3.1](#)
  - [Retail 7.3.2 and later](#)
  - [Retail 7.3.5 and later](#)
  - [Retail 8.1.1 and later](#)



```

public Task<bool> SalesTransactionSignatureServiceIsReady()
{
    return Task.Run(() =>
CommerceRuntimeManager.Runtime.Execute<SalesTransactionSignatureServiceIsReadyResponse>(new
SalesTransactionSignatureServiceIsReadyRequest(), null).IsReady);
}
public Task<FiscalTransaction> GetLastFiscalTransaction(string storeNumber, string terminalId)
{
    return Task.Run(() => CommerceRuntimeManager.Runtime.Execute<GetLastFiscalTransactionResponse>
(new GetLastFiscalTransactionRequest(), null).FiscalTransaction);
}

```

6. Update the `dllhost.exe.config` file so that the client broker loads the new RetailProxy assembly.

```

<add key="RetailProxyAssemblyName" value="Contoso.Commerce.RetailProxy" />
<add key="AdaptorCallerFullTypeName" value="Contoso.Commerce.RetailProxy.Adapters.AdaptorCaller" />

```

#### Retail proxy extension component (Retail 7.3.1 and later)

Complete the following procedure only if you're using Retail 7.3.1 and later.

1. In the `RetailSDK\SampleExtensions\RetailProxy\RetailProxy.Extensions.SalesTransactionSignatureSample` folder, find the `RetailServer.Extensions.SalesTransactionSignatureSample` project, and build it.
2. In the `RetailProxy\RetailProxy.Extensions.SalesTransactionSignatureSample\bin\Debug` folder, find the `Contoso.Commerce.RetailProxy.SalesTransactionSignatureSample` assembly file.
3. Copy the assembly files to the `\ext` folder under the local CRT client broker location.
4. Register the Retail proxy change in the extension configuration file. The file is named `RetailProxy.MPOSOffline.ext.config`, and it's under the local CRT client broker location.

```

<add source="assembly" value="Contoso.Commerce.RetailProxy.SalesTransactionSignatureSample" />

```

#### Modern POS extension components

1. Open the solution at `RetailSdk\POS\ModernPOS.sln`, and make sure that it can be compiled without errors. Additionally, make sure that you can run Modern POS from Microsoft Visual Studio by using the **Run** command.

#### NOTE

Modern POS must not be customized. You must enable User Account Control (UAC), and you must uninstall previously installed instances of Modern POS as required.

2. Include the following existing source code folders in the `Pos.Extensions` project.

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)
- `AuditEventExtensionSample`

- [SalesTransactionSignatureSample](#)
3. Enable the extensions to be compiled in `tsconfig.json` by removing the following folders from the exclude list.
    - [Application update 4](#)
    - [Application update 5 and later](#)
    - [Retail 7.3.1](#)
    - [Retail 7.3.2 and later](#)
    - [Retail 7.3.5 and later](#)
    - [Retail 8.1.1 and later](#)
  4. Enable the extensions to be loaded in `extensions.json` by adding the following lines in the appropriate place.
    - [Application update 4](#)
    - [Application update 5 and later](#)
    - [Retail 7.3.1](#)
    - [Retail 7.3.2 and later](#)
    - [Retail 7.3.5 and later](#)
    - [Retail 8.1.1 and later](#)

```
{
  "baseUrl": "AuditEventExtensionSample"
},
{
  "baseUrl": "SalesTransactionSignatureSample"
}
```

#### **NOTE**

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the `readme.md` file in the `Pos.Extensions` project.

5. Rebuild the solution.
6. Run Modern POS in the debugger, and test the functionality.

### **Cloud POS extension components**

1. Open the solution at `RetailSdk\POS\CloudPOS.sln`, and make sure that it can be compiled without errors.
2. Include following existing source code folders in the `Pos.Extensions` project.
  - [Application update 4](#)
  - [Application update 5 and later](#)
  - [Retail 7.3.1](#)
  - [Retail 7.3.2 and later](#)
  - [Retail 7.3.5 and later](#)
  - [Retail 8.1.1 and later](#)

- AuditEventExtensionSample
- SalesTransactionSignatureSample

3. Enable the extensions to be compiled in **tsconfig.json** by removing following folders from the exclude list.

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

- AuditEventExtensionSample
- SalesTransactionSignatureSample

4. Enable the extensions to be loaded in **extensions.json** by adding the following lines in the appropriate place.

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```
{
  "baseUrl": "AuditEventExtensionSample"
},
{
  "baseUrl": "SalesTransactionSignatureSample"
}
```

#### **NOTE**

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the readme.md file in the **Pos.Extensions** project.

5. Rebuild the solution.

6. Run the solution by using the **Run** command and following the steps in the Retail SDK handbook.

7. Test the functionality.

#### **Set up required parameters in Headquarters**

For more information, see [Cash register functionality for Norway](#).

## Production environment

Follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Complete the steps in the [Cloud POS extension components](#) or [Modern POS extension components](#) section earlier in this topic.

2. Make the following changes in the package configuration files under the **RetailSdk\Assets** folder:

a. In the **commerceruntime.ext.config** and **CommerceRuntime.MPOSOOffline.Ext.config** configuration files, add the following lines to the **composition** section:

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```
<add source="assembly" value="Contoso.Commerce.Runtime.ReceiptsNorway" />
<add source="assembly" value="Contoso.Commerce.Runtime.RegisterAuditEventSample" />
<add source="assembly" value="Contoso.Commerce.Runtime.SalesPaymentTransExt" />
<add source="assembly" value="Contoso.Commerce.Runtime.SalesTransactionSignatureSample" />
<add source="assembly" value="Contoso.Commerce.Runtime.XZReportsNorway" />
```

b. Enable Commerce Proxy customization:

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

In the **dllhost.exe.config** configuration file, add the following lines to the **appSettings** subsection of the **configuration** section.

```
<add key="RetailProxyAssemblyName" value="Contoso.Commerce.RetailProxy"/>
<add key="AdaptorCallerFullTypeName" value
="Contoso.Commerce.RetailProxy.Adapters.AdaptorCaller"/>
```

3. Make the following changes in the **Customization.settings** package customization configuration file:

a. Enable Retail Proxy customization.

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

Add the following lines to the **<ItemGroup**

**Condition="'@(RetailServerLibraryPathForProxyGeneration)' == ''>** section.

```
<RetailServerLibraryPathForProxyGeneration
Include="$(SdkReferencesPath)\Contoso.RetailServer.SalesTransactionSignatureSample.dll"/>
```

b. Add the following lines to the **ItemGroup** section to include the CRT extensions in the deployable packages:

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.ReceiptsNorway.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.RegisterAuditEventSample.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SalesPaymentTransExt.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll.con
fig" />
<ISV_CommerceRuntime_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.XZReportsNorway.dll" />
```

c. Add following lines to the **ItemGroup** section to include the Retail Server extension in the deployable packages:

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

```
<ISV_RetailServer_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.RetailServer.SalesTransactionSignatureSample.dll" />
<ISV_RetailServer_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll" />
<ISV_RetailServer_CustomizableFile
Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll.con
fig" />
```

4. Modify the following files to include the resource files for Norway in deployable packages:

- Packages\_SharedPackagingProjectComponents\Sdk.ModernPos.Shared.csproj
- Packages\RetailServer\Sdk.RetailServerSetup.proj
- For the **Sdk.ModernPos.Shared.csproj** file
  - Add line to the **ItemGroup** section

```
<<File_name> Include="$(SdkReferencesPath)\nb-NO\*" />
```

#### NOTE

Instead of the <File\_name> specify a name of the resource file. The same is relevant for the other examples given below.

- Add line to the **Target Name="CopyPackageFiles"** section

```
<Copy SourceFiles="@(<File_name>)"
DestinationFolder="$(OutputPath)content.folder\CustomizedFiles\ClientBroker\ext\nb-NO"
SkipUnchangedFiles="true" />
```

- For the **Sdk.RetailServerSetup.proj** file

- Add line to the **ItemGroup** section

```
<<File_name> Include="$(SdkReferencesPath)\nb-NO\*" />
```

- Add line to the **Target Name="CopyPackageFiles"** section

```
<Copy SourceFiles="@(<File_name>)"
DestinationFolder="$(OutputPath)content.folder\RetailServer\Code\bin\ext\nb-NO"
SkipUnchangedFiles="true" />
```

5. Modify the certificate's configuration file by specifying the thumbprint, store location, and store name for the certificate that should be used to sign sales transactions. Then copy the configuration file to the **References** folder.

- [Application update 4](#)
- [Application update 5 and later](#)
- [Retail 7.3.1](#)
- [Retail 7.3.2 and later](#)
- [Retail 7.3.5 and later](#)
- [Retail 8.1.1 and later](#)

The file is named **Contoso.Commerce.Runtime.SalesTransactionSignatureSample.dll.config**, and it's under **CommerceRuntime\Extensions.SalesTransactionSignatureSample\bin\Debug**.

6. Update Retail Server configuration file. In the **RetailSDK\Packages\RetailServer\Code\web.config** file, add the following lines to the **extensionComposition** section.

```
<add source="assembly" value="Contoso.RetailServer.SalesTransactionSignatureSample" />
```

7. Run **msbuild** for the whole Retail SDK to create deployable packages.
8. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).

### **Enable the digital signature in offline mode for Modern POS**

To enable the digital signature in offline mode for Modern POS, you must follow these steps after you activate Modern POS on a new device.

1. Sign in to POS.
2. On the **Database connection status** page, make sure that the offline database is fully synchronized. When the value of the **Pending downloads** field is **0** (zero), the database is fully synchronized.
3. Sign out of POS.
4. Wait a while for the offline database to be fully synchronized.
5. Sign in to POS.
6. On the **Database connection status** page, make sure that the offline database is fully synchronized. When

the value of the **Pending transactions in offline database** field is **0** (zero), the database is fully synchronized.

7. Restart Modern POS.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cash register functionality for Sweden

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic provides an overview of the cash register functionality that is available for Sweden in Dynamics 365 Commerce. It also provides guidelines for setting up the functionality. The functionality consists of the following parts:

- Common point-of sale (POS) features that are made available to customers in all countries or regions, such as an option to prevent sales and returns from being combined on one receipt
- Sweden-specific features, such as additional counters in daily POS reports
- A sample for integration of POS with Sweden-specific fiscal devices that are known as control units.

## Overview of cash register functionality for Sweden

### Common POS features

To learn about common POS features that are available to customers in all countries or regions, see [Commerce home page](#).

Additionally, the following POS features that were implemented for Sweden have been made available to customers in all countries or regions:

- **Prohibit sales and returns from being combined on one receipt.** When you set the **Prohibit mixing sales and returns in one receipt** parameter in the POS functionality profile to **Yes**, Cloud POS and Modern POS won't let users create a transaction that contains both positive and negative lines.
- **Print text fields on the receipt in a large font size.** You can use the **Font size** parameter in the Receipt format designer to specify that the large font size should be used for a field in a receipt format. (The large font size is approximately double the usual font size.) For example, you can use this parameter to print the "Copy" indicator on a receipt copy in large characters.
- **Register the printing of receipt copies in the POS audit event log.** You can use the **Audit** parameter in the POS functionality profile to enable the printing of receipt copies and other POS audit events to be registered. The audit events are registered in the channel database and in Headquarters. You can view the audit events on the **Audit events** page.
- **Prevent a copy of a receipt from being printed more than one time.** When the parameter **Audit** in the POS functionality profile is enabled, the **Allow printing receipt copies** POS permission controls whether receipt copies can be printed. There is also an option to prevent a copy of a receipt from being printed more than one time.

### Sweden-specific POS features

The following Sweden-specific POS features are enabled when the **ISO code** parameter in the POS functionality profile is set to **SE**:

- Additional counters on daily POS reports, such as X-reports and Z-reports:
  - Receipt copy count and total amount
  - Value added tax amounts per tax rate
  - Total quantities of items and services sold
  - Total sales excluding and including VAT
  - Grand total sales, returns, and net
- An **Electronic journal (Sweden)** channel report that lists continuous use events in the POS, such as sales, returns, receipt copies, drawer openings, and price overrides.



#### NOTE

Currently, the **Electronic journal (Sweden)** report can't be exported or printed. However, functionality for exporting and printing the report will be added later.

### Integration of Retail POS with control units

- [Retail 10.0.6 and earlier](#)
- [Retail 10.0.7 and later](#)

For more information about the integration with control units that is available in Retail versions up to and including Retail 10.0.6, see [Sample for POS integration with control units for Sweden \(legacy\)](#).

## Setting up Commerce for Sweden

This section describes the settings that are specific to and recommended for Sweden. For more information, see [Commerce home page](#).

To use the Sweden-specific functionality, you must complete these tasks:

- Set the **Country/region** field to **SWE** (Sweden) in the primary address of the legal entity.
- Set the **ISO code** field to **SE** (Sweden) in the POS functionality profile of every store that is located in Sweden.

Sweden-specific settings can be divided into two groups:

- General settings
- Control unit-specific settings

### General settings

You must specify the following general settings for Sweden.

1. Set up the following parameters for value-added tax (VAT) per Swedish requirements:

- Sales tax codes
- Sales tax groups
- Item sales tax groups
- Sales tax settings in items (item sales tax groups for sales)

For more information about how to set up and use sales tax, see [Sales tax overview](#).

2. On the **All stores** page, update store details. Specifically, set the following parameters:

- In the **Sales tax group** field, set the sales tax group that should be used for sales to the default customer.
- Select the **Prices include sales tax** check box.
- Set the **Store name** field so that it includes the company name. This change helps guarantee that the company name appears on a sales receipt. Alternatively, you can add the company name to the sales receipt layout as free-format text.
- Set the **Tax identification number (TIN)** field so that it includes the company identification number. This change helps guarantee that the company identification number appears on a sales receipt. Alternatively, you can add the company identification number to the sales receipt layout as free-format text.

3. Set up POS functionality profiles:

- On the **Functions** FastTab, select the **Audit** and **Prohibit mixing sales and returns in one**

receipt check boxes.

- On the **Receipt numbering** FastTab, set up receipt numbering. Create or update records for the **Sale** and **Return** receipt transaction types. Set the formats so that they include only numeric characters. Clear the **Independent sequence** check box in both records.
4. Update POS permissions groups and individual permission settings for store workers. Set the **Allow printing receipt copy** permission to an appropriate value:
    - **Allow always** – The operator can print a copy of a receipt multiple times.
    - **Allow only once** – The operator can print a copy of a receipt only one time.
    - **Allow only once, and only if HQ DB is available** – The operator can print a copy of a receipt only one time, and only if the headquarters database is available through Real-Time service, so that the system can verify that no copies of the receipt have previously been printed in any store.
    - **Never** – The operator can't print a copy of a receipt.
  5. Make the required changes to receipt formats for sales receipts:
    - Change the value of the **Print behavior** field to **Always print** for the receipt format.
    - In the Receipt format designer, make these changes:
      - Add at least the following fields to the **Header** section of the receipt layout:
        - **Store name** and **Tax Identification Number** fields, so that the company name and identity number are printed receipts. Alternatively, you can add the company name and identity number to the layout as free-format text.
        - **Store address**, **Date**, **Time 24H**, **Receipt Number**, and **Register number** fields.
        - **Reprint Message** field, so that the "Copy" indicator is printed on receipt copies. Set the **Font size** field for the **Reprint Message** field to **Large**.
      - Add at least the following field to the **Lines** section of the receipt layout: **Item name**, **Qty**, and **Total price with tax**.
      - Add at least the following fields to the **Footer** section of the receipt layout:
        - Tax fields, so that the receipt tax amounts for each tax rate are printed. For example, add the **Tax Id**, **Tax percentage**, and **Tax amounts** fields to one line of the layout.
        - Payment fields, so that the payment amounts for each payment method are printed. For example, add the **Tender name** and **Tender amount** fields to one line of the layout.
  6. On the **Channel reports configuration** page, set up the **Electronic journal (Sweden)** report. In the **Permission groups** field, select the POS permission groups that should be allowed to run the report.

### Control unit-specific settings

- [Retail 10.0.6 and earlier](#)
- [Retail 10.0.7 and later](#)

For more information about setting up and configuring the integration with control units that is available in Retail versions up to and including Retail 10.0.6, see [Sample for POS integration with control units for Sweden \(legacy\)](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Control unit integration sample for Sweden

2/18/2021 • 20 minutes to read • [Edit Online](#)

## NOTE

This sample fiscal integration functionality replaces the earlier [Sample for POS integration with control units for Sweden](#). The earlier sample doesn't take advantage of the [fiscal integration framework](#) and will become obsolete in later updates. For information about how to migrate from the earlier sample to the current sample, see the [Migrating from the earlier integration sample](#) section.

## Introduction

The Commerce functionality for Sweden includes a sample integration of the point of sale (POS) with Sweden-specific fiscal devices that are known as *control units*. This sample extends the [fiscal integration functionality](#). It's assumed that a control unit is physically connected to a Hardware station that the POS is paired with. As an example, this sample uses the application programming interface (API) of the [CleanCash Type A](#) control unit by Retail Innovation HTT AB. Version 1.1.4 of the CleanCash API is used.

The sample is provided in the form of source code and is part of the Retail software development kit (SDK).

Microsoft doesn't release any hardware, software, or documentation from Retail Innovation HTT AB. For information about how to get the control unit and operate it, contact [Retail Innovation HTT AB](#).

## Scenarios

The control unit integration sample for Sweden includes the following capabilities:

- Sales, returns, and receipt copies are automatically registered in a control unit that is connected to the Hardware station that is paired with the POS.
- The control code and the manufacturing number of the control unit for a registered transaction are captured from the control unit and saved in the transaction. This data is also referred to as a *fiscal response*. The fiscal response can be viewed on the [Store transactions](#) page.
- Custom fields for the control code and the manufacturing number of the control unit can be added to a receipt layout. In that way, you can print the fiscal response for a transaction on a receipt.
- The fiscal response for a transaction is shown on the [Electronic journal \(Sweden\)](#) channel report.
- Several error handling options are available. Here are some examples:
  - Retry fiscal registration, if a retry is possible. You can retry fiscal registration if, for example, the control unit isn't connected, isn't ready, or isn't responding.
  - Postpone fiscal registration.
  - Skip fiscal registration, or mark the transaction as registered, and include info codes to capture the reason for the failure and additional information.
  - Verify the availability of the control unit before a new sales transaction is opened or a sales transaction is finalized.

### Default data mapping

The following default data mapping is included in the fiscal document provider configuration that is distributed as a part of the fiscal integration sample.

**Value-added tax (VAT) code mapping** sets device-specific value-added tax (VAT) codes to corresponding sales tax codes. VAT code mapping should have the following format:

```
1 : code1 ; 2 : code2
```

Here is an explanation of this format:

- *1* and *2* are device-specific VAT codes.
- A semicolon (;) is used as a separator.
- *code1* and *code2* are sales tax codes that are configured in Headquarters.

Control units support up to four different VAT codes. Therefore, the VAT code mapping might be set up as shown here:

```
1 : code1 ; 2 : code2 ; 3 : code3 ; 4 : code4
```

## NOTE

Multiple sales tax codes can be mapped to the same device-specific VAT code.

### Limitations of the sample

The control unit integration sample for Sweden doesn't currently support customer order scenarios.

## Setting up the integration with control units

For more information about the setup that is required for Sweden, see [Setting up Commerce for Sweden](#).

## Configuring Sweden-specific receipts

### Configure custom fields so that they can be used in receipt formats for sales receipts

You can configure the language text and custom fields that are used in POS receipt formats. The default company of the user who creates the receipt setup should be the same legal entity where the language text setup is created. Alternatively, the same language texts should be created in both the user's default company and the legal entity of the store that the setup is created for.

On the **Language text** page, add the following records for the labels of the custom fields for receipt layouts. Note that the **Language ID**, **Text ID**, and **Text** values that are shown in the table are just examples. You can change them to meet your requirements. However, the **Text ID** values that you use must be unique, and they must be equal to or greater than 900001.

Add the following POS labels in the **POS** section of the **Language text** page.

LANGUAGE ID	TEXT ID	TEXT
en-US	900001	Register control code
en-US	900002	Register device

On the **Custom fields** page, add the following records for the custom fields for receipt layouts. Note that **Caption text ID** values must correspond to the **Text ID** values that you specified on the **Language text** page.

NAME	TYPE	CAPTION TEXT ID
SE_FISCALREGISTERCONTROLCODE	Receipt	900001
SE_FISCALREGISTERID	Receipt	900002

### Configure receipt formats

For every receipt format that is required, change the value of the **Print behavior** field to **Always print**.

In the Receipt format designer, add the following custom fields to the **Footer** section. Note that field names correspond to the language texts that you defined in the previous section of this topic.

- **Register control code** – This field prints the control code.
- **Register device** – This field prints the manufacturing number of the control unit.

For more information about how to work with receipt formats, see [Receipt templates and printing](#).

### Configure fiscal integration

Complete the fiscal integration setup steps that are described in [Set up the fiscal integration for Commerce channels](#):

- [Set up a fiscal registration process](#). Be sure to note the settings for the fiscal registration process that are [specific to this control unit integration sample](#).
- [Set error handling settings](#).
- [Enable manual execution of postponed fiscal registration](#).

## Deployment guidelines for cash registers for Sweden

This topic serves as a deployment guide that shows how to enable the localization of Microsoft Dynamics 365 Commerce for Sweden. The localization is part of the Retail SDK. For information about how to install and use the Retail SDK, see the [Retail SDK documentation](#).

This sample consists of extensions for the Commerce runtime (CRT), POS, and Hardware station. To run this sample, you must modify and build the CRT and Hardware station projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Microsoft Azure DevOps, where no files have been changed yet.

Follow these steps to configure a development environment so that you can test and extend the sample. You must also complete all the required setup tasks that are described in the [Setting up the integration with control units](#) section.

### Enable CRT extensions

The CRT extension components are included in the CRT samples. To complete the following procedures, open the CRT solution, `CommerceRuntimeSamples.sln`, under `RetailSdk\SampleExtensions\CommerceRuntime`.

#### DocumentProvider.CleanCashSample component

1. Find the `Runtime.Extensions.DocumentProvider.CleanCashSample` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.CleanCashSample\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit:** Copy the assembly to the `\bin\ext` folder under the Microsoft Internet Information Services (IIS) Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `commercerruntime.ext.config`, and it's in the `bin\ext`

folder under the IIS Commerce Scale Unit site location.

- **Local CRT on Modern POS:** The file is named **CommerceRuntime.MPOSOOffline.Ext.config**, and it's under the local CRT client broker location.

5. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample" />
```

#### Update the extension configuration file

1. Find the extension configuration file for CRT:

- **Commerce Scale Unit:** The file is named **commerceruntime.ext.config**, and it's in the **bin\ext** folder under the IIS Commerce Scale Unit site location.
- **Local CRT on Modern POS:** The file is named **CommerceRuntime.MPOSOOffline.Ext.config**, and it's under the local CRT client broker location.

2. Register the CRT change in the extension configuration file.

```
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsSweden" />
```

#### Enable Hardware station extensions

The Hardware station extension components are included in the Hardware station samples. To complete the following procedures, open the solution, **HardwareStationSamples.sln.sln**, under **RetailSdk\SampleExtensions\HardwareStation**.

##### CleanCash component

1. Find the **HardwareStation.Extension.CleanCashSample** project, and build it.

2. In the **Extension.CleanCashSample\bin\Debug** folder, find the **Contoso.Commerce.HardwareStation.CleanCashSample.dll** assembly file.

3. Copy the assembly file to the Hardware station extensions folder:

- **Shared hardware station:** Copy the file to the **bin** folder under the IIS Hardware station site location.
- **Dedicated hardware station on Modern POS:** Copy the file to the Modern POS client broker location.

4. Find the extension configuration file for the Hardware station's extensions. The file is named **HardwareStation.Extension.config**.

- **Shared hardware station:** The file is under the IIS Hardware station site location.
- **Dedicated hardware station on Modern POS:** The file is under the Modern POS client broker location.

5. Add the following line to the **composition** section of the configuration file.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.CleanCashSample.dll" />
```

#### Enable Modern POS extension components

1. Open the solution at **RetailSdk\POS\ModernPOS.sln**, and make sure that it can be compiled without errors. Additionally, make sure that you can run Modern POS from Microsoft Visual Studio by using the **Run** command.

##### NOTE

Modern POS must not be customized. You must enable User Account Control (UAC), and you must uninstall previously installed instances of Modern POS as required.

2. Enable the extensions that must be loaded by adding the following lines in the **extensions.json** file.

```
{
  "extensionPackages": [
    {
      "baseUri": "Microsoft/AuditEvent.SE"
    }
  ]
}
```

##### NOTE

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the **readme.md** file in the **Pos.Extensions** project.

3. Rebuild the solution.

4. Run Modern POS in the debugger, and test the functionality.

#### Enable Cloud POS extension components

1. Open the solution at **RetailSdk\POS\CloudPOS.sln**, and make sure that it can be compiled without errors.

2. Enable the extensions that must be loaded by adding the following lines in the `extensions.json` file.

```
{
  "extensionPackages": [
    {
      "baseUri": "Microsoft/AuditEvent.SE"
    }
  ]
}
```

#### NOTE

For more information, and for samples that show how to include source code folders and enable extensions to be loaded, see the instructions in the `readme.md` file in the `Pos.Extensions` project.

3. Rebuild the solution.
4. Run the solution by using the `Run` command and following the steps in the Retail SDK handbook.

### Set up the registration process

To enable the registration process, follow these steps to set up Headquarters. For more details, see [Set up a fiscal registration process](#).

1. Go to **Retail and Commerce > Headquarters setup > Parameters > Shared parameters**. On the **General** tab, set the **Enable fiscal integration** option to **Yes**.
2. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connectors**, and load the connector configuration. The file location is `RetailSdk\SampleExtensions\HardwareStation\Extension.CleanCashSample\Configuration\ConnectorCleanCashSample.xml`.
3. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal document providers**, and load the document provider configuration. The file location is `RetailSdk\SampleExtensions\CommerceRuntime\Extensions.DocumentProvider.CleanCashSample\Configuration\DocumentProviderF`
4. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector functional profiles**. Create a new connector functional profile, and select the document provider and the connector that you loaded earlier. Update the data mapping settings as required.
5. Go to **Retail and Commerce > Channel setup > Fiscal integration > Connector technical profiles**. Create a new connector technical profile, and select the connector that you loaded earlier. Update the connection settings as required.
6. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal connector groups**. Create a new fiscal connector group for the connector functional profile that you created earlier.
7. Go to **Retail and Commerce > Channel setup > Fiscal integration > Fiscal registration processes**. Create a new fiscal registration process, create a fiscal registration process step, and select the fiscal connector group that you created earlier.
8. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Functionality profiles**. Select a functionality profile that is linked to the store where the registration process should be activated. On the **Fiscal registration process** FastTab, select the fiscal registration process that you created earlier.
9. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**. Select a hardware profile that is linked to the Hardware station that the control unit will be connected to. On the **Fiscal peripherals** FastTab, select the connector technical profile that you created earlier.
10. Open the distribution schedule (**Retail and Commerce > Retail and Commerce IT > Distribution schedule**), and select jobs **1070** and **1090** to transfer data to the channel database.

### Production environment

The previous procedure enables the extensions that are components of the fiscal registration service integration sample. In addition to completing that procedure, you must follow these steps to create deployable packages that contain Commerce components, and to apply those packages in a production environment.

1. Make the following changes in the package configuration files under the `RetailSdk\Assets` folder:

- In the `commerceruntime.ext.config` and `CommerceRuntime.MPOSOOffline.Ext.config` configuration files, add the following lines to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsSweden" />
```

- In the `HardwareStation.Extension.config` configuration file, add the following line to the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.CleanCashSample" />
```

2. Make the following changes in the `Customization.settings` package customization configuration file under the `BuildTools` folder:

- Add the following lines to include the CRT extensions in the deployable packages.

```
<ISV_CommerceRuntime_CustomizableFile
  Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample.dll"
/>
```

- Add the following line to include the Hardware station extension in the deployable packages.

```
<ISV_HardwareStation_CustomizableFile
  Include="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.CleanCashSample.d11" />
```

3. Enable the POS extension by adding the following lines in the `extensions.json` file under the `RetailSDK\POS\Extensions` folder.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/AuditEvent.SE"
    }
  ]
}
```

4. Start the MSBuild Command Prompt for Visual Studio utility, and run `msbuild` under the Retail SDK folder to create deployable packages.
5. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Create deployable packages](#).

## Design of the extensions

### CRT extension design

The purpose of the extension that is a fiscal document provider is to generate service-specific documents and handle responses from the fiscal registration service.

The CRT extension is `Runtime.Extensions.DocumentProvider.CleanCashSample`.

For more details about the design of the fiscal integration solution, see [Fiscal registration process and fiscal integration samples for fiscal devices](#).

#### Request handler

There is a single `DocumentProviderCleanCash` request handler for the document provider. This handler is used to generate fiscal documents for the fiscal registration service.

This handler is inherited from the `INamedRequestHandler` interface. The `HandlerName` method is responsible for returning the name of the handler. The handler name should match the connector document provider name that is specified in Headquarters.

The connector supports the following requests:

- `GetFiscalDocumentDocumentProviderRequest` – This request contains information about what document should be generated. It returns a service-specific document that should be registered in the fiscal registration service.
- `GetSupportedRegistrableEventsDocumentProviderRequest` – This request returns the list of events to subscribe to. Currently, sales events and audit events are supported.

#### Configuration

The `DocumentProviderFiscalCleanCashSample` configuration file is in the `Configuration` folder of the extension project. The purpose of this file is to enable settings for the document provider to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added:

- VAT codes mapping

### Hardware station extension design

The purpose of the extension that is a fiscal connector is to communicate with the fiscal registration service.

The Hardware station extension is `HardwareStation.Extension.CleanCashSample`. It uses the HTTP protocol to submit documents that the CRT extension generates to the fiscal registration service. It also handles the responses that are received from the fiscal registration service.

#### Request handler

The `CleanCashHandler` request handler is the entry point for handling requests to the fiscal registration service.

The handler is inherited from the `INamedRequestHandler` interface. The `HandlerName` method is responsible for returning the name of the handler. The handler name should match the fiscal connector name that is specified in Headquarters.

The connector supports the following requests:

- `SubmitDocumentFiscalDeviceRequest` – This request sends documents to the fiscal registration service and returns a response from it.
- `IsReadyFiscalDeviceRequest` – This request is used for a health check of the fiscal registration service.
- `InitializeFiscalDeviceRequest` – This request is used to initialize the fiscal registration service.

#### Configuration

The configuration file is in the `Configuration` folder of the extension project. The purpose of the file is to enable settings for the fiscal connector to be configured from Headquarters. The file format is aligned with the requirements for fiscal integration configuration. The following settings are added:

- `Connections string` – The control unit connection settings.

- **Timeout** – The amount of time, in milliseconds, that the driver will wait for a response from the fiscal registration service.

## Migrating from the earlier integration sample

If you're using the earlier [Sample for POS integration with control units for Sweden](#), you might have to migrate from it to the current integration sample. To uptake the change and receive timely updates for the features for Sweden in the future, you might have to upgrade, make minor code and configuration adjustments in the extensions that you built, and rebuild your solutions. No major changes are required in the extension logic that you created. The earlier integration sample and your customizations will continue to work if no changes are made from your side. Therefore, you can plan, prepare for, and do the uptake for your environment.

### Migration process

The migration from the earlier integration sample to the current control unit integration sample should be based on the concept of a gradual update. In other words, all Headquarters and Commerce Scale Unit components should already be updated before you start to update the POS and Hardware station components.

To help prevent a situation where an event or transaction is signed twice (that is, it's signed by both the earlier extension and the current extension), or where it can't be signed because of the missing configuration, we recommend that you turn off all POS and Hardware station devices that use the earlier sample, and then update them simultaneously. This simultaneous update can be done, for example, on a store-by-store basis by updating the store's functionality profile and the Hardware station's hardware profile.

The migration process should consist of the following steps.

1. Update the Headquarters components.
2. Update the Commerce Scale Unit components, and enable the extensions of the current sample.
3. Make sure that all offline transactions are synced from offline-enabled MPOS devices.
4. Turn off all devices that use the components of the earlier sample.
5. Complete the setup tasks that are described in the [Setting up the integration with control units](#) section.
6. Update the POS and Hardware station components, disable the extensions that are parts of the earlier sample, and enable the extensions of the current sample.

#### NOTE

Depending on the type of environment, you can find more technical details about the migration process in either the [Migration in a development environment](#) section or the [Migration in a production environment](#) section.

### Migration in a development environment

#### Update CRT

1. Find the `Runtime.Extensions.DocumentProvider.CleanCashSample` project, and build it.
2. In the `Runtime.Extensions.DocumentProvider.CleanCashSample\bin\Debug` folder, find the `Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample.dll` assembly file.
3. Copy the assembly file to the CRT extensions folder:
  - **Commerce Scale Unit:** Copy the assembly to the `\bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** Copy the assembly to the `\ext` folder under the local CRT client broker location.
4. Find the extension configuration file for CRT:
  - **Commerce Scale Unit:** The file is named `CommerceRuntime.ext.config`, and it's in the `bin\ext` folder under the IIS Commerce Scale Unit site location.
  - **Local CRT on Modern POS:** The file is named `CommerceRuntime.MPOSOOffline.Ext.config`, and it's in the `bin\ext` folder under the local CRT client broker location.

#### WARNING

Do not edit the `CommerceRuntime.config` and `CommerceRuntime.MPOSOOffline.config` files. These files aren't intended for any customizations.

5. Find and remove the earlier CRT extension from the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.Runtime.FiscalRegisterReceiptSample" />
```

#### WARNING

Don't complete this step until you update all POS devices that work with this CRT instance.

6. Register the current sample CRT extensions in the extension configuration file by adding the following lines.



```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsSweden" />
```

#### Update Hardware station

1. Find the `HardwareStation.Extension.CleanCashSample` project, and build it.
2. In the `Extension.CleanCashSample\bin\Debug` folder, find the `Contoso.Commerce.HardwareStation.CleanCashSample.dll` assembly file.
3. Copy the assembly file to the Hardware station extensions folder:
  - **Shared hardware station:** Copy the file to the `bin` folder under the IIS Hardware station site location.
  - **Dedicated hardware station on Modern POS:** Copy the file to the Modern POS client broker location.
4. Find the `HardwareStation.Extension.config` extension configuration file:
  - **Remote Hardware station:** The file is under the IIS Hardware station site location.
  - **Local Hardware station on Modern POS:** The file is under the Modern POS client broker location.

#### WARNING

Do not edit the `CommerceRuntime.config` and `CommerceRuntime.MPOSOOffline.config` files. These files aren't intended for any customizations.

5. Find and remove the earlier Hardware station extension from the extension configuration file.
  - [Retail 7.3 and earlier](#)
  - [Retail 7.3.1 and later](#)
  - [Retail 10.0 and later](#)

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.Extension.FiscalRegisterSample" />
```

6. Add the following line to the **composition** section of the extension configuration file.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.CleanCashSample.dll" />
```

#### Update Modern POS

1. Open the solution at `RetailSdk\POS\CloudPOS.sln`.
2. Disable the earlier POS extension by removing the following lines from the `extensions.json` file.

```
{
  "baseUrl": "FiscalRegisterSample"
}
```

3. Enable the current sample POS extension by adding the following lines in the `extensions.json` file.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/AuditEvent.SE"
    }
  ]
}
```

#### Update Cloud POS

1. Open the solution at `RetailSdk\POS\ModernPOS.sln`.
2. Disable the earlier POS extension by removing the following lines from the `extensions.json` file.

```
{
  "baseUrl": "FiscalRegisterSample"
}
```

3. Enable the current sample POS extension by adding the following lines in the `extensions.json` file.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/AuditEvent.SE"
    }
  ]
}
```

#### Migration in a production environment

##### Update CRT

1. Remove the earlier CRT extension from the `CommerceRuntime.ext.config` and `CommerceRuntime.MPOSOOffline.Ext.config` configuration files under the `RetailSdk\Assets` folder.

```
<add source="assembly" value="Contoso.Commerce.Runtime.FiscalRegisterReceiptSample" />
```

#### WARNING

Don't complete this step until you update all POS devices that work with this CRT instance.

2. Enable the current sample CRT extensions by making the following changes in the **CommerceRuntime.ext.config** and **CommerceRuntime.MPOffline.Ext.config** configuration files under the **RetailSdk\Assets** folder.

```
<add source="assembly" value="Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample" />
<add source="assembly" value="Microsoft.Dynamics.Commerce.Runtime.ReceiptsSweden" />
```

3. In the **Customization.settings** package customization configuration file under the **BuildTools** folder, add the following lines to include the current sample CRT extension in deployable packages.

```
<ISV_CommerceRuntime_CustomizableFile
  Include="$(SdkReferencesPath)\Contoso.Commerce.Runtime.DocumentProvider.CleanCashSample.dll" />
```

#### Update Hardware station

1. Remove the earlier Hardware station extension by modifying the **HardwareStation.Extension.config** configuration file.

- [Retail 7.3 and earlier](#)
- [Retail 7.3.1 and later](#)
- [Retail 10.0 and later](#)

Remove the following section from the **HardwareStation.Shared.config** and **HardwareStation.Dedicated.config** configuration files.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.Extension.FiscalRegisterSample" />
```

2. Enable the current sample Hardware station extension by adding the following line to the **composition** section in the **HardwareStation.Extension.config** configuration file.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.CleanCashSample" />
```

3. Make the following changes in the **Customization.settings** package customization configuration file under the **BuildTools** folder:

- Remove the following lines to exclude the earlier Hardware station extension from deployable packages.

```
<ISV_CommerceRuntime_CustomizableFileInclude="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.Extension.FiscalRegisterSample.dll" />
```

- Add the following line to include the current sample Hardware station extension in deployable packages.

```
<ISV_HardwareStation_CustomizableFile
  Include="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.CleanCashSample.dll" />
```

#### Update Modern POS

1. Open the solution at **RetailSdk\POS\CloudPOS.sln**.

2. Disable the earlier POS extension:

- In the **tsconfig.json** file, add the **FiscalRegisterSample** folder to the exclude list.
- Remove the following lines from the **extensions.json** file under the **RetailSDK\POS\Extensions** folder.

```
{
  "baseUri": "FiscalRegisterSample"
}
```

3. Enable the current sample POS extension by adding the following lines in the **extensions.json** file under the **RetailSDK\POS\Extensions** folder.

```
{
  "extensionPackages": [
    {
      "baseUri": "Microsoft/AuditEvent.SE"
    }
  ]
}
```

#### Update Cloud POS

1. Open the solution at `RetailSdk\POS\ModernPOS.sln`.
2. Disable the earlier POS extension:
  - In the `tsconfig.json` file, add the `FiscalRegisterSample` folder to the exclude list.
  - Remove the following lines from the `extensions.json` file under the `RetailSDK\POS\Extensions` folder.

```
{
  "baseUrl": "FiscalRegisterSample"
}
```

3. Enable the current sample POS extension by adding the following lines in the `extensions.json` file under the `RetailSDK\POS\Extensions` folder.

```
{
  "extensionPackages": [
    {
      "baseUrl": "Microsoft/AuditEvent.SE"
    }
  ]
}
```

#### Create deployable packages

Run `msbuild` for the whole Retail SDK to create deployable packages. Apply the packages via LCS or manually. For more information, see [Retail SDK packaging](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Sample for POS integration with control units for Sweden (legacy)

2/18/2021 • 9 minutes to read • [Edit Online](#)

## NOTE

This sample fiscal integration functionality does not take advantage of the [fiscal integration framework](#) and will be deprecated in later updates. You should use the [Control unit integration sample for Sweden](#) instead.

This sample shows how to create Dynamics 365 Commerce extensions to integrate Retail Modern POS or Cloud POS with a fiscal register. Specifically, this sample includes the code for integrating Retail POS with control units for Sweden. It's assumed that a control unit is physically connected to a Hardware station that POS is paired with. As an example, this sample uses the application programming interface (API) of the CleanCash® Type A control unit by Retail Innovation HTT AB. Version 1.1.4 of the CleanCash® API is used. For the integration package that includes the API and documentation, contact the manufacturer of the device.

This sample is a part of the Retail software development kit (SDK). For information about how to install and use the Retail SDK, see the [Retail software development kit \(SDK\) architecture](#).

This sample consists of extensions for the Hardware station, commerce runtime (CRT), and point of sale (POS). To run this sample, you must modify and build the Hardware station, CRT, and POS projects. We recommend that you use an unmodified Retail SDK to make the changes that are described in this topic. We also recommend that you use a source control system, such as Microsoft Visual Studio Online (VSO), where no files have been changed yet.

## NOTE

Some steps in the procedures in this topic differ, depending on the version of Commerce that you're using. For more information, see [What's new or changed in Dynamics 365 Retail](#).

## NOTE

In Commerce 10.0.8 and above, Retail Server is known as Commerce Scale Unit. Because this topic applies to multiple previous versions of the app, *Retail Server* is used throughout the topic.

## Overview of integration with control units

The sample includes the following capabilities:

- Sales, returns, and receipt copies are automatically registered in a control unit that is connected to the Hardware station that is paired with the POS.
- The control code and the manufacturing number of the control unit for a registered transaction are captured from the control unit and saved in the transaction. (This data is also referred to as *fiscal data*.) The fiscal data can be viewed on the **Store transactions** page.
- Custom fields for the control code and the manufacturing number of the control unit can be added to a receipt format, so that you can print the fiscal data for the transaction on a receipt.
- The fiscal data for a transaction is printed on the **Electronic journal (Sweden)** channel report.

- If a failure occurs during the registration of a transaction in the control unit, the fiscal data for the transaction remains blank. In this case, a new transaction can't be started, and the current shift can't be closed. The operator will be asked to try to register the unregistered transaction again in the control unit. If the second attempt fails, the operator can skip the registration, provided that the operator has a special permission. If the operator skips the registration of a transaction in the control unit, information about this event is saved in the transaction instead of the fiscal data.

#### NOTE

Currently, the control unit integration sample doesn't support customer orders. However, a sample that supports customer orders will be available at a later date.

## Setting up integration with control units

You must specify the following settings, so that Retail POS is integrated with control units for Sweden.

1. Create fiscal register configurations, and assign them to hardware profiles:
  - a. On the **Fiscal register configurations** page, create a new fiscal register configuration record. Set the name and the description of the configuration.
  - b. Fill in the configuration content. For this sample, a configuration is an XML file that establishes the mapping between sales tax codes and a control unit's VAT groups. You can map up to four sales tax codes. In the following example of a configuration, **VAT10** and **VAT20** represent sales tax codes that must be mapped.

```
<UnitConfiguration>
  <TaxMapping>
    <Tax taxCode="VAT10" controlUnitTaxId="1"/>
    <Tax taxCode="VAT20" controlUnitTaxId="2"/>
  </TaxMapping>
</UnitConfiguration>
```

You can also export a sample configuration by clicking **Export sample configuration** on the Action Pane.

- c. On the **Hardware profiles** page, select the hardware profile of the Hardware station that the POS is paired with and the control unit is connected to. On the **Fiscal register** FastTab, set the following fields:
    - In the **Fiscal register** field, select **Third-party driver**.
    - In the **Configuration** field, select the name of the fiscal register configuration that you just created.
2. Set up custom fields for receipt layouts, so that the control code and the manufacturing number of the control unit are printed on receipts:
    - a. On the **Language text** page, add two records for the captions of the custom receipt layout fields. In the appropriate fields, specify the language ID for the captions (for example, **sv-se**), the text ID (for example, **900001** and **900002**), and the caption text (for example, **Control code** and **Control unit ID**).
    - b. On the **Custom fields** page, add two records for the custom receipt layout fields. In the **Type** field, select **Receipt**. Specify names and captions for the custom receipt layout fields:
      - Control code:

- o **Name:** `FiscalRegisterControlCode`
  - o **Caption text ID:** The text ID that you specified for the control code field (**900001** in the preceding example)
  - Manufacturing number of the control unit:
    - o **Name:** `FiscalRegisterId`
    - o **Caption text ID:** The text ID that you specified for the control unit ID field (**900002** in the preceding example)
  - c. For sales receipt formats, in the Receipt format designer, in the **Footer** section of the receipt layout, add the fields for the specified captions (**Control code** and **Control unit ID** in the preceding example).
3. Update POS permissions groups and individual permission settings for store workers. To allow workers who are assigned to the permission group to skip the fiscal registration, select the **Allow skip fiscal registration** check box.

## Development environment

Follow these steps to set up a development environment so that you can test and extend the sample.

1. Extend the Hardware station component:
  - a. Open the Hardware station solution under `RetailSDK\SampleExtensions\HardwareStation`.
  - b. Find the `HardwareStation.Extension.FiscalRegisterSample.csproj` extension project, and compile it.
  - c. Find extension assemblies and configurations.
    - [Retail 7.3 and earlier](#)
    - [Retail 7.3.1 and later](#)
    - [Retail 10.0 and later](#)

Find the following files in `Extension.FiscalRegisterSample\bin\Debug`:

- The `Contoso.Commerce.HardwareStation.FiscalRegisterSample.dll` assembly
  - The `Contoso.Commerce.HardwareStation.FiscalRegisterSample.dll.config` configuration
  - The `Interop.CleanCash_1_1.dll` assembly
- d. Copy the files to a deployed Hardware station machine:
    - **Remote Hardware station:** Copy the files to the `bin` folder under the Microsoft Internet Information Services (IIS) Hardware station site location.
    - **Local Hardware station:** Copy the files to the Modern POS client broker location.
  - e. Find the configuration file for Hardware station's extensions.
    - [Retail 7.3 and earlier](#)
    - [Retail 7.3.1 and later](#)
    - [Retail 10.0 and later](#)
    - **Remote Hardware station:** The file is named `hardwarestation.shared.config`, and it's under the IIS Hardware station site location.
    - **Local Hardware station:** The file is named `HardwareStation.Dedicated.config`, and it's under the Modern POS client broker location.

f. Add the following section to the **composition** section of the config file.

- [Retail 7.3 and earlier](#)
- [Retail 7.3.1 and later](#)
- [Retail 10.0 and later](#)

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.Extension.FiscalRegisterSample" />
```

g. Restart the Hardware station service:

- **Remote Hardware station:** Restart the Hardware station site from IIS Manager.
- **Local Hardware station:** End the **dllhost.exe** process in Task Manager, and then restart Modern POS.

2. Extend the CRT component:

a. Open the CRT solution, **CommerceRuntimeSamples.sln**, under **RetailSdk\SampleExtensions\CommerceRuntime**.

b. Find the **Runtime.Extensions.FiscalRegisterReceiptSample** project, and build it.

c. Find the ext.config file for CRT:

- **Retail Server:** The file is named **commerceRuntime.ext.config**, and it's under the **bin\ext** folder under the IIS Retail server site location.
- **Local CRT on Modern POS:** The file is named **CommerceRuntime.MPOSOffline.Ext.config**, and it's under the **bin\ext** folder under the local CRT client broker location.

d. Register the CRT change in the configuration file.

```
<add source="type" value="Contoso.Commerce.Runtime.FiscalRegisterReceipt, Contoso.Commerce.Runtime.FiscalRegisterReceipt" />
```

#### NOTE

Do **not** edit the **commerceruntime.config** and **CommerceRuntime.MPOSOffline.config** files. These files aren't intended for any customizations.

e. Find the **Contoso.Commerce.Runtime.FiscalRegisterReceiptSample.dll** assembly file in **Extensions.FiscalRegisterReceiptSample\bin\Debug**.

f. Copy the assembly to the CRT extensions folder:

- **Retail Server:** Copy the assembly to the **\bin\ext** folder under the IIS Retail server site location.
- **Local CRT on Modern POS:** Copy the assembly to the **\ext** folder under the local CRT client broker location.

#### NOTE

All the code changes for CRT and Retail Server are all part of **RetailSdk\SampleExtensions**. Therefore, the preceding steps show how to build, deploy, and test these code changes.

3. Extend the Modern POS component:

- a. Open the solution at **RetailSdk\POS\ModernPOS.sln**, and make sure that it can be compiled without errors. Also make sure that Modern POS can be run from Microsoft Visual Studio using the **Run** command. (Modern POS must not be customized. You must enable User Account Control [UAC], and you must uninstall previously installed instances of Modern POS as required.)
- b. Include **FiscalRegisterSample** in the **Pos.Extensions** project.
- c. Enable the extension to be compiled in **tsconfig.json** by removing the **FiscalRegisterSample** folder from the exclude list.
- d. Enable the extension in **Extensions\extensions.json** by adding the following lines in the appropriate place.

```
{
  "debugBuildsOnly": false,
  "baseUrl": "FiscalRegisterSample"
}
```

- e. Rebuild the solutions.
  - f. Run Modern POS in the debugger, and test the functionality.
4. Extend the Cloud POS component:
- a. Open the solution at **RetailSdk\POS\CloudPOS.sln**, and make sure that it can be compiled without errors.
  - b. Include **FiscalRegisterSample** in the **Pos.Extensions** project.
  - c. Enable the extension to be compiled in **tsconfig.json** by removing the **FiscalRegisterSample** folder from the exclude list.
  - d. Enable the extension in **Extensions\extensions.json** by adding the following lines in the appropriate place.

```
{
  "debugBuildsOnly": false,
  "baseUrl": "FiscalRegisterSample"
}
```

- e. Rebuild the solutions.
  - f. Run the solution by using the **Run** command and following the steps in the Retail SDK handbook.
  - g. Test the functionality.
5. Set up the fiscal register configuration and other required parameters in Headquarters. For more information, see [Cash register functionality for Sweden](#).

## Production environment

Follow these steps to create and apply deployable packages that contain Commerce components in a production environment.

1. Extend the POS component
  - a. Enable the extension to be compiled in **tsconfig.json** by removing the **FiscalRegisterSample** folder from the exclude list.



- b. Enable the extension in **Extensions\extensions.json** by adding the following lines in the appropriate place.

```
{
  "baseUri": "FiscalRegisterSample"
}
```

2. Make the following changes in the package config files under the **RetailSdk\Assets** folder:
- a. Add the following section to the **composition** section of the **commerceruntime.ext.config** and **CommerceRuntime.MPOSOOffline.Ext.config** config files.

```
<add source="assembly" value="Contoso.Commerce.Runtime.FiscalRegisterReceiptSample" />
```

- b. Add the following section to the **composition** section of the Hardware station configuration file.
- [Retail 7.3 and earlier](#)
  - [Retail 7.3.1 and later](#)
  - [Retail 10.0 and later](#)

Modify the **HardwareStation.Shared.config** and **HardwareStation.Dedicated.config** configuration files.

```
<add source="assembly" value="Contoso.Commerce.HardwareStation.Extension.FiscalRegisterSample"
/>
```

3. Make the following changes in the **BuildTools\Customization.settings** package customization configuration file:

- Add the following line to include the Hardware station extension in deployable packages:

```
<ISV_CommerceRuntime_CustomizableFileInclude="$(SdkReferencesPath)\Contoso.Commerce.HardwareStation.Extension.FiscalRegisterSample.dll"/>
```

4. Run **msbuild** for the whole Retail SDK to create deployable packages.
5. Apply the packages via Microsoft Dynamics Lifecycle Services (LCS) or manually. For more information, see [Retail SDK packaging](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Reporting and analytics with Power BI home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic points you to resources that you can use to learn more about the business intelligence (BI) and reporting tools that are available.

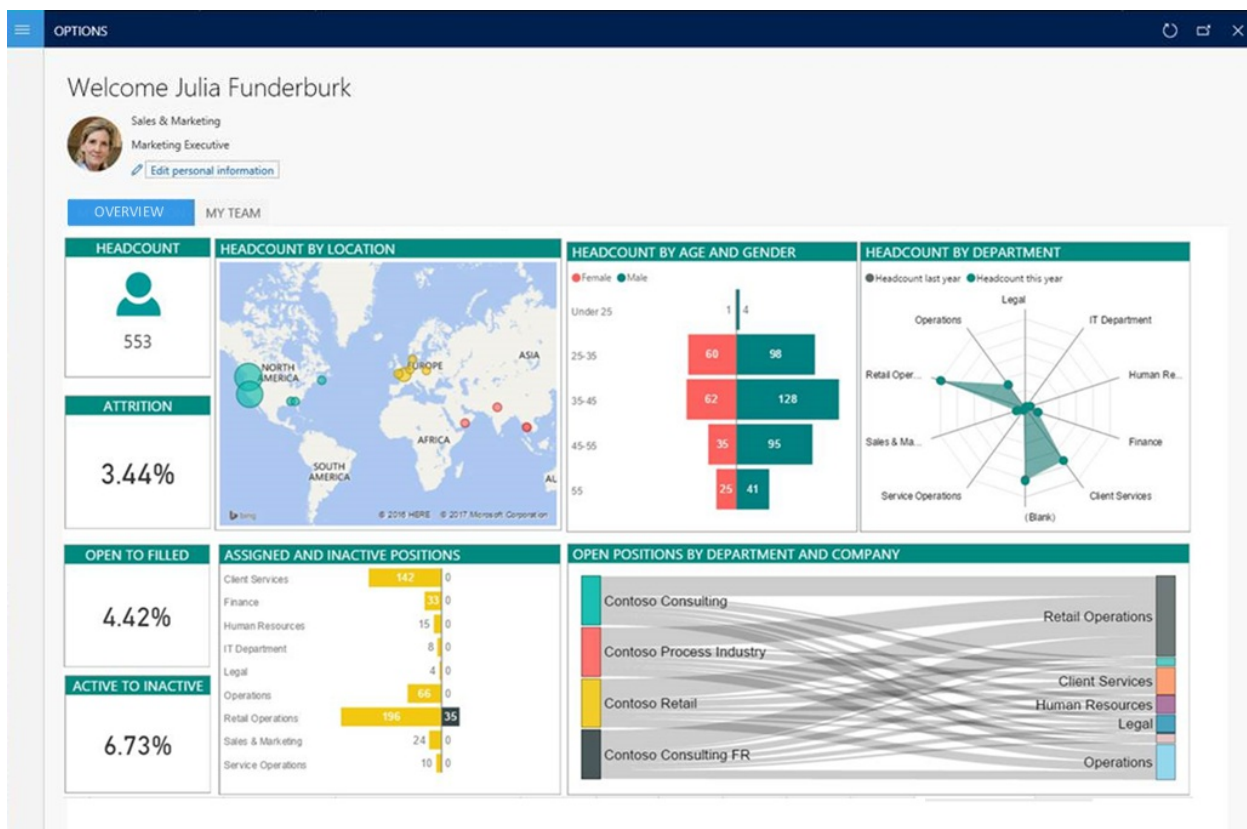
## Get started

- [Information access and reporting](#)
- [Tech Talk: Reporting options](#) (video)
- [Finance and Operations: Power BI Analytics & Reporting Services blog](#) (blog)

## Analytical workspaces

Workspaces can use rich infographics and visuals that are supported by Microsoft Power BI. These infographics and visuals include many controls that are provided by third parties. Therefore, workspaces can provide a highly visual, interactive experience for users.

Users can interact with data by clicking or touching visuals on the page. They can see cause and effect, and do simple what-if operations without leaving the workspace. Thanks to stunning, interactive visuals, your users will have fun exploring data and discovering hidden trends.



To learn more, see the following topics:

- [Embedded Power BI in workspaces](#)
- [Power BI Embedded integration](#)
- [Add analytics to workspaces by using Power BI Embedded](#)
- [Help secure analytical workspaces and reports by using Power BI Embedded](#)

- [Power BI content home page](#)

## Business documents and printing

Reporting solutions are often used to capture and communicate the details of business transactions. Therefore, a reporting solution must be able to produce physical manifestations of business data by using existing devices, such as network printers. Examples of business documents include sales invoices, customer statements, and checks.

The image shows a configuration window for a 'Customer account statement' on the left, which is linked by a blue arrow to a sample sales invoice on the right. The configuration window includes sections for 'Parameters', 'Criteria', 'Maturity Distribution', and 'Destination'. The sales invoice is a Microsoft document for 'Ciel Wholesale' with a total amount of \$315,479.00. It contains a table of items and various payment and contact details.

ITEM	DESCRIPTION	QUANTITY	SALES PRICE	DISCOUNT	AMOUNT	
DX001	Mid Range Speaker	14	Each	480.00	0.00	6,720.00
UX001	Mid Range Speaker 2	35	Each	500.00	0.00	17,500.00
FX001	Acoustic Foam panel	117	Each	37.00	0.00	4,329.00
DX002	Standard Speaker	23	Each	230.00	0.00	5,290.00
TX001	Speaker cable 10	65	Each	500.00	0.00	32,500.00
DX004	High End Speaker	12	Each	2,000.00	0.00	24,000.00
TX004	Television M720 37" Silver	53	Each	350.00	0.00	18,550.00
TX002	Projector Television	23	Each	3,750.00	0.00	86,250.00
TX005	Television HDTV X590 52" White	33	Each	2,890.00	0.00	95,370.00
TX003	Surround Sound Receiver	56	Each	450.00	0.00	25,200.00
SALES SUBTOTAL AMOUNT						315,479.00
SALES TAX						0.00
<b>USD TOTAL</b>						<b>\$315,479.00</b>

User parameters

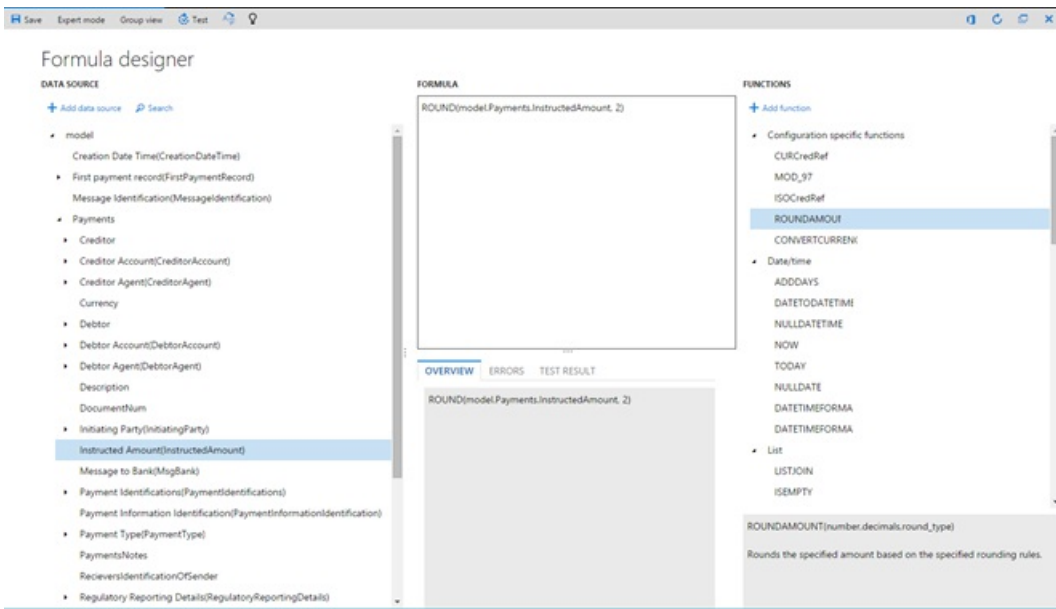
Paginated document

To learn more, see the following topics:

- [Document Reporting Services](#)
- [Document printing overview](#)
- [Install the Document Routing Agent to enable network printing](#)

## Electronic reporting

Electronic reporting (ER) is the tool that you use to configure electronic document formats that comply with the legal requirements of various countries or regions. The applications of electronic reporting include financial auditing, tax reporting, and electronic invoicing.

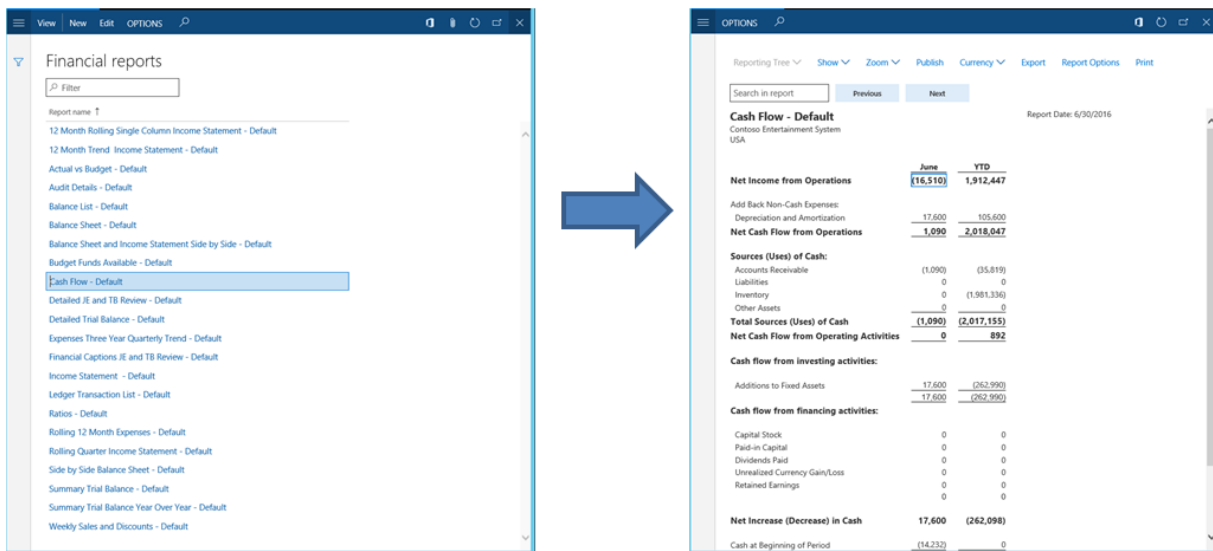


To learn more, see the following topics:

- [Electronic reporting \(ER\) overview](#)
- [MManage the Electronic reporting \(ER\) configuration lifecycle](#)
- [Create Electronic reporting \(ER\) configurations](#)

## Financial reporting

Standard financial reports are provided that use the default main account categories. You can use the report designer to create or modify traditional financial statements, such as income statements and balance sheets. You can then share the results with other members of your organization. Examples of financial reporting include balance sheets, cash flow, and summary trial balance year over year.



**Standard templates (GL > Inquiries > Financial reports)**

**Interactive financial reports**

To learn more, see the following topics:

- [Financial reporting](#)
- [Generate financial reports](#)
- [Financial report components](#)

## Technical reference reports

The following reports provide reference information about the objects:

- [Find information about standard data entities](#)
- [License codes and configuration keys report](#)
- [SQL Server Reporting Services \(SSRS\) reports that are available](#)
- [Workflow types report](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Information access and reporting

2/18/2021 • 11 minutes to read • [Edit Online](#)

This topic explains the various reporting options available as part of the platform.

## Why information access is important

Information access is an essential part of an ERP solution and represents a significant portion of the user engagement with the system. Consider the numerous methods of capturing information related to daily activities and the level of investments required to manage the data. Employees depend on logical interpretations of massive amounts of data to stay on top of their daily tasks. Out of the box the platform provides a collection of reporting solutions to address the various information access needs of an ERP solution. In an increasingly competitive environment, mergers and acquisitions seem to occur as often as the change in seasons. More than ever before, businesses today are finding ways to expand their global reach to attract more customers. To be successful, they must adapt legacy solutions used to communicate with customers and prepare for the enforcement of regional regulatory requirements associated with new markets. Beyond the primitive functions provided by Microsoft Azure – compute, networking, storage, and authentication – the platform provides tools to manage resources for organizations that span in size from small businesses to global enterprise conglomerates. These tools are designed with flexibility in mind, to accommodate a dynamic world of business.

## What is a "report" in the application?

A *report* can be defined simply as a structured presentation of data. Under ideal circumstances, reports materialize data in such a way that it allows the user to make informed decisions. The application supports a broad spectrum of information access scenarios: cross company all-up financial reporting; analytical dashboards and tiles; electronically transferable funds; customer sales invoices; checks and tax documents; and many more. Examples of integrated report scenarios that involve the consumption of business data include:

- **Native controls** including list pages, grid controls, and chart visualizations.
- **Dashboards and workspaces** containing predefined and personalized views.
- **Financial reporting** providing all-up views across legal entities.
- **Structured documents** distributed internally to employees or externally to customers and vendors.

Although each of these scenarios at its core involves the presentation of structured business data, the process of accessing these reports and how the results are subsequently consumed varies greatly. Flexibility in the user tooling is essential for scenarios that involve data exploration. By contrast, layout precision is required for compliance with most regulatory documents. Given the diversity of information access scenarios it's understandable that not all reports are created equal. This topic is intended to help familiarize you with the various reporting options available as part of the platform.

## Common myths of reporting

To become a proficient *report maker*, it's often useful to let go of past inhibitions. The following section seeks to rebuke three common myths about reporting.

- **Myth #1: Operational reports require "real-time" data** To the contrary, there are relatively few reporting scenarios that require *real-time* results. And, in the grand scheme of things, taking a critical stance on a request for *real-time* views is recommended given the high development costs and potential heavy burden these solutions may incur on production environments.
- **Myth #2: The best tool is the one the developer is most comfortable using** Consider a customer

request for a report that allows them to monitor company's assets. In the past, a developer would build a static report displaying a list of inventory items with complex calculations relying entirely on the user to provide filters to sufficiently reduce the result set. This solution may function perfectly in developer environments with a reduced data set. However, this approach is prone to unnecessarily consume significant amounts of compute resources when utilized in production.

- **Myth #3: Developers are good at creating visually compelling designs** In reality, developers are the often the worst offenders when it comes to producing elegant design layouts that will appeal to the customer's aesthetic preferences. When it comes to analytical reports you're better off empowering users to both explore the data directly and share personalized views.

## Understanding report requirements

The best reporting solutions are designed with the expertise, daily functions, and information access needs of the target user in mind. The platform offers several tools designed to meet the functional requirements which are common across various reporting experiences. Without question, selecting the right tool that most effectively addresses the *need* requires a clear understanding of the customer experience. You can drastically increase your chances of delivering a complete and robust solution that fully satisfies customer requirements by simply asking the right questions. Here are some leading questions to ask when evaluating customer requirements for reporting solutions:

- **Get to know the user**
  - What is the proficiency of the target persona? Are they familiar with analytical tools like Microsoft Excel?
  - Does the user require a guided parameter experience to refine the dataset?
  - How frequently will the report be accessed?
- **Familiarize yourself with the data**
  - Are they looking for transactional, analytical, and/or predictive information?
  - Does the shape of the data change and if so, how often?
  - Will the report include data from external sources?
- **Determine how the results will be used**
  - Are you going to explore the data to gain insights?
  - How will the results be shared with others?
  - Is there a fixed document structure for the target output?

It's understandable that customers *want* a solution that aligns with the existing processes they are comfortable using. However, there's a lot you can learn through these leading questions used to discover what the customer actually *needs* to be successful in their task. Delight your customers by providing them with solutions that empower them to be more productive.

## Reporting experiences

Information access scenarios supported in applications can be broken down into five distinct reporting experiences. Specialized tools are provided to meet the complex and diverse reporting needs of various functions throughout the organization.

- **Operational views** – Designed to address the specific needs of a given business persona.
- **Business documents** – Static documents used to capture and exchange processed business data.
- **Analytical tools and visualizations** – Personalized presentations of logical calculations that allow the user to explore their data.
- **Electronic reporting** – Tool used to configure formats for electronic documents.

- **Financial reporting** – Designed to provide in-depth accounting management tools based on standard views of financial activities across legal entities.

## Scorecard

The following table can be used as a guide when choosing the right tool for the reporting solution.

MAKER	OPERATIONAL VIEWS	BUSINESS DOCUMENTS	ANALYTICAL TOOLS & VISUALIZATIONS	ELECTRONIC REPORTING	FINANCIAL REPORTING
Persona	Developer	Developer	Power user	Power user	Power user
Authoring tool	Visual Studio	Visual Studio	PowerBI.com PowerBI app	Excel	Management Reporter Designer
Time to market	Weeks	Weeks	Hours	Hours	Hours
Data sources	Entity DB OLTP	OLTP	Entity DB Azure Catalog	OLTP	OLTP
Effort	Days	Days	Minutes	Hours	Hours
VIEWER	OPERATIONAL VIEWS	BUSINESS DOCUMENTS	ANALYTICAL TOOLS & VISUALIZATIONS	ELECTRONIC REPORTING	FINANCIAL REPORTING
Target	Organization	Back Office	Power user	Power user	Finance officers
Data accuracy	Near real-time	Real-time	Near real-time	Real-time	Cached views
Personalization	Medium-Modeled	None	High - Free form designer	Low - Expressions	Medium - Modeled
Sharing	None	PDF export O365 export Email	Dashboards Reports Tiles	Excel export	Excel export
Printing	Screen captures	Local printer Network devices	Screen captures	Excel	Local printer
Automation	Auto-refresh	Batch integration	Scheduled refresh	Batch jobs	None
Scenarios	Monitoring	Transactions	Exploratory	Transactions	Accounting

### NOTE

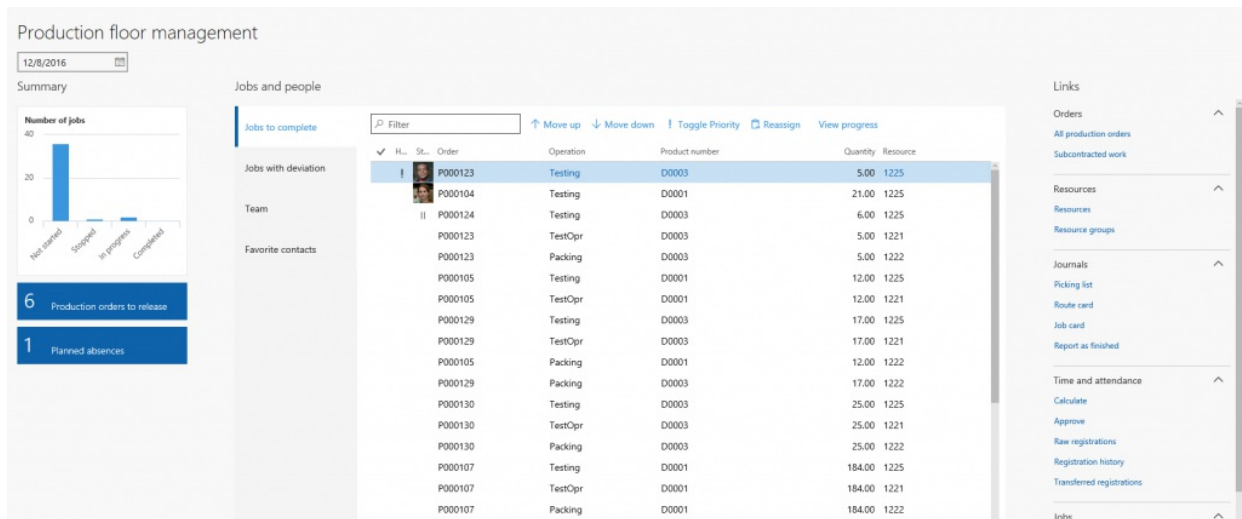
"Near real-time" denotes processed data that is slightly slower than real-time.

## Operational views

Operational views represent an essential part of the average employee's daily life. As important as a brush is to a painter, operational views are used to empower people to be productive. These views contain logical presentations of data to help the user discover patterns, highlight anomalies, and act on the most important



tasks. Targeted experiences are used to satisfy the unique information access requirements for a given persona. These views provide actionable controls that help to maximize efficiency for common user actions. Learn more about constructing custom operational workspaces in [Build operational workspaces](#). Example applications of operational views include controller operations, production floor management, and customer collections monitoring.



### What are the characteristics and capabilities?

- A fully-integrated experience with responsive visualizations fully-aware of user context and selections.
- Views can be personalized to a large extent, to meet the unique desires of the user that are prone to change.
- Actionable controls allow the user to efficiently transact and monitor activities.
- Combination of analytical data to help answer general questions and transactional views to access record details.

### What distinguishes "operational views" from other types of visualizations?

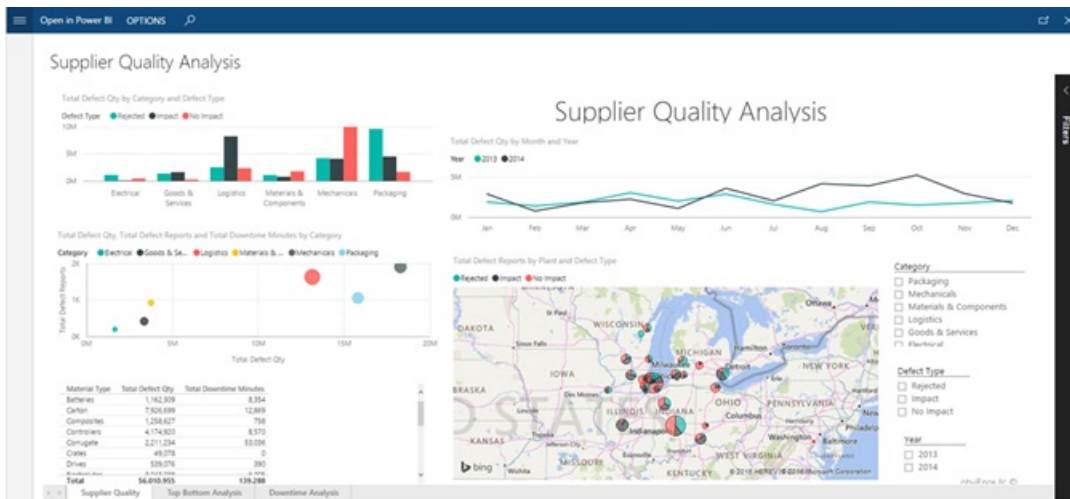
- General purpose tools designed to be utilized at all levels of the organization.
- Pre-defined views are provided based on common information access requirements associated with a specific role within the organization.
- Highly responsive to user interactions and changes made to the transactional database.

### What's important to consider when selecting this tool for the job?

- The platform allows users to embed Power BI tiles and links to reports directly in workspaces.
- Users can introduce personalized workspaces to create their own custom operational views.
- Form data sources now support aggregate queries for analytical views using native controls.

## Analytical tools and visualizations

Embedded visuals based on analytical data allow users to navigate between aggregate views down to the transactional details that affect them. Power BI service integration delivers world-class analytical tools with built-in support for accessing data. These tools empower "citizen developers" to author the reports they need and share the reports with others within the organization. Use the Power BI content packs available in Lifecycle Services to get started. Learn more about Power BI integration in [Features and services available through Power BI integration](#). Example applications of analytical tools and visualizations include customer sales per quarter, total revenue by region, and inventory turn-over.



### What are the characteristics and capabilities?

- Near real-time results that provide macro level insights based on micro level activities.
- Common applications include charts, KPIs, and more complex visuals.
- Offer a deep exploratory experience with interactive controls that provide drill-thru navigations.

### What distinguishes "analytical tools and visualizations" from other types of reports?

- Highly graphical in nature, these presentations are used to find the hidden meaning behind the data.
- Free form web designer that supports rich visualizations with built-in user interactions.
- Utilized by power users to explore data and gain insight through analysis.
- Personal in nature by allowing the user to choose which information to include.
- Built-in sharing capabilities and user controlled access management.

### What's important to consider when selecting this tool for the job?

- Developers are responsible for publishing data entities that can be consumed by Power BI.
- Power users can produce mash-up views based on application data combined with external data sources.
- Visuals are highly-responsive to user interactions and provide near real-time results when using Direct Query access to the data source.

## Business documents

These reporting solutions are often used to capture and communicate the details of business transactions. As such, this requires a reporting solution capable of producing physical manifestations of business data using existing devices like network printers. Learn more about the enhancements to the Document reporting service in [Document Reporting Services](#). Example applications of business documents include sales invoice, customer statements, and checks.

Customer account statement

Parameters

Use print management destination  
No

Show credit limit  
No

CRITERIA

From date

To date

Show due until

Only open  
No

Include reversed  
No

Associated payment attachment on acc...  
None

Go with sum  
No

Balance other than zero  
No

Show payment schedule  
No

Destination  
[Change](#)

MATURITY DISTRIBUTION

Show maturity distribution  
No

AGING PERIOD DEFINITIONS

Aging period definition

Print period description  
No

MANUAL SETUP OF MATURITY DISTRIBUTION

Internal

Day/Mth

Day

Printing direction  
Forward

OK Cancel



Microsoft  
123 Second Street  
Ballwin City, MO 63005  
USA  
Telephone 0123456789  
www.microsoft.com

Invoice CIV-000576  
30 November 2012  
Payment terms: Net 45 days  
Payment due 14 January 2013  
**\$315,479.00**

Our Wholesaler  
123 Violet Road  
Phoenix AZ 85003  
USA

ITEM	DESCRIPTION	QUANTITY	SALES PRICE	DISCOUNT	AMOUNT
00001	Mid Range Speaker	14 Each	400.00	0.00	5,700.00
00001	Mid Range Speaker 2	25 Each	500.00	0.00	12,500.00
00001	Acoustic Foam panel	117 Each	37.00	0.00	4,329.00
00003	Standard Speaker	23 Each	220.00	0.00	5,060.00
10001	Speaker cable 10	65 Each	500.00	0.00	32,500.00
00004	High End Speaker	12 Each	2,000.00	0.00	24,000.00
10004	Television M120 37" Silver	53 Each	350.00	0.00	18,550.00
10002	Projector Television	23 Each	3,750.00	0.00	86,250.00
10005	Television HDTV X590 52" White	33 Each	2,890.00	0.00	95,370.00
10002	Surround Sound Receiver	56 Each	450.00	0.00	25,200.00
SALES SUBTOTAL AMOUNT					315,479.00
SALES TAX					0.00
<b>USD TOTAL</b>					<b>\$315,479.00</b>

This text is from the Sales Invoice form notes

METHODS OF PAYMENT

Electronic payment  Check   
 Payment reference US-009  Make check payable to Microsoft  
 Settlement  Write reference US-009 on reverse of check  
 Account No. 34567

OTHER INFORMATION

Tax registration number 1234567890  
 Our reference Karl Nyström

Page 1 of 1

## User parameters

## Paginated document

### What are the characteristics and capabilities?

- Paginated documents that are ultimately destined to be printed on paper or distributed via email.
- Heavily dependent on parameters to filter and produce the desired result set.
- Business documents capture a snap-shot of customer and vendor activity that can be archived for future reference.
- The complex solutions are developed in Visual Studio and deployed as part of the application.

### What distinguishes "business documents" from other types of visualizations?

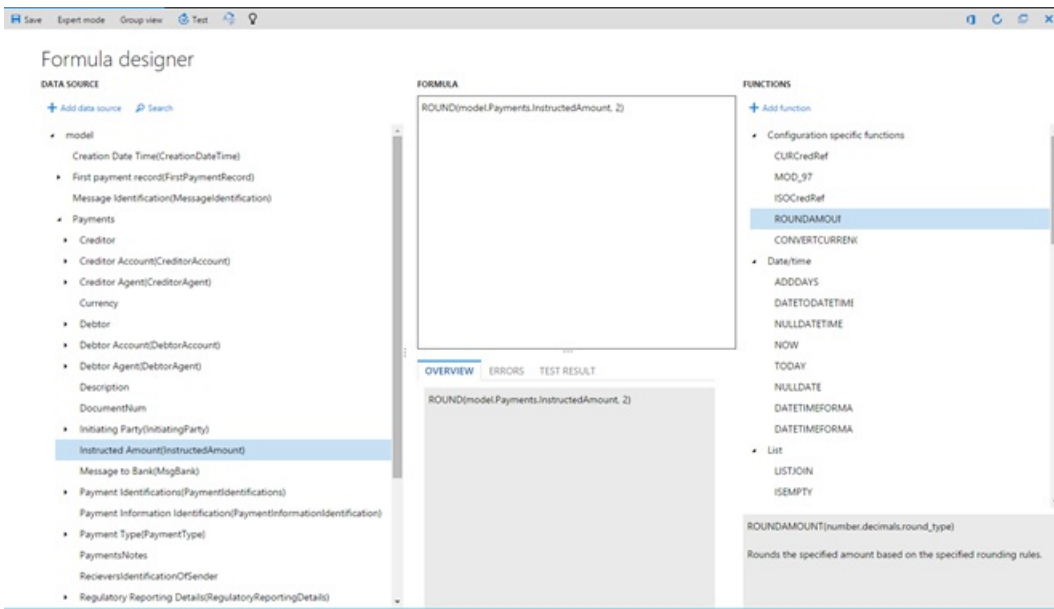
- Asynchronous data access and rendering solution designed to handle relatively large data sets.
- Dedicated reporting services offer distributed resource utilization.
- Ideal solution for automated processes that involve bulk generation of business documents.
- Includes built-in support for document archive and data extraction via file export to PDF in addition to Word, Excel, and CSV.

### What's important to consider when selecting this tool for the job?

- Application suite reports are intended to be used as a starting point for custom solutions.
- Solutions are heavily dependent on metadata changes and do not offer personalization.
- Modifications to out-of-box solutions must be managed as a metadata change.

## Electronic reporting

Electronic reporting (ER) is the tool to use to configure electronic document formats in accordance with the legal requirements of various countries/regions. For more information about the Electronic reporting tool, see [Electronic reporting \(ER\) overview](#). Example applications of electronic reporting include financial auditing, tax reporting, and electronic invoicing.



### What are the characteristics and capabilities?

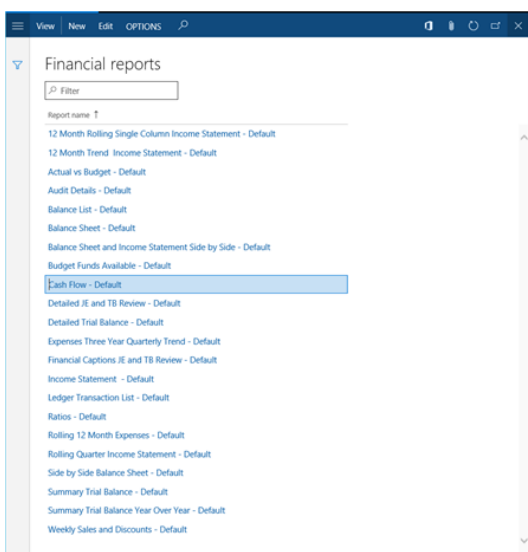
- Perfect tool for producing TEXT, XML, and OPENXML worksheet formats.
- Tooling is designed for business users familiar with Excel-based formulas.
- Highly adaptable to adhere to changes in regulatory requirements.
- Component versioning is available to manage draft definitions.

### What distinguishes "electronic reports" from other types of visualizations?

- Designed for electronic submission to banks, governments, and other external entities.
- Use formulas to define data transformations into groups containing summary data and logical calculations.

## Financial reporting

Standard financial reports are provided using the default main account categories. Use the report designer to create or modify traditional financial statements, such as Income statement and Balance sheet and share the results with other members of your organization. For detailed information about the Financial reporting tooling, see [General ledger and Financial reporting overview](#). Example applications of financial reporting include balance sheets, cash flow, and summary trial balance year over year.



	June	YTD
<b>Net Income from Operations</b>	<b>16,510</b>	<b>1,912,447</b>
Add Back Non-Cash Expenses:		
Depreciation and Amortization	17,600	105,600
<b>Net Cash Flow from Operations</b>	<b>1,090</b>	<b>2,018,047</b>
<b>Sources (Uses) of Cash:</b>		
Accounts Receivable	(1,090)	(35,819)
Liabilities	0	0
Inventory	0	(1,981,336)
Other Assets	0	0
<b>Total Sources (Uses) of Cash</b>	<b>(1,090)</b>	<b>(2,017,155)</b>
<b>Net Cash Flow from Operating Activities</b>	<b>0</b>	<b>892</b>
<b>Cash flow from investing activities:</b>		
Additions to Fixed Assets	17,600	(262,990)
	17,600	(262,990)
<b>Cash flow from financing activities:</b>		
Capital Stock	0	0
Paid-in Capital	0	0
Dividends Paid	0	0
Unrealized Currency Gain/Loss	0	0
Retained Earnings	0	0
<b>Net Increase (Decrease) in Cash</b>	<b>17,600</b>	<b>(262,098)</b>
Cash at Beginning of Period	(14,232)	0

Standard templates (GL > Inquiries > Financial reports)

Interactive financial reports

### What are the characteristics and capabilities?

- Built-in flexible financial reporting solution designed to handle complex organizational structures.
- Fully-integrated with General ledger.
- Create custom financial reports using the default solutions as a starting point.
- Interactive reports with drill-down capabilities to navigate down to transaction details.

#### **What distinguishes "financial reports" from other types of visualizations?**

- User controls are tailored for the specialized needs of financial reporting.
- Create roll-up reports containing data across companies or business units.
- Utilizes a financial data mart for optimized performance.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Features available through the Power BI integration

2/18/2021 • 7 minutes to read • [Edit Online](#)

Microsoft Power BI is a suite of business analytics tools that let you analyze data and share insights. By using Power BI tools, you can explore data and quickly create rich reports and dashboards. You and your colleagues can then use the reports interactively on many devices. The application uses Power BI for data exploration.

## Data exploration through Power BI

There are various types of reporting.

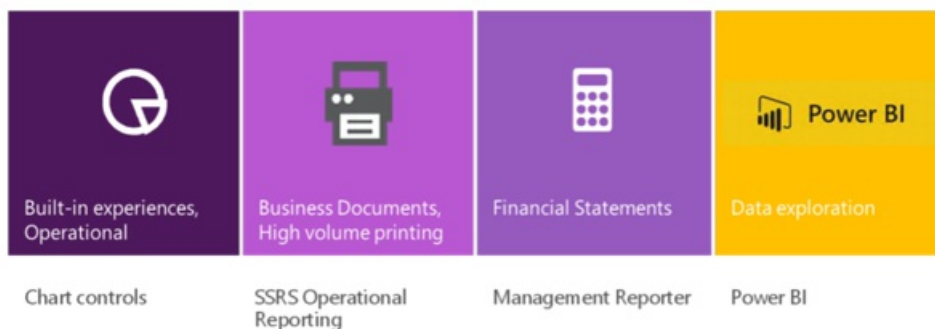


Chart controls are used to build embedded experiences that require visuals.

Microsoft SQL Server Reporting Services (SSRS) is an engine that is designed for pixel-perfect, formatted reports that often require printing. We use SSRS for document-style reports, such as invoices and purchase orders. Our investments in SSRS integration focus on document generation and printing scenarios.

For all non-document reports or reports that don't have to be printed, we want to embrace Power BI.

In the past, we have used the terms *self-service reports* and *ad-hoc reports* to refer to Power BI. We will now use the term *data exploration*. This change in terminology reflects a subtle shift in paradigms. Self-service reports were reports that the users themselves created. (Alternatively, a power user created the reports and shared them with other people. These people then continued to adjust the reports according to their requirements.) Often, users had to change the shape of a chart, add a new column, change the grouping, or just create a new view of data. Although we might think of the results as reports, users are just trying to understand the data by exploring it, pivoting it around columns, and changing the shape of charts. Technologies of the past didn't let users interactively explore large volumes of data. Therefore, users had to create "reports," or many views of the same data.

In Power BI, thanks to the in-memory database technologies in Microsoft SQL Server, a report is a just the starting point for interactive data exploration. Charts in a Power BI report invite users to click them, visuals change shape interactively, and data can easily be filtered. Users can easily adjust existing reports and create their own views of the data. The reports can be shared, and teams can collaborate over data.

Therefore, although you might use the term *report* to refer to Power BI artifacts, you should think about the larger scenario that is involved. Your users are exploring data! Power BI is the tool of choice when you must address data exploration and visualization requirements.

For a detailed discussion of reporting concepts, see [Information access and reporting](#).

## Ready-made Power BI content

You can use ready-made Power BI reports right away. Two types of Power BI content are available:

- Power BI content that is available in Microsoft Dynamics Lifecycle Service (LCS)
- Power BI content packs that are distributed in the PowerBI.com marketplace

Depending on your version, you can use one of both types of content.

### **Power BI content that is available in LCS**

LCS is a service that can manage your environments. LCS is operated by Microsoft. Power BI reports are developed by using Entity store and are then distributed in LCS as implementation assets. In LCS, you will find not only content that is developed by Microsoft, but also content that is developed by independent software vendors (ISVs) and partners.

We will continue to release Power BI content that is based on Entity store. For information, see the [Roadmap](#).

### **Power BI content packs that are distributed in the PowerBI.com marketplace**

There are several Power BI content packs in the PowerBI.com marketplace. Although these content packs will continue to be supported until further notice, our future investments in content packs will be based on Entity store, and content will be released via LCS.

For more information about the content, see [Power BI content](#).

## **Extending, creating, and distributing Power BI reports**

You should use the ready-made Power BI content as a first step. You can modify ready-made reports or extend them by using capabilities that are built into PowerBI.com. In addition to modifying ready-made reports, you can extend them using Power BI authoring tools such as Power BI desktop. You can also create new reports. You can use several approaches to create new Power BI reports.

### **Creating high-volume, near-real-time "operational Power BI reports" by using Entity store**

Entity store is an operational data store that is built specifically for Power BI integration. To create high-volume, near-real-time Power BI reports that use Entity store, a business analyst or a developer can use Power BI desktop, which is the authoring tool for Power BI reports. Like other artifacts that developers create, these reports must be distributed to your users via LCS.

Reports that are created by using Entity store take advantage of DirectQuery technology. This technology enables reports to be created over large volumes of data. Reports that are created by using DirectQuery technology don't cache data in the PowerBI.com service. Instead, data is always stored in application.

For an overview of Power BI integration with Entity store, see [Power BI integration with Entity store](#).

If you're upgrading from Microsoft Dynamics AX 2012, you can upgrade cubes to aggregate measurements that use Entity store. You can then create Power BI reports by using Entity store. For more information, see [Migrate upgraded AX 2012 R3 sales cubes to the entity store](#).

### **Creating Power BI reports by using Excel**

In addition to using the Power BI desktop authoring tool, you can use "Power tools" that are incorporated into Excel to create visualizations. Your organization might have many users who already use Excel every day. For a quick "one-off" report, Excel might be the best option for these users.

There are several scenarios where you can use Excel:

- Export data from a page in Dynamics 365 into Excel. You can then use the Power View add-in that is built into Excel to visualize the data. The Excel workbook can be used as a stand-alone visualization. In addition, you can import the report into the PowerBI.com service.
- Use the Power Query extension in Excel to combine the data in another worksheet (or data that is imported from OData endpoints) with external data. You can visualize the resulting data by using Power View.

- Use the PowerPivot extension in Excel to ingest a larger amount of data into Excel.

Consider using Export to Excel functionality for ad-hoc "one-off" reports. If the reports will be shared with a group of users, you should consider using Entity store to create them.

## Sharing and using reports in PowerBI.com

PowerBI.com is a service that is offered by Microsoft. It lets you create dashboards and reports, and also enables collaboration with a group of users. Regardless of how you create your reports, you can share reports with users by uploading them into the PowerBI.com service. (This process is also known as *publishing*.)

After your reports are uploaded, your users can view, adjust, and explore them either on the web (when they are connected to the Internet at home or in the office) or by using apps on a device.

For more information about Power BI concepts, see the [Power BI documentation](#).

## Pinning Power BI content to the client

PowerBI.com can be used on its own as a reporting and dashboard solution for your organization or business unit. However, users can also pin tiles and reports from their own PowerBI.com accounts to workspaces. Power BI content in Dynamics 365 provides contextual insights that are related to business operations.

You can pin two types of objects from PowerBI.com: tiles in PowerBI.com dashboards, and reports.

Before you can pin Power BI content, you must configure Power BI in your Dynamics 365 environment.

### **One-time configuration of PowerBI.com integration**

Before you can pin tiles or reports, an administrator must configure integration with PowerBI.com in your environment. This configuration must be done only one time per environment. For instructions, see [Configure Power BI integration for workspaces](#).

### **Pinning PowerBI.com tiles**

Power BI tiles that are pinned to the Dynamics 365 client provide insightful visuals at a glance. They also let users open PowerBI.com for interactive analysis. For more information, see [Configure Power BI integration for workspaces](#).

### **Pinning PowerBI.com reports to workspaces**

As a power user, a business analyst, or a developer, you can use Power BI desktop to create reports that use Entity store. Not only are these reports rich and interactive, but your users can make changes without having to rely on another person.

Although the reports in PowerBI.com are powerful and interactive on their own, they can also be pinned into workspaces. Your users can pin reports to workspaces themselves. For more information about how to pin reports to workspaces, see [Pin Power BI reports to workspaces](#).

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Pin Power BI content

2/18/2021 • 3 minutes to read • [Edit Online](#)

Microsoft Dynamics 365 Finance uses Power BI for data exploration. This topic explains how you can pin full-page Power BI reports to workspaces to give your users an interactive data exploration experience.

This topic assumes that you're familiar with the feature that lets you pin Microsoft Power BI tiles to a workspace. For more information, see [Features and services available through Power BI integration](#). If you're a developer who is creating a workspace, to let users pin Power BI tiles to the workspace, embed the Power BI tile control.

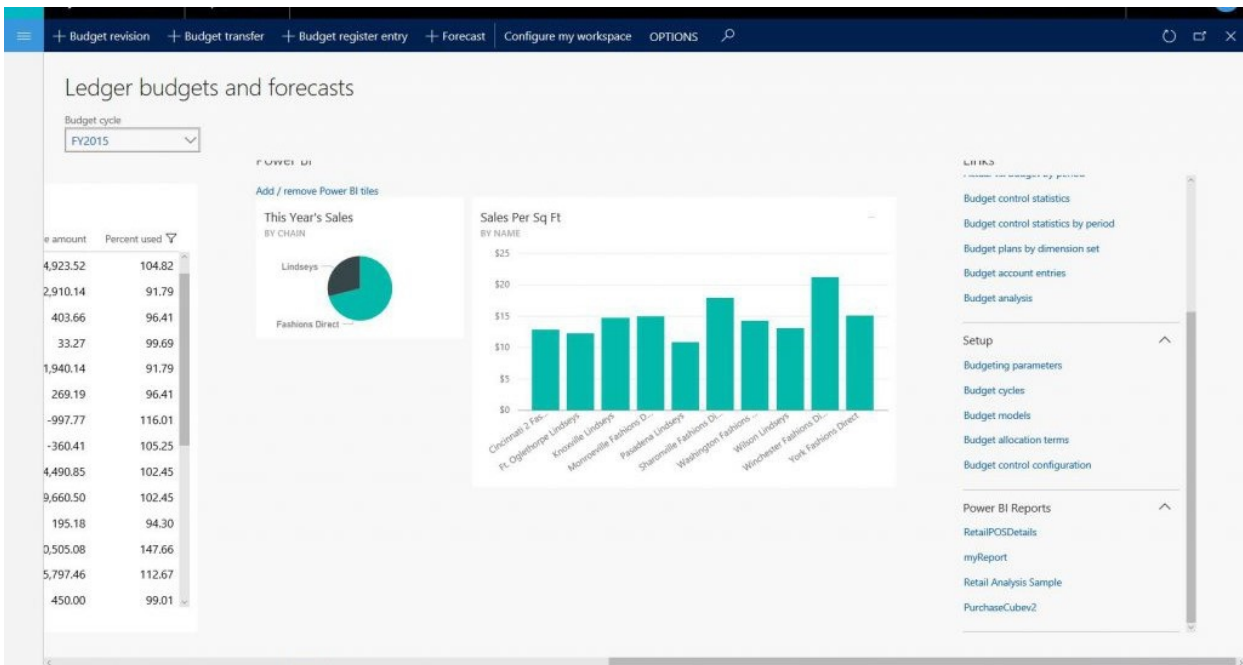
## Pin Power BI reports to workspaces

Microsoft Dynamics AX platform update 1 (May 2016) introduced the capability to pin Power BI reports to workspaces. Power BI reports can be added to any workspace that contains a **Links** section. In other words, the reports can be added to most of the out-of-box workspaces that are included in the product. To enable Power BI reports and tiles, you must configure Power BI to work with the application. This one-time operation must be completed by an administrator of the environment. For instructions, see [Configure Power BI integration for workspaces](#). After you've configured Power BI to work with the application, open the **Ledger budgets and forecasts** workspace in the client. In the workspace, click the **Options** tab. Notice that this tab contains buttons to open the (Power BI) tile catalog and the (Power BI) report catalog. Click **Open report catalog**. A dialog box that contains a list of reports appears. The list of reports comes from the reports that you have in your Power BI account. If you open PowerBI.com in a browser, you will see that the same list of reports is used across your Power BI dashboards. Select some reports, as shown in the following illustration, and then click **OK** to continue.

The screenshot displays the 'Ledger budgets and forecasts' workspace. The 'Options' tab is active, showing a list of reports in the 'Open report catalog' dialog box. The dialog box has a search filter and a list of reports with checkboxes. The following table represents the data shown in the 'Open report catalog' dialog box:

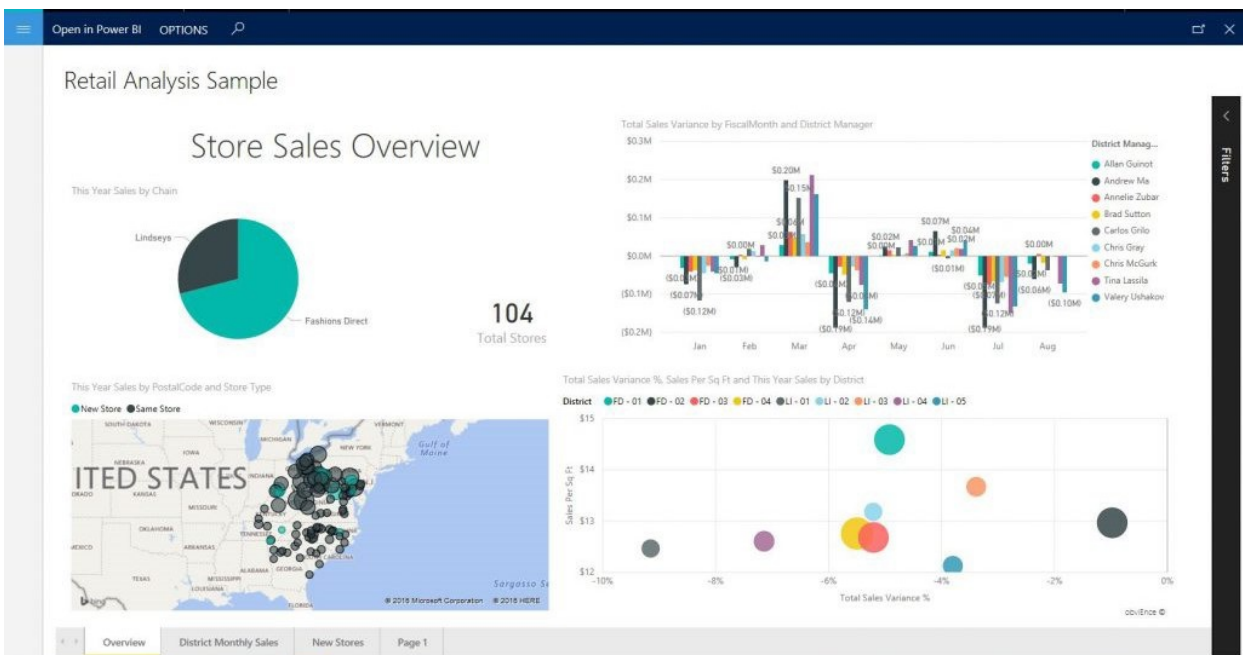
Report Name	Selected
Supplier Quality Analysis	<input type="checkbox"/>
Financial Performance Data Mo...	<input type="checkbox"/>
Retail Analysis Sample	<input checked="" type="checkbox"/>
myReport	<input checked="" type="checkbox"/>
RetailPOSDetails	<input checked="" type="checkbox"/>
RetailPOSDetails	<input type="checkbox"/>
CustomerReport	<input type="checkbox"/>
Retail Sales by Item	<input type="checkbox"/>
SalesInvoices-Import	<input type="checkbox"/>
Cost Management	<input type="checkbox"/>
Retail Channel Performance	<input type="checkbox"/>
VendorInvoices_PurchaseCubev...	<input type="checkbox"/>
Practice Manager BI Report - R1...	<input type="checkbox"/>
myLedgerReport	<input type="checkbox"/>
PurchaseCubev2	<input checked="" type="checkbox"/>
Workforce	<input type="checkbox"/>

Next, scroll to the bottom of the **Links** section in the workspace. Notice that a new section for Power BI reports has been added to your links.

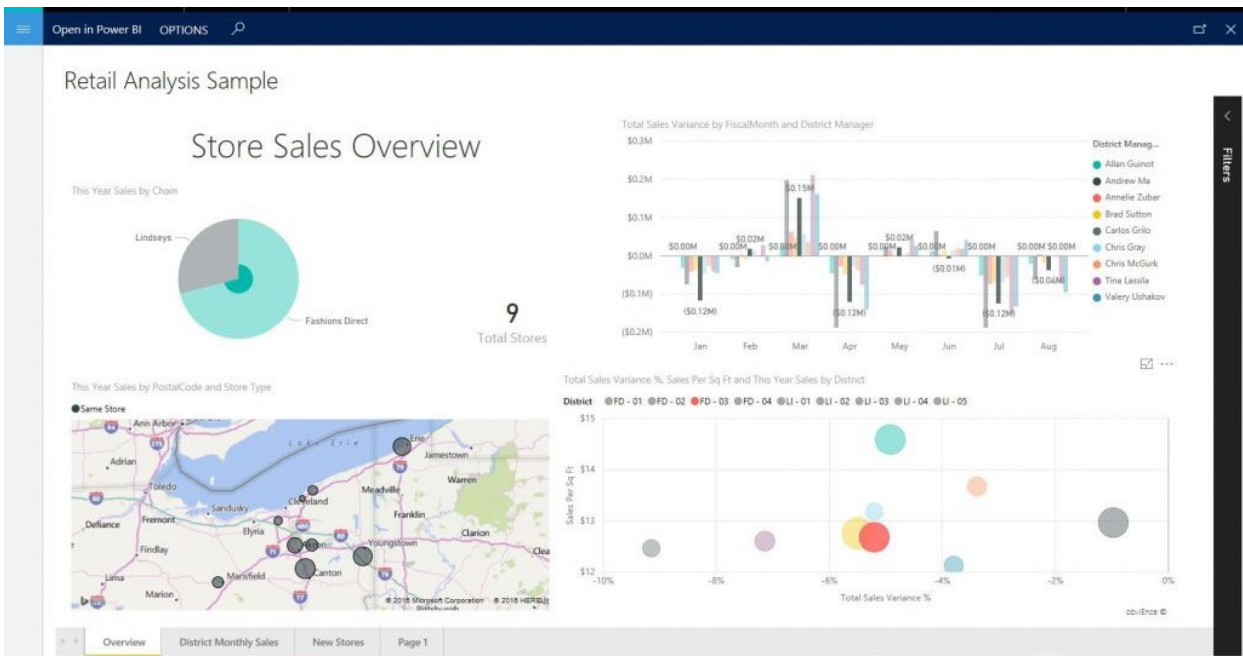


## Full-page Power BI reports in the client

You can open and run Power BI reports in the client. The functionality resembles the functionality for running Microsoft SQL Server Reporting Services (SSRS) reports. To run a Power BI report, in the **Links** section, click the link for one of the Power BI reports. For this example, click the **Retail Analysis Sample** link. The Power BI report is opened in the client in a full-page view, as shown in the following illustration. This report is interactive. As you click regions of the report, the remaining visuals react to your selection.



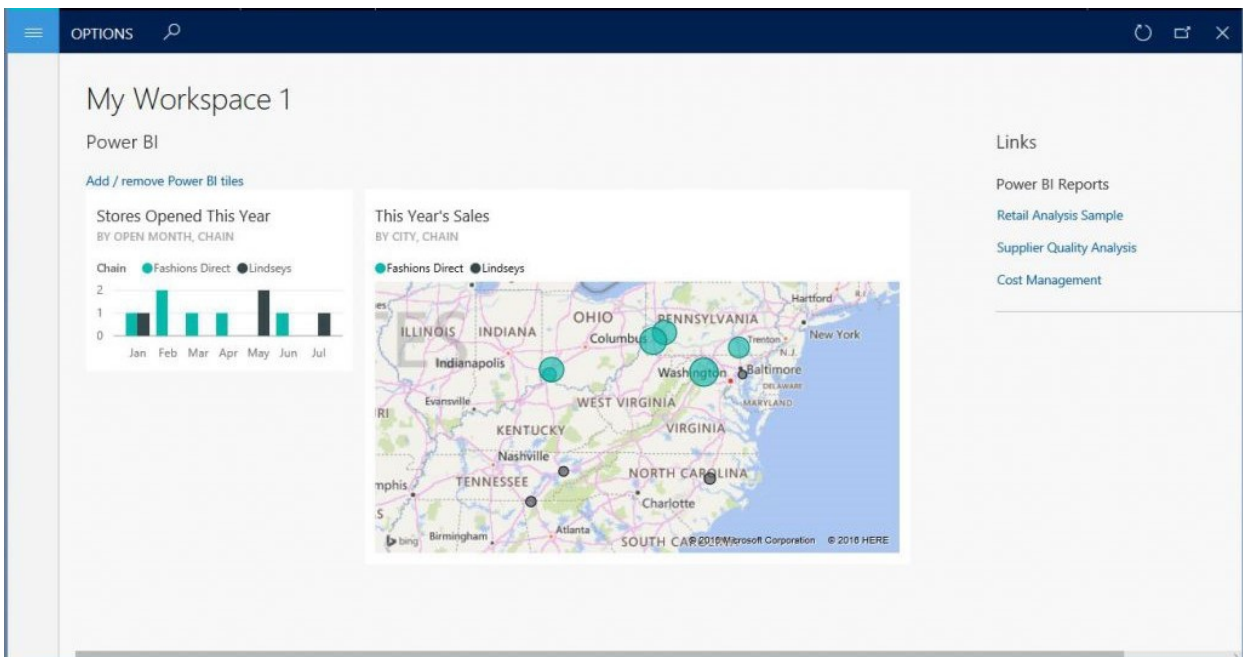
You can filter the data on the report by using the filter pane. The following illustration shows the report after filters have been applied.



You can also open this report on PowerBI.com and make changes. You can then save the modified report as another copy that has a different name, and even pin the new report to the workspace.

## Power BI in user-created workspaces

So far, we have described how to add Power BI tiles and reports to "developer-created" workspaces. Developer-created workspaces are workspaces that are created by Microsoft (that is, they are built into the product), by your independent software vendor (ISV) or partner, or by in-house developers. However, in Microsoft Dynamics AX platform update 1 (May 2016), users can create new workspaces by using the personalization capabilities of the client. To create a new workspace, on the home page (or the dashboard), right-click the tile for a workspace, and then click **Add a workspace**. A new workspace is created. New workspaces are named **My Workspace 1**, **My Workspace 2**, and so on. You can change the name later. Click the workspace that you just created. You can now add Power BI tiles and reports by using the same options that we discussed earlier. The following illustration shows an example.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Select analytical workspaces from Power BI

2/18/2021 • 3 minutes to read • [Edit Online](#)

## Analytical workspaces

The analytical workspaces that are bundled with the application suite offer users relevant insights into their business data. However, in some cases, it might make sense to replace standard reports with custom reports that are designed specifically for the users in your organization.

The world-class tooling that PowerBI.com provides lets you produce analytical reports that contain mashup views that use data from external sources. In Platform update 26 for Finance and Operations, power users can replace the standard embedded reports with those that are hosted on PowerBI.com.

### IMPORTANT

The functionality that this topic describes isn't a personalization. The customization of analytical workspaces applies to all users in the active legal entity.

### Motivations for embedding PowerBI.com reports

Although standard reports deliver insights that are tailored for a given business persona, an organization might prefer a custom report in some cases. The application lets power users promote custom reports that are hosted on PowerBI.com and shared with members of the organization.

Here are some of the top motivations for selecting reports that are hosted on PowerBI.com:

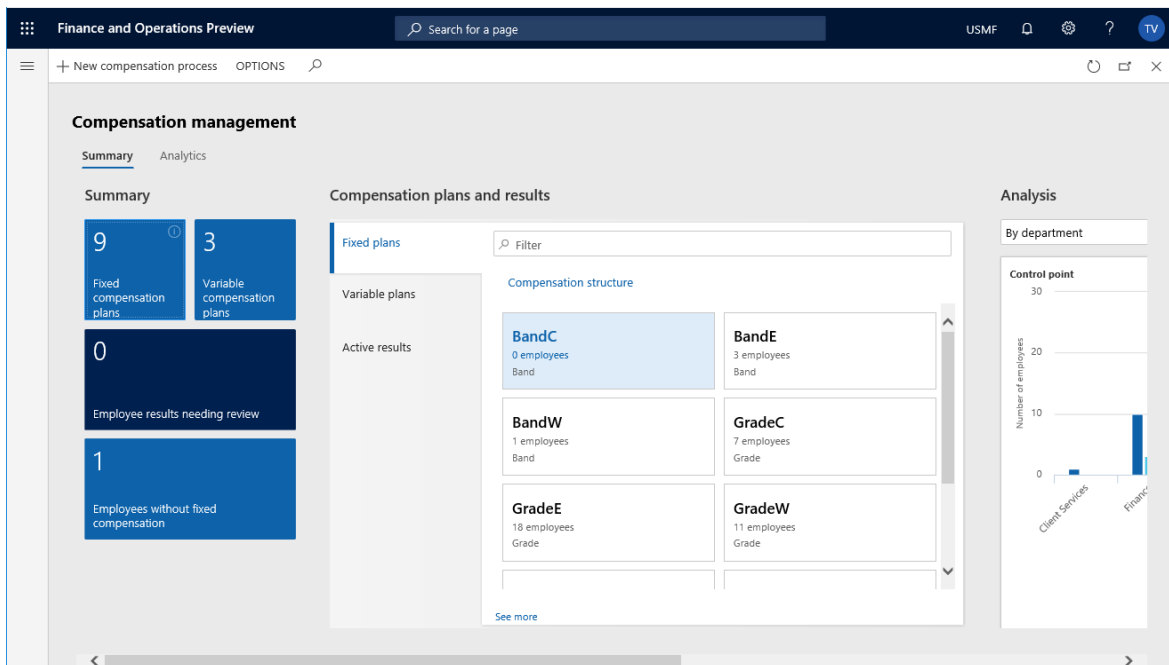
- PowerBI.com reports support data mashups that use external data sources and can be accessed outside the application.
- The reports are appropriate for demonstrating custom solutions that are hosted on PowerBI.com and embedded in the application in one-box deployments.
- Organizations that have Microsoft Power BI Premium services want to augment the standard reports.

### Embed a PowerBI.com report in an analytical workspace

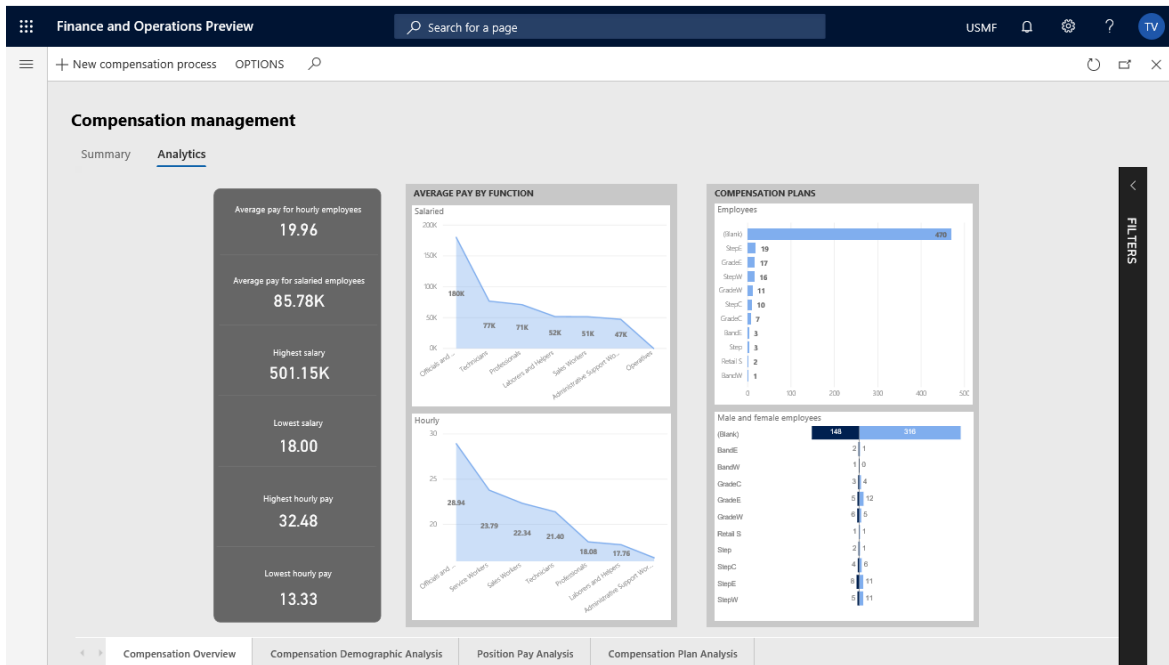
The [How to embed PowerBI.com reports](#) video (shown above) is included in the [Finance and Operations playlist](#) available on YouTube.

To replace the standard reports, you must be a member of the System Report Editors security group. Members of this security group can access the options in application workspaces that let them customize the standard reports. This example shows how to replace the standard analytical report with a customized report that is hosted on PowerBI.com.

1. Sign in and open the application report that you want to customize. In this example, you will replace the standard analytical report that is embedded in the **Compensation management** workspace.



2. Select the **Analytics** tab to access the workspace's embedded analytical report.

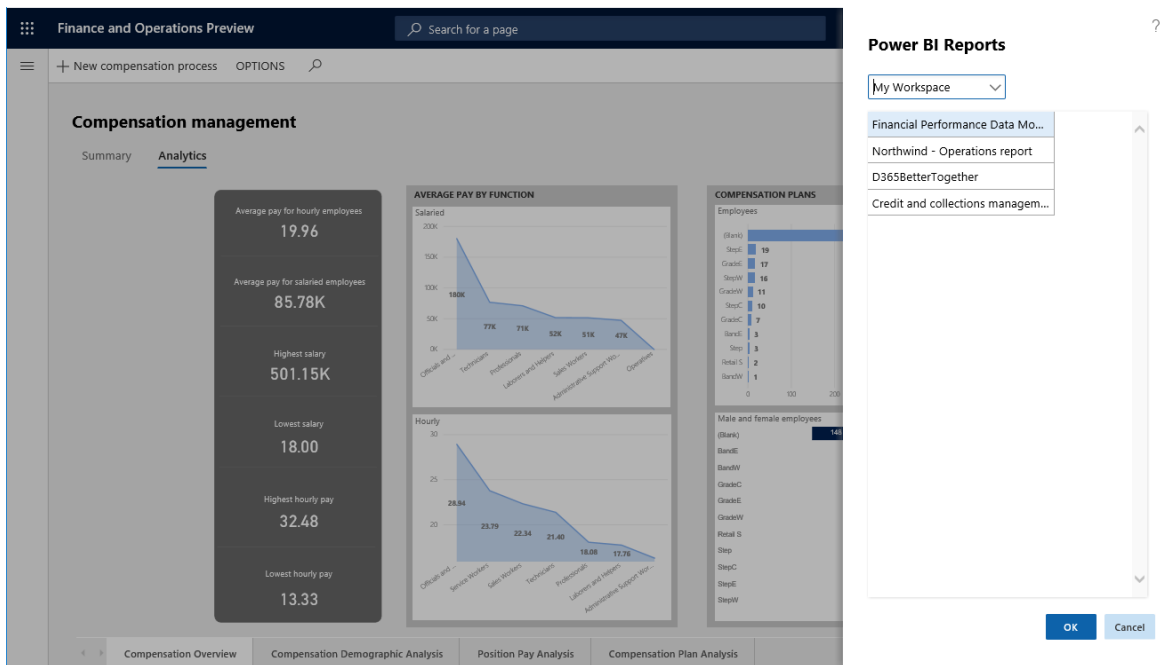


By default, you will see the standard analytical workspace solution that is included with your application. The reports in this solution are automatically deployed and configured for your environment during the provisioning process.

**NOTE**

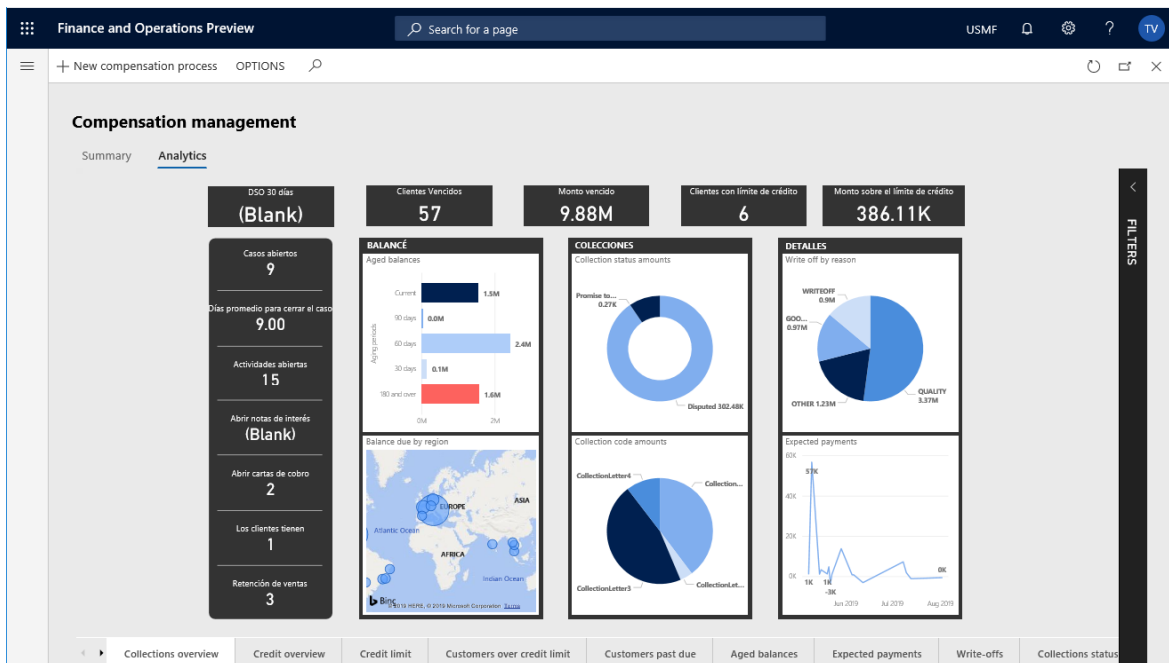
The analytical workspaces require a hosted Power BI service that is available only for dedicated environments. For more information, see the blog post, [Accessing Analytical Workspaces and Reports on 1-Box Environments](#).

3. On the Action Pane, on the **Options** tab, in the **Power BI** group, select **Select Analytics** to open the **Power BI Reports** dialog box.



This dialog box lets you select among the reports that have been shared on the PowerBI.com service. The reports are organized by workspace.

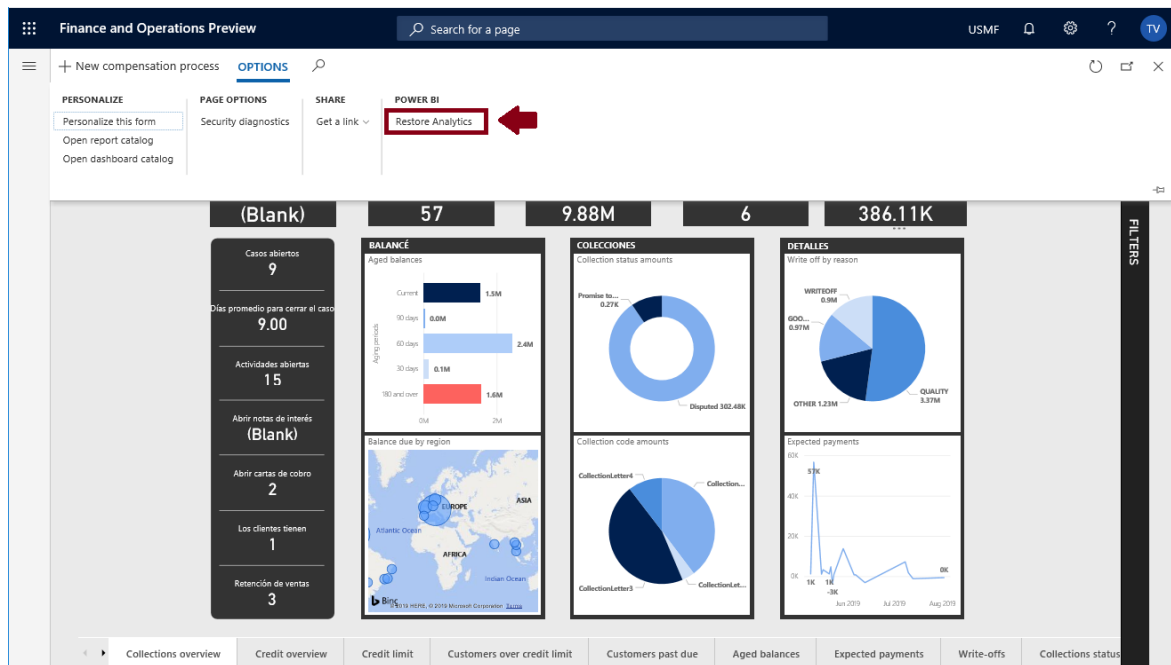
4. In the drop-down list, select the workspace that contains the report.
5. Select the report to embed in the application workspace, and then select **OK**.
6. To view the updates to the workspace, you must reload the page. Either move away from the workspace and then return, or refresh your browser.
7. In the **Compensation management** workspace, select the **Analytics** tab to access the PowerBI.com report that is now embedded in the analytical workspace.



### Revert to the standard solution

After a PowerBI.com report has been embedded in an application workspace, updates to the report are reflected immediately for users. However, to replace the report with another PowerBI.com solution, a power user must first revert to the standard application solution. Follow these steps to revert to the standard application solution.

1. On the Action Pane, on the **Options** tab, in the **Power BI** group, select **Restore Analytics**.



2. To view the updates to the workspace, you must reload the page. Either move away from the workspace and then return, or refresh your browser.
3. In the **Compensation management** workspace, select the **Analytics** tab to access the standard solution that is now embedded in the analytical workspace.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Power BI content home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following PowerBI.com solutions are available for application environments.

## PowerBI.com solutions available from Lifecycle Services (LCS)

The following PowerBI.com solutions are available in the Shared assets library of LCS. Refer to the version information in the Shared assets library to download the correct version for your environment.

### NOTE

Most of the PowerBI.com solutions in the following list have been embedded in analytical workspaces in recent versions. Using analytical workspaces eliminates the need to download these solutions from the Shared assets library in LCS. The topics in this list indicate where these PowerBI.com solutions have been embedded where applicable.

- [Actual vs budget](#)
- [Cash overview](#)
- [Compensation and benefits](#)
- [Cost accounting analysis](#)
- [Credit and collections management](#)
- [Employee competencies and development](#)
- [Financial performance](#)
- [Fixed asset management](#)
- [Organizational training](#)
- [Practice manager](#)
- [Production performance](#)
- [Purchase spend analysis](#)
- [Recruiting](#)
- [Sales and profitability performance](#)
- [Vendor payments](#)
- [Warehouse performance](#)
- [Workforce metrics](#)

## PowerBI.com solutions available from AppSource

The following PowerBI.com solutions are available from [Microsoft AppSource](#).

### NOTE

These solutions have been deprecated as documented in [Power BI content packs available on AppSource](#).

- [Cost management](#)
- [Financial performance](#)
- [Retail channel performance](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Learning Power BI content

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the **Learning** Microsoft Power BI content.

## Reports that are included in the Power BI content

The reports that are included in the **Learning** Power BI content have both charts and tables that contain additional information. The following table describes the reports.

REPORT	CONTENTS
Learning Overview	Summary of other reports
Course Analysis	Registration by location, attendee by status, courses by type per company, and course attendance by job
Registration Analysis	Registration list
Course Types	Course types by skill
Instructor Analysis	Ratio of courses to instructors, number of instructors, courses by instructor, courses per instructor, and course agenda by instructor
Courses Offered	List of courses
Courses Design	Course agenda

You can filter the charts and tiles on these reports, and pin the charts and tiles to the dashboard. For more information about how to filter and pin in Power BI, see [Create and Configure A Dashboard](#).

## Understanding the data model and entities

The following data is used to fill the reports in the **Learning** Power BI content. This table shows the entities that the content was based on.

ENTITY	CONTENTS	RELATIONSHIPS WITH OTHER ENTITIES
Calendar Offset	Calendar offsets to slice reports	Course Agenda, Course Attendees
Company	Companies to filter reports by	Course Agenda, Course Attendees
Course	Course, description, instructor name, location, room, and status	Course Agenda, Course Attendees, Course Skill
Course Agenda	Agenda, course, and start and end times	Company, Calendar Offset, Date, Course

ENTITY	CONTENTS	RELATIONSHIPS WITH OTHER ENTITIES
Course Attendees	Name, status, job, and registration date	Company, Calendar Offset, Date, Course, Demographics, Employment, Course, Employee Name, Employee Title, Job, Position
Course Skill	Skill, skill type, and level	Course
Date	Days, weeks, months, and years	Course Agenda, Course Attendees
Demographics	Date of birth, gender, ethnic origin, and marital status	Course Agenda, Course Attendees
Employment	Start date, end date, and transition date	Course Agenda, Course Attendees
Job	Function, type, and title	Course Agenda, Course Attendees
Position	Position, title, and full-time equivalent (FTE)	Course Agenda, Course Attendees
Employee Name	First name, last name, and full name	Course Attendees
Employee Title	Title and seniority date	Course Attendees

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Organizational training Power BI content

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the Finance and Operations - Organizational training Power BI content.

## Reports that are included in the content pack

After you've connected the content pack to your data, the reports show your organization's data. If you've never used Microsoft Power BI before, you can learn more about it on the [Guided Learning page for Power BI](#). The reports that are included in the content pack have both charts and tables that contain additional information. The following table describes the reports.

REPORT	CONTENTS
Course Analysis	Registration by location, course attendees by status, and registration list
Course Types	Course types by skill

You can filter the charts and tiles on these reports, and pin the charts and tiles to the dashboard. For more information about how to filter and pin in Power BI, see [Create and Configure A Dashboard](#).

## Understanding the data model and entities

Application data is used to populate the reports in the Organizational training content pack. The following table shows the entities that the content pack was based on.

ENTITY	CONTENTS	RELATIONSHIPS WITH OTHER ENTITIES
Training_CalendarOffset	Calendar offsets to slice reports	Training_CourseAgenda, Training_CourseAttendees
Training_Company	Companies to filter reports by	Training_CourseAgenda, Training_CourseAttendees
Training_Course	Course, description, instructor name, location, room, and status	Training_CourseAgenda, Training_CourseAttendees, Training_CourseSkill
Training_CourseAgenda	Agenda, course, and start and end times	Training_Company, Training_CalendarOffset, Training_Date, Training_Course
Training_CourseAttendees	Name, status, job, and registration date	Training_Company, Training_CalendarOffset, Training_Date, Training_Demographics, Training_Employment, Training_Course, Training_WorkerName, Training_WorkerTitle, Training_Job, Training_Position

ENTITY	CONTENTS	RELATIONSHIPS WITH OTHER ENTITIES
Training_CourseSkill	Skill, skill type, and level	Training_Course
Training_Date	Days, weeks, months, and years	Training_CourseAgenda, Training_CourseAttendees
Training_Demographics	Date of birth, gender, ethnic origin, and marital status	Training_CourseAgenda, Training_CourseAttendees
Training_Employment	Start date, end date, and transition date	Training_CourseAgenda, Training_CourseAttendees
Training_Job	Function, type, and title	Training_CourseAgenda, Training_CourseAttendees
Training_Position	Position, title, and full-time equivalent (FTE)	Training_CourseAgenda, Training_CourseAttendees
Training_WorkerName	First name, last name, and full name	Training_CourseAttendees
Training_WorkerTitle	Title and seniority date	Training_CourseAttendees

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail channel performance PowerBI.com solution

2/18/2021 • 3 minutes to read • [Edit Online](#)

## NOTE

This PowerBI.com solution has been deprecated as documented in [Power BI content packs available on AppSource](#).

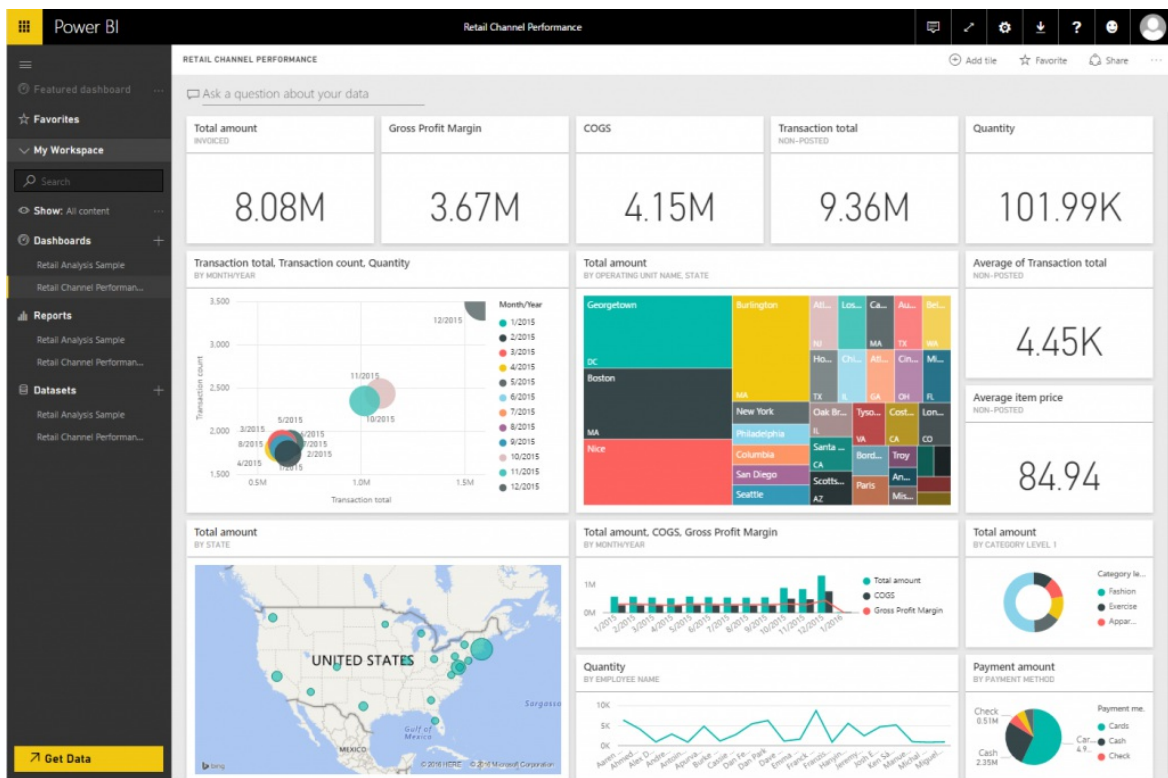
This topic provides information about the Retail channel performance PowerBI.com solution for Dynamics AX. This PowerBI.com solution lets channel managers quickly build channel performance analytics to predict trends and uncover insights, based on sales performance.

The Retail channel performance PowerBI.com solution lets you quickly build your channel performance analytics. The PowerBI.com solution is designed specifically for channel managers who focus on sales performance to predict trends and uncover insights. Its components draw directly from Retail and commerce data in the Dynamics AX database, and provide drill-down reports about organization-wide sales performance across global geography by employee, category, product, terminal, channel, and more. Power BI automatically creates reports and dashboards that give you a great starting point for exploring and analyzing your Retail and commerce data. This article includes the following information:

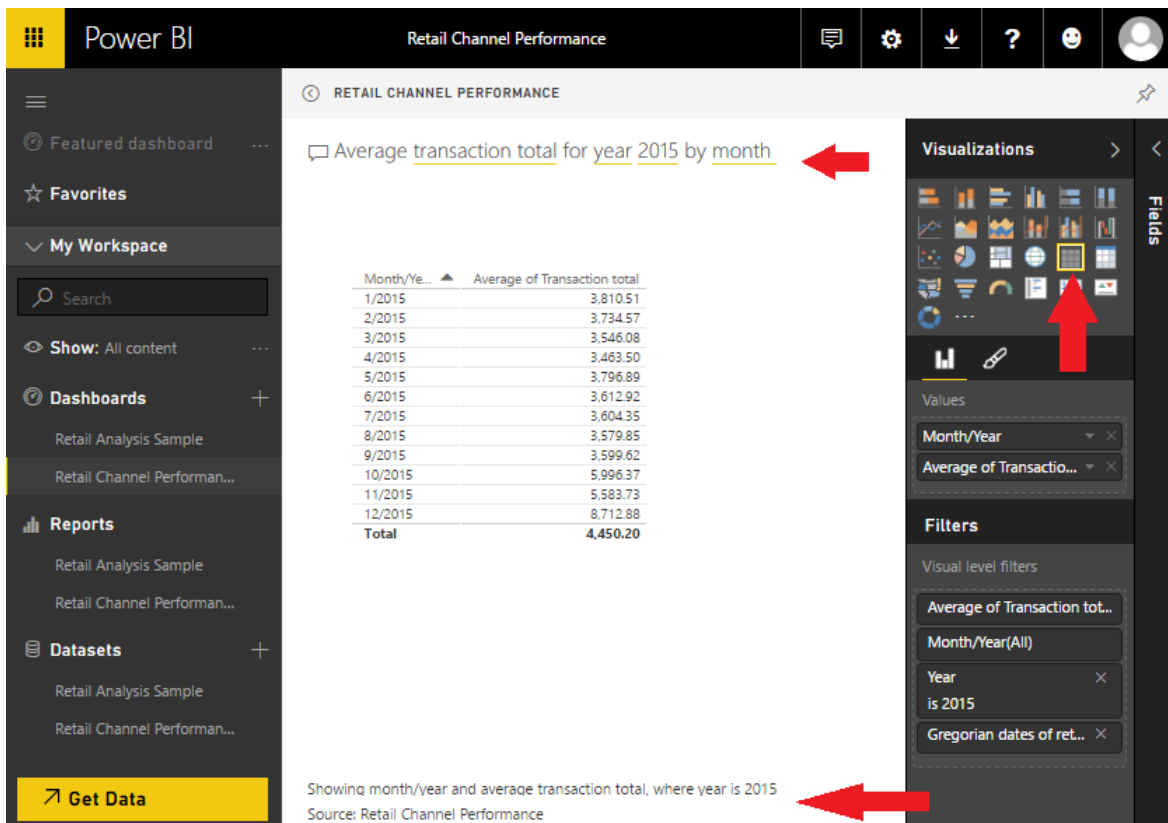
- Learn how to connect the Retail channel performance PowerBI.com solution to a Dynamics AX data source.
- View a list of reports that provide insights into retail channel performance.
- Learn how to modify an existing report in the PowerBI.com solution to make it self-authored.
- Get a glimpse of an actual data model that enables the whole experience in Power BI.

## Connect the Retail channel performance PowerBI.com solution to a Dynamics AX data source

1. Go to <https://www.powerbi.com>, and click **Sign in**. If you don't have an account, you can sign up to try the new Power BI Preview for free.
2. To sign in, enter a Microsoft 365 account that has a Power BI account.
3. If your workspace appears, click **Get Data** at the bottom of the left navigation pane.
4. In **Services** section, click **Get**.
5. Scroll or search to find **Microsoft Dynamics AX Retail channel performance**, and then click **Get it now**.
6. Enter your Dynamics AX URL in the following format: `https://<tenant>.cloudax.dynamics.com` (for example, `https://YourAOSTenant.cloudax.dynamics.com`). Then click **Next** to pull data from Dynamics AX data storage into this Power BI dashboard.
7. Select **oAuth2** as the authentication method, and then click **Sign in**.
8. To sign in, enter a Microsoft 365 account that has permission to access your Dynamics AX environment.
9. After data is successfully pulled from Dynamics AX into Power BI, you can view your personal **Retail channel performance** dashboard in Power BI by clicking **Retail channel performance dashboard** in the left navigation pane.



10. You can then take advantage of the Q&A feature in Power BI to query your Dynamics AX sales data by using natural language.



## View a list of reports

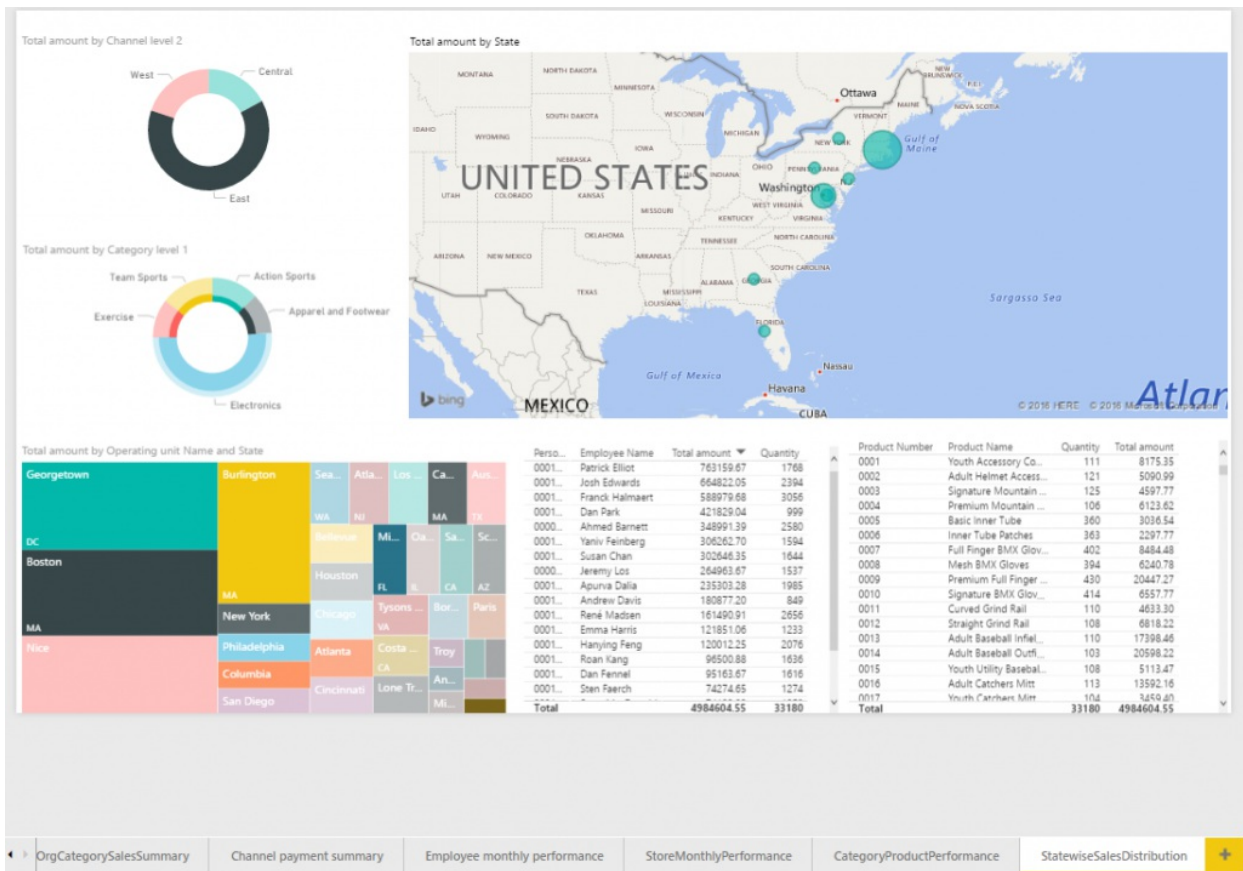
By clicking through any of the pinned tiles on the dashboard, you can navigate the following list of reports that provide insights into retail channel performance:

- Geographical sales distribution
- Category sales performance
- Sales summary by Tender type or payment method



- Employee monthly performance
- Store monthly performance
- Product sales performance for the given category in the given store

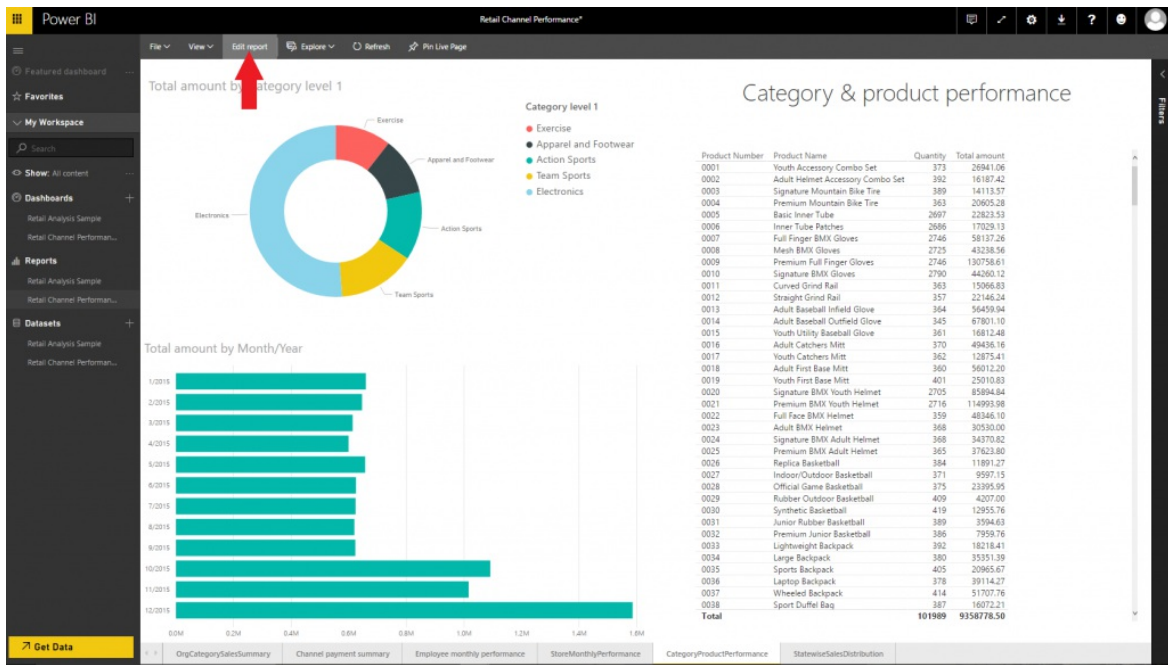
For example, you might want to do a deeper analysis of geographical sales distribution.



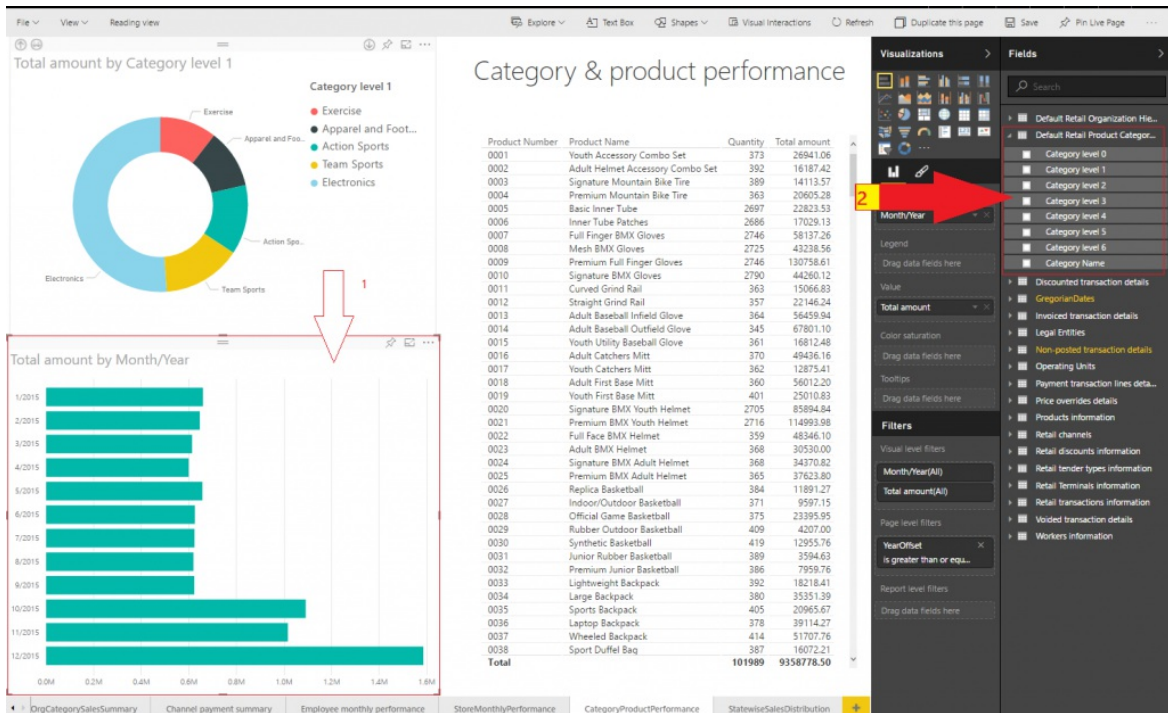
## Modify an existing report in the PowerBI.com solution to make it self-authored

Here's an example that shows how easy it is to modify an existing report in the PowerBI.com solution to make it self-authored. In this example, we will modify an existing report that is named **Category & product performance** by adding **Category level 1** to the **Total amount by Month/Year** chart on that report.

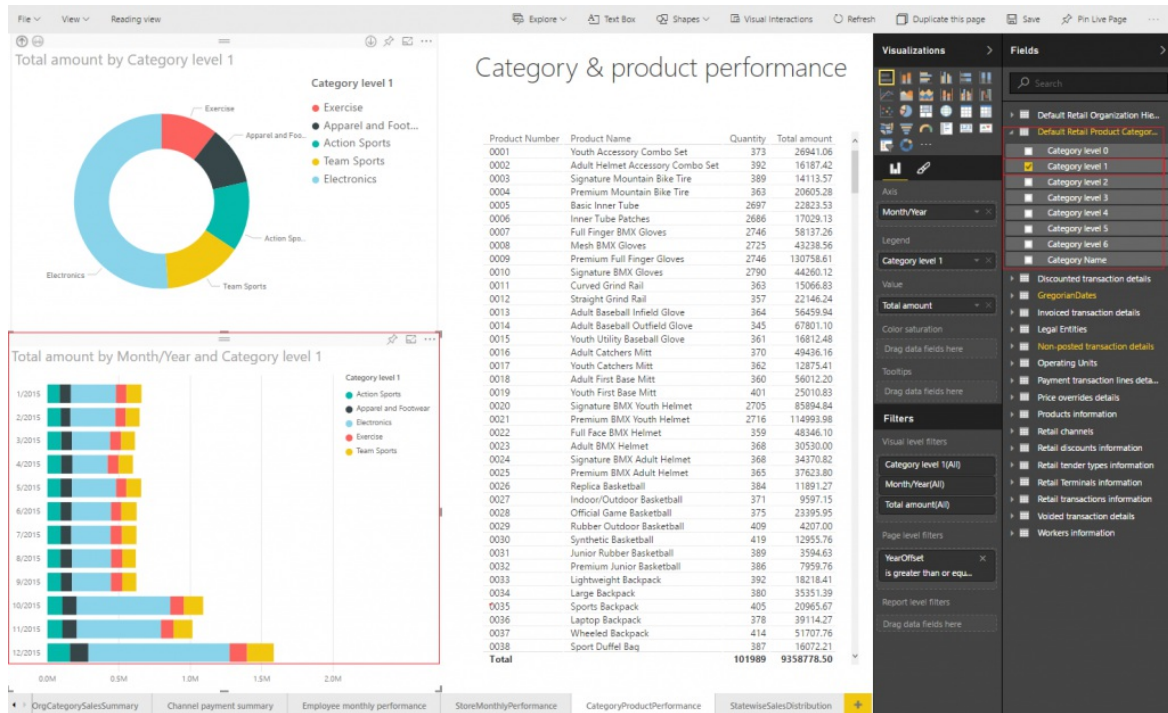
1. Click the **CategoryProductPerformance** tab at the bottom of the window to open the **Category & product performance** report, and then click **Edit report**.



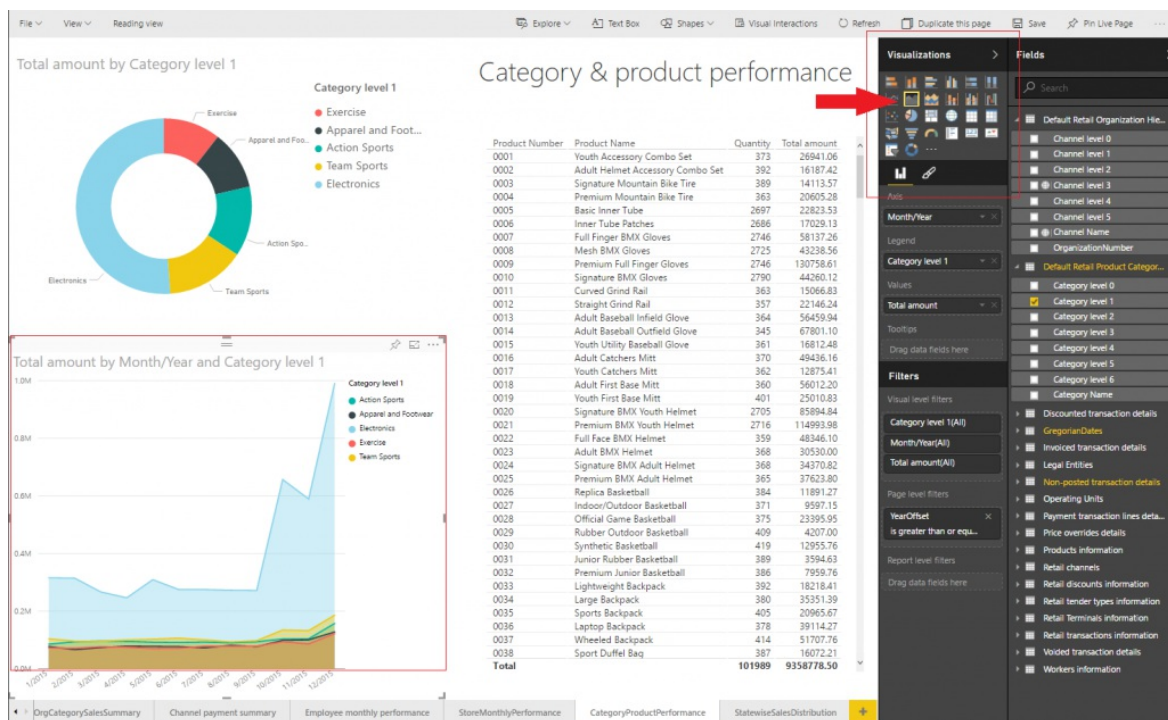
2. Select the chart that is named **Total amount by Month/Year**. Then, on the right side of the window, in the **Fields** pane, expand the **Default Retail Product Category Hierarchy** node.



3. In the list of category levels for this hierarchy, select **Category Level 1**. The name of the chart that you selected this attribute for changes to **Total amount by Month/Year and Category level 1**, and the chart now shows the share of sales in each category for each month.

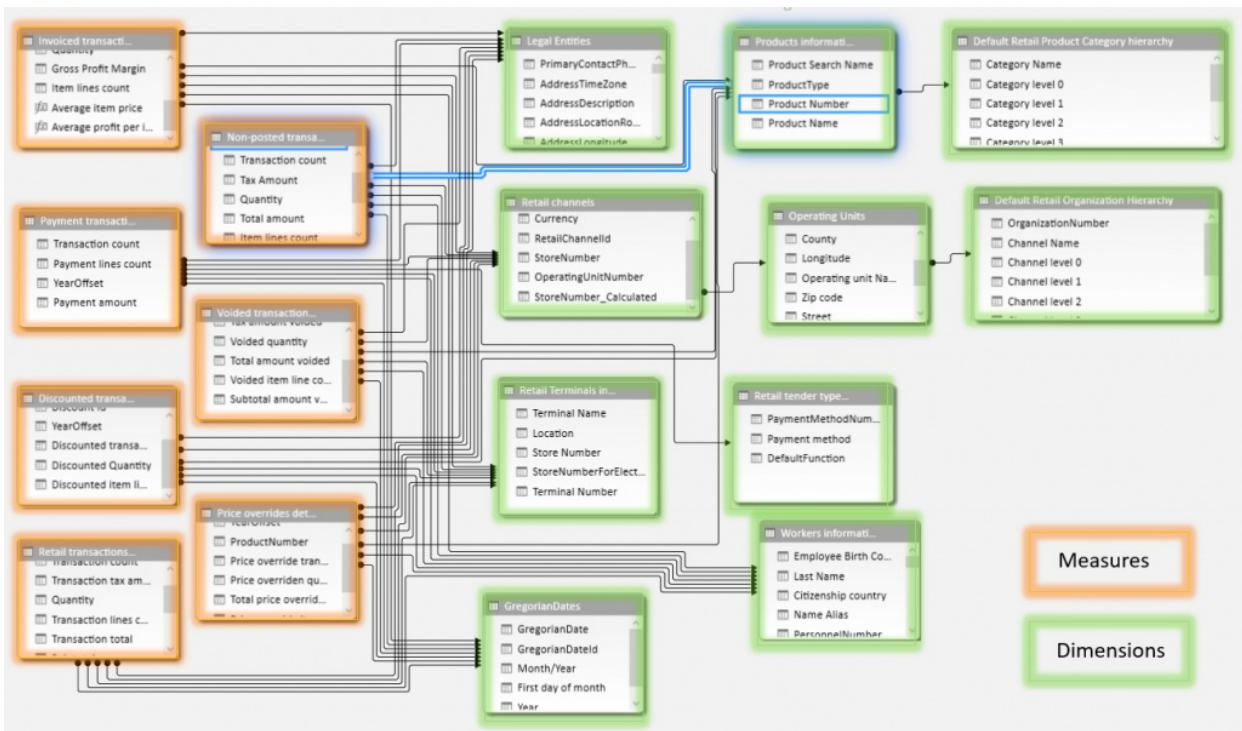


4. Finally, try to change the visualization itself. Select the Total amount by Month/Year and Category level 1 chart, and then, in the Visualizations pane, click Area chart or Stacked area chart, and see the effect.



## Get a glimpse of the actual data model

The data model that is included in the PowerBI.com solution for the Dynamics AX data entities and aggregated data entities lets you slice and dice various measures by using different dimensions.



## Additional resources

Features and services available through Power BI integration

Configure Power BI integration for workspaces

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Analytical Workspaces (using Power BI Embedded)

2/18/2021 • 2 minutes to read • [Edit Online](#)

Dynamics Finance and Operations apps now deliver rich, interactive reports seamlessly integrated into application workspaces. By using graphics and visuals supported by Power BI, workspaces can provide a highly-visual, yet interactive experiences for users.

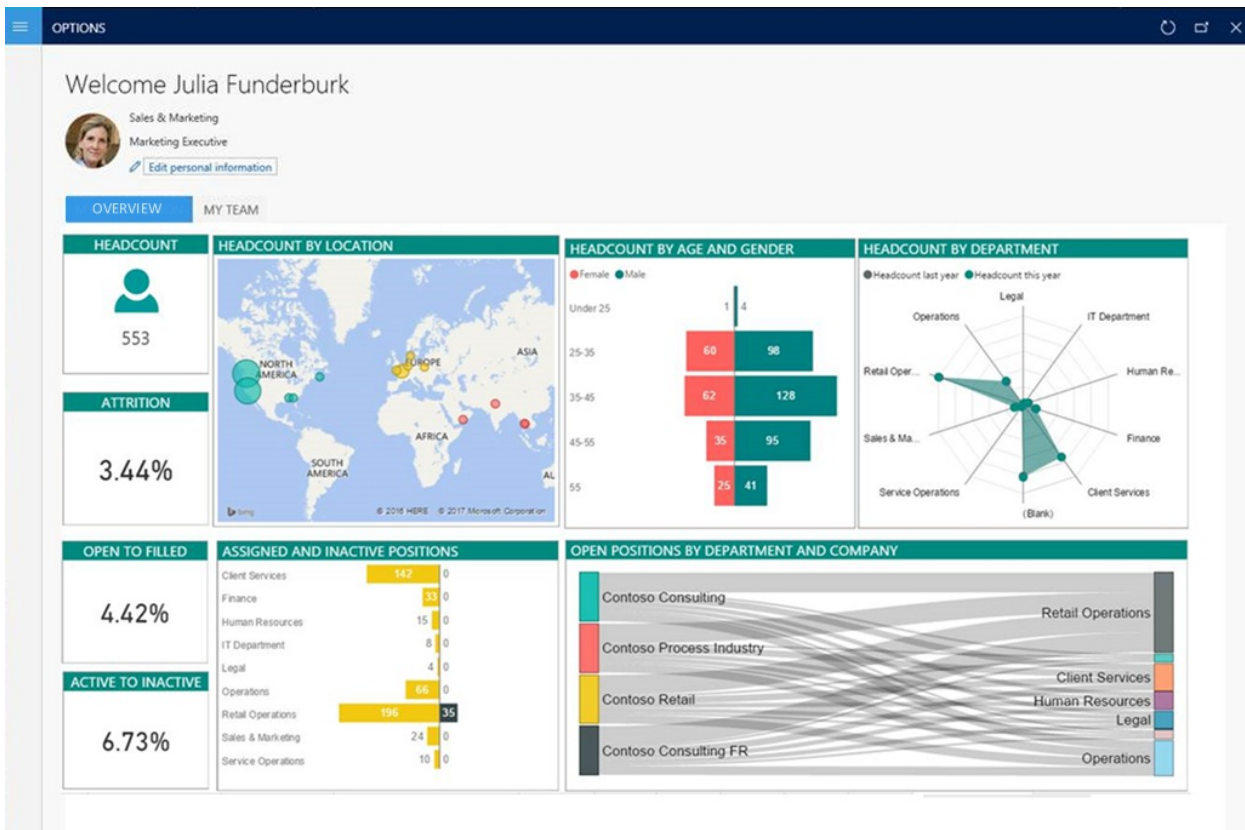
## Overview

Workspaces in the application provide an overview of business processes or business units. With rich workspaces, users can get a bird's-eye view of the state of business before diving into details and taking action. Workspaces contain visuals, count tiles, and KPIs as well as quick links to reports and pages. Within a workspace, all controls are tightly integrated to provide a highly-productive and engaging work environment to the user. Power BI Embedded is a Microsoft Azure service that enables ISVs and app developers to surface Power BI data experiences within their applications. With Power BI Embedded, developers can deliver always-up-to-date views with Direct Query. To learn more about how the Power BI Embedded service integrates with the application, see [Power BI Embedded integration](#).

## Power BI in workspaces

The application now delivers interactive reports that seamlessly integrate into application workspaces. By using rich infographics and visuals supported by Power BI (including the large number of controls provided by third parties), workspaces can provide a highly-visual, yet interactive experience for users. Using infographics in the overview page, users can get a quick glance of the state of the business. They can interact with data by simply clicking or touching visuals on the page. They can see the cause and effect, perform simple what-if operations without leaving the workspace. Thanks to stunning yet interactive visuals, your users will have fun exploring data and discovering hidden trends.

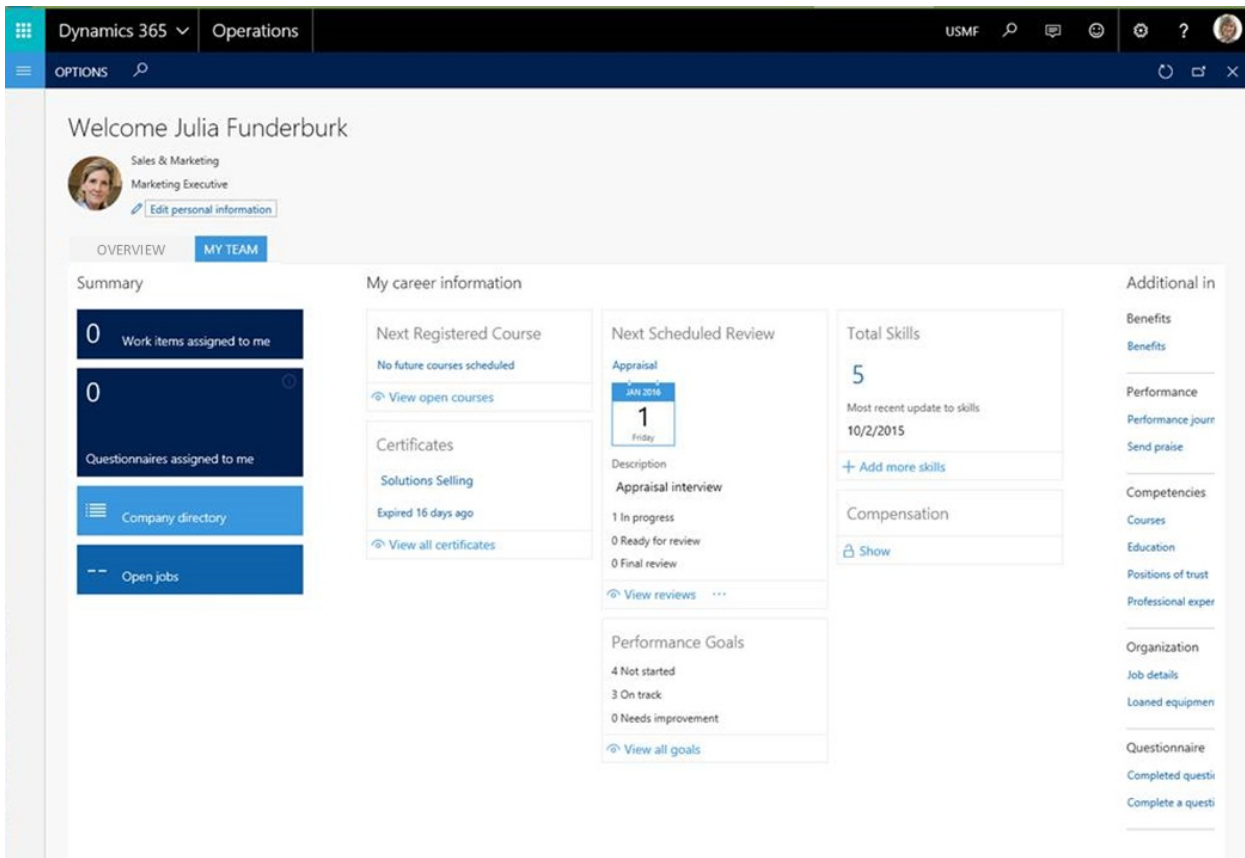
The following screenshot shows Power BI in a workspace.



## Power BI vs operational workspaces

Power BI workspaces complement operational views with analytical insights based on near real-time information. The following offers a visual comparison of a Power BI workspace and an operational workspace.

The following screenshot shows an operational workspace.



Edit embedded reports in analytical workspaces

The [How to edit an embedded report in an analytical workspace](#) video (shown above) is included in the [Playlist](#) available on YouTube.

## What's next?

Going forward, new cloud deployments will come bundled with the Power BI Embedded service. Additional documentation describing the Developer ALM process will be made available to help partners and ISVs create new solutions that take advantage of the Power BI Embedded service integration options that are available.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Power BI Embedded integration

2/18/2021 • 5 minutes to read • [Edit Online](#)

Microsoft Power BI content that partners and independent software vendors (ISVs) developed can be embedded directly into the Microsoft Dynamics 365 Finance. This topic describes some of the ways that you can use the Microsoft Power BI Embedded integration.

## Overview

The integration of the application and [Power BI](#) enables data mash-up scenarios that require access to external data sources that are supported through Microsoft Power Query for Excel. Users can personalize workspaces by embedding tiles that are hosted on PowerBI.com. Users can also add direct links to reports that are hosted on PowerBI.com. In this way, users can access and interact with the reports without leaving the application. Power BI content (PBIX files) that partners and ISVs develop can be embedded directly into the application. PBIX files that are associated with a model file are automatically published in Power BI Embedded as part of the application deployment process. Additionally, you can add X++ extensions for embedded reporting scenarios that require the following functionality:

- Drill-down navigation into detailed pages in response to user interactions
- Report filters that are based on user and session context information, such as company or date range
- The ability to navigate directly to a specific tab on a Power BI report via menu items

For more information about customizations that use extensions, see [Customize through extension and overlaying](#).

## Advantages of Power BI Embedded

- **Deliver Power BI workspaces and reports in the application.** If you're a power user or a business analyst, you can tweak ready-made reports or create new reports by using Power BI tools. As a developer, you can use the reports that your users create to provide rich navigation experiences in the product through workspaces. If you're in the partner and ISV community, you can build rich workspaces that include Power BI experiences, and then release those workspaces as part of your solution.
- **The Power BI Embedded service license is bundled with the application.** If you're an ISV or a systems integrator, you can package workspaces that are enabled for Power BI (and navigational experiences that those workspaces provide) as part of a Microsoft Dynamics Lifecycle Services (LCS) solution. Your customers get the same experience without having to have a PowerBI.com subscription. The workspaces just work with Finance and Operations applications.
- **Enable drill-down into detailed pages from Power BI.** The visuals are the starting point for action. Your users can drill down to business processes and pages to act immediately on issues that they uncover. The visuals let users filter data and uncover trends. Action pages reflect just the set of data that requires attention.
- **Help secure access to Power BI reports by using menu items.** As a developer, you can use familiar programming concepts that are available in Finance and Operations apps, because we have extended the same concepts to workspaces that are based on Power BI. You can create new workspaces or extend existing workspaces by adding an overview page that is driven by Power BI. Developers can associate menu items with Power BI reports and include them as links in workspaces. The role-based and task-based security in Finance and Operations apps can be used to help secure these menu items.
- **Filter reports based on application context.** You can build navigation experiences by passing one or more filters to Power BI reports. For example, depending on a user's actions or context, you can filter the Power BI report to reflect data from one business unit or a specific product. The user doesn't have to filter the



data. You can define drill-through links to Finance and Operations pages, so that users can go directly to the transactional details pages.

For more information about the Power BI Embedded service, see the [Power BI FAQ](#).

## Service availability

The Power BI Embedded service is automatically deployed and configured for all cloud-hosted, multi-box deployments. Because the service relies on Microsoft Azure services, application analytical workspaces and reports are unavailable in one-box environments. The Power BI Embedded service is already available in most Azure datacenters. You can check the latest availability on the [Azure status](#) page.

## Frequently asked questions

### **Can I customize the Power BI embedded reports?**

Yes. To customize the Power BI embedded reports, just install Power BI Desktop in a one-box environment, and follow the steps in [Create analytical reports by using Power BI Desktop](#).

### **Do customers have to purchase a separate Power BI license to use the new embedded analytics?**

No, customers don't have to purchase a separate Power BI license to use the new embedded analytics. However, a Power BI Pro license is required in order to connect to Entity Store from PowerBI.com by using DirectQuery.

### **Can I do data mash-ups by using external data in the embedded reports?**

No, you can't currently do data mash-ups by using external data in the embedded reports.

### **Can I help secure data to only those companies that I have access to?**

Yes, the single company view prevents users from accessing data from companies that they don't have access to. For more information about how to help secure custom solutions, see [Help secure analytical workspaces and reports by using Power BI Embedded](#).

### **How is currency shown across multiple companies?**

Currency is shown as a system currency. The system currency is defined on the [System parameters](#) page.

### **Can I drill from summary balances back into Finance and Operations?**

Yes, you can drill into the details on a Power BI report. However, there is limited support for drill-down into Finance and Operations apps.

### **What languages are currently supported?**

Currently, only English is supported. However, the Power BI team plans to add support for other languages.

### **Can I access analytical workspaces and reports in the on-premises version of Finance and Operations?**

No, you can't currently access analytical workspaces and reports in Dynamics 365 Finance + Operations (on-premises). Systems of Intelligence functions rely on cloud-hosted solutions.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customize embedded reports in analytical workspaces

2/18/2021 • 4 minutes to read • [Edit Online](#)

## Analytical workspaces

Analytical workspaces are bundled with the application suite. Through reporting, they offer users insights into data that is based on standard business operations. The reports are generic reports that are defined by business professionals. They include metrics that are considered interesting to a wide range of users from any industry.

However, in some cases, the standard reports include data that isn't relevant to all customers. More often, customers might want to access data points or calculations that are left out of the standard reports.

Power users can use web-friendly design tools to customize the analytical reports that are embedded in the application. By using the free-form canvas designer, users who are familiar with the relevant business insights that are required can help make the organization successful.

### IMPORTANT

Customizations that are made to the embedded analytical reports are automatically deployed by the service and made available to other users of the system.

### Edit embedded reports in analytical workspaces

The [How to edit an embedded report in an analytical workspace](#) video (shown above) is included in the [Playlist](#) available on YouTube.

### Important points about embedded analytical reports

Although the standard reports deliver insights that are tailored to a given business persona, customizations can often maximize the value of these standard reports.

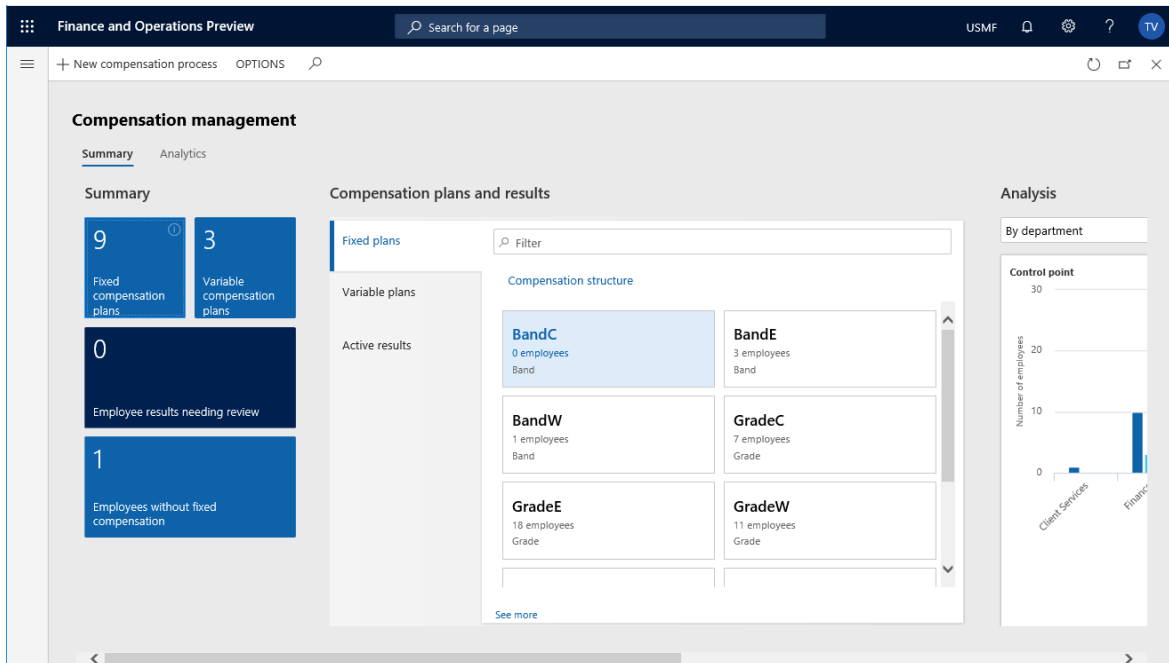
Here are some important points to note about this service capability:

- Customizations are limited to the report design canvas. Users can't change the definitions of report data sets.
- Report customizations that are made to the analytical workspace apply to all users in the environment.
- The service automatically preserves report customizations during product upgrades.
- The service doesn't support the export of customizations that are made to analytical workspaces.

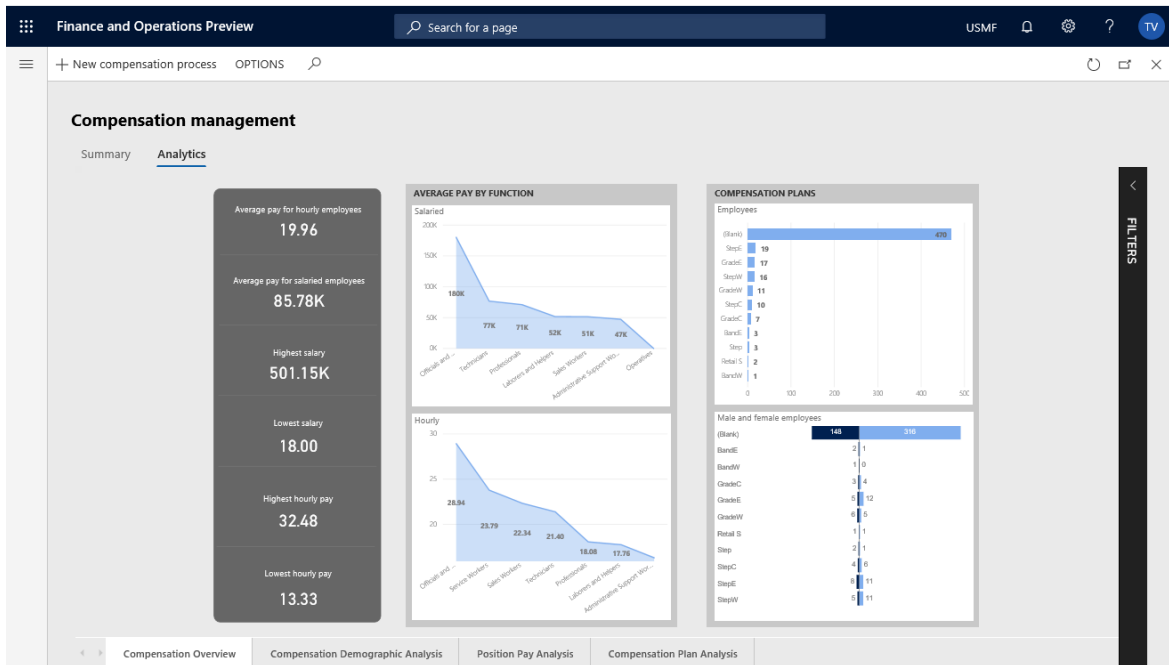
## Customize an analytical workspace

To customize the embedded application solutions, a user must be a member of the System Report Editors security group. Members of this security group can do customizations by using the buttons on the **Options** tab on the Action Pane of the application workspaces. This example shows how to customize one of the standard analytical workspaces that are bundled with the application suite.

1. Sign in and open the application workspace that you want to customize. In this example, you will replace the standard analytical report that is embedded in the **Compensation management** workspace.



2. Select the **Analytics** tab to access the workspace's embedded analytical report.

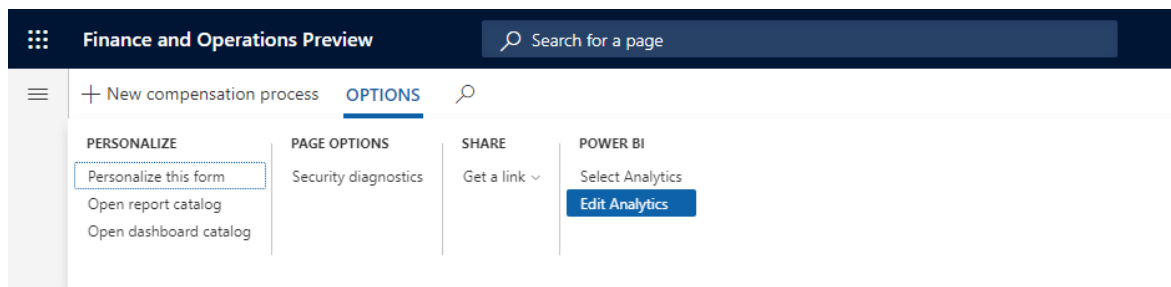


By default, you see the standard analytical workspace solution that is packaged with your application. The reports in this solution are automatically deployed and configured for your environment during the provisioning process.

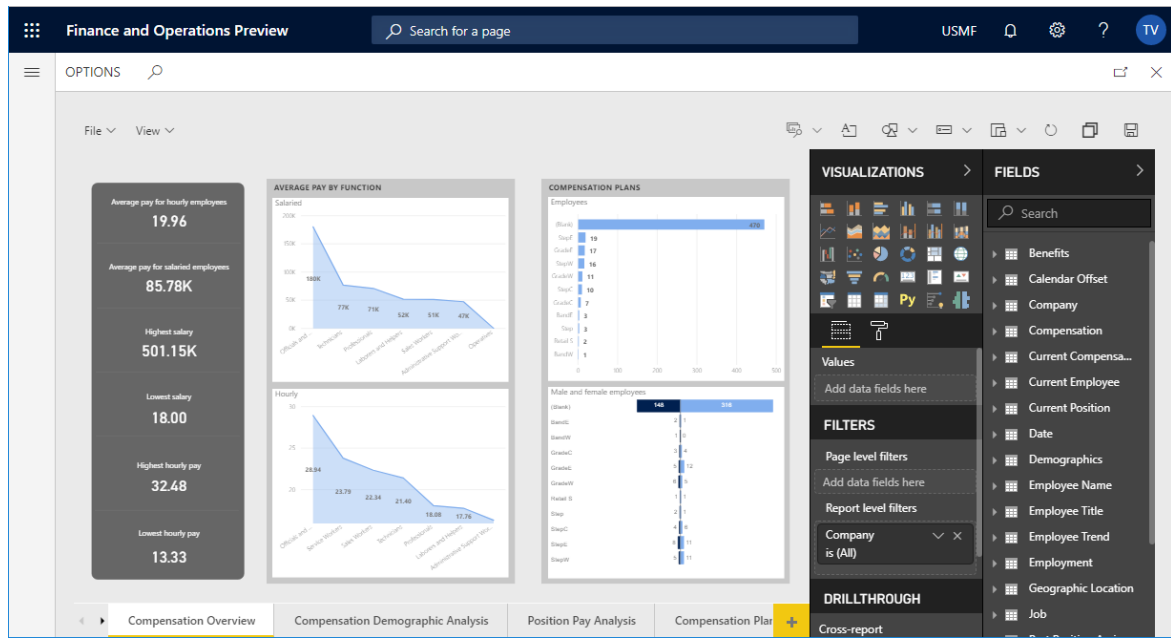
**NOTE**

The analytical workspaces require a hosted Microsoft Power BI service that is available only for dedicated environments. For more information, see [Accessing Analytical Workspaces and Reports on 1Box environment](#).

3. On the Action Pane, on the **Options** tab, in the **Power BI** group, select **Edit Analytics**.



The analytical workspace is opened in edit mode, and you have direct access to the Power BI web designer tools.



4. Use the Power BI web designer tools to customize the report canvas. The intuitive web controls let you perform typical actions such as adding and removing visuals, changing visual types, and formatting the content. You can also inspect the source of the report visualizations to make sure that decisions are based on the most relevant data that is available in the system. For more information, see [Add visualizations to a Power BI report](#).
5. After you've completed your report customizations, select the **Save** button to promote the report edits. Customizations to the report are reflected immediately in the service. Therefore, users in your organization have access to the latest innovations.

#### NOTE

Customizations made in the web editor are not saved back to the underlying PBIX report and are not retrievable if the PBIX report is later customized on a development environment.

## Restore the standard application solution

Follow these steps to restore the analytical workspaces that are bundled with the application solution.

1. In the analytical workspace, on the Action Pane, on the **Options** tab, in the **Power BI** group, select **Restore Analytics**.
2. To view the updates to the workspace, reload the page. Either move away from the workspace and then return, or refresh your browser.
3. In the **Compensation management** workspace, select the **Analytics** tab to access the original analytical workspace that was packaged with the application.

# Troubleshooting

Follow these steps to address common issues encountered while attempting to use analytical workspaces.

## Error message: *Please log into Power BI to access its resource*

The Power BI service requires explicit permission from the user to allow access to hosted content. Use the following steps to ensure the current user is able to connect to reports hosted on PowerBI.com from the application suite.

1. Open any application workspace containing a section titled **Link**. For example, "Bank management".
2. Select **Options**, and then select **Open report catalog** on the top left.
3. Follow the steps in the dialog box to **Authorize to Power BI** to access Finance and Operations apps for the current user.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Document Reporting Services

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article describes the integrated reporting solution that are available. This solution simplifies service administration, increases developer productivity, and provides an enhanced report viewing experience for users.

## Document Reporting Services

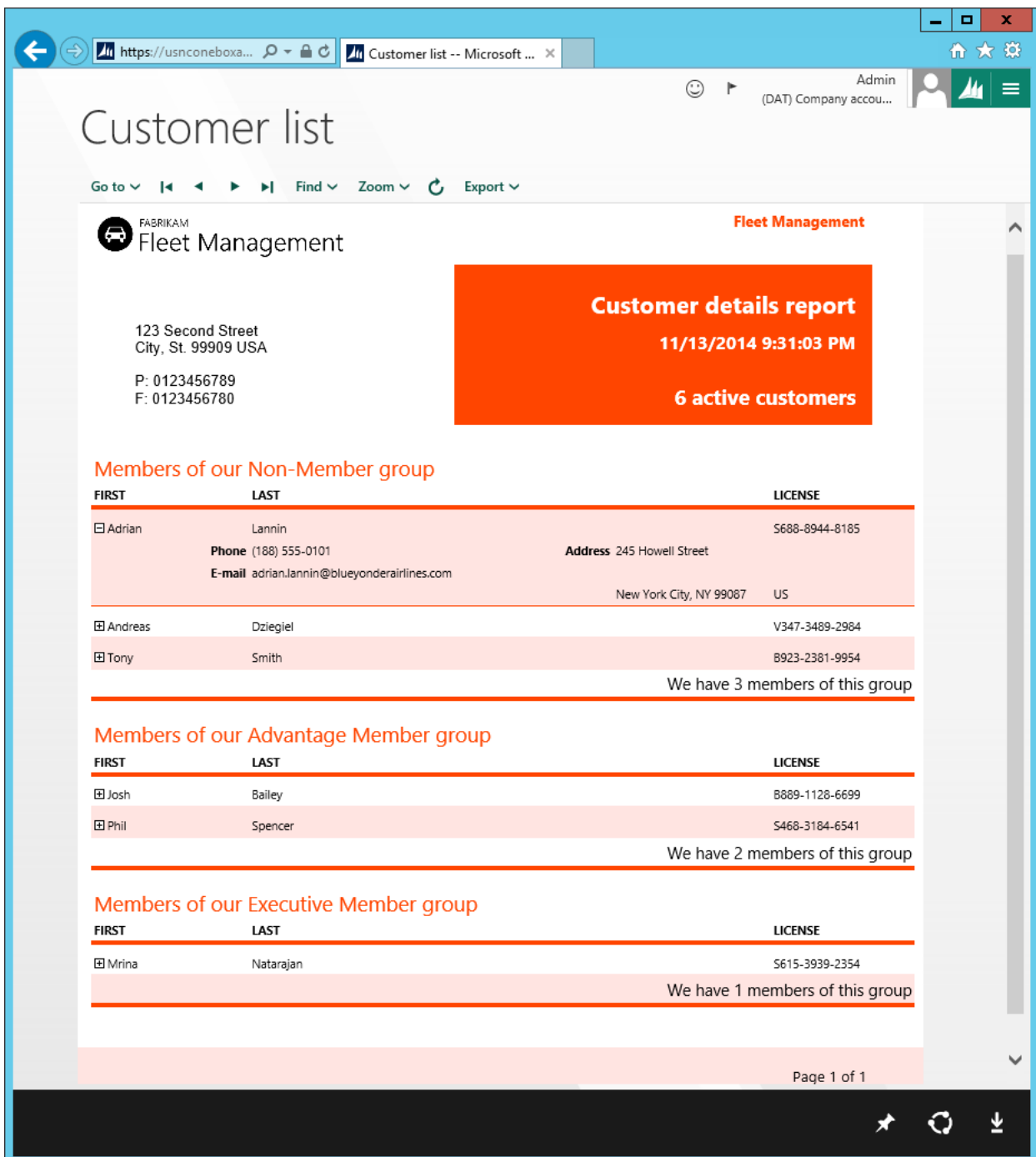
Document Reporting Services are based on Microsoft SQL Server Reporting Services (SSRS). In the current version of the application, these services are hosted in the Microsoft Azure compute service. If you're developing in a one-box environment, the services also run locally in the Azure compute emulator.

### **Service deployment – Local vs. cloud**

In a one-box environment, developers can create, modify, and preview reports, from end to end, by using Microsoft Visual Studio. A separate process isn't required in order to add reports to the application's metadata store. Changes to reports are packaged together with other solution updates and then deployed to the cloud after development is completed in the local environment.

### **Viewing reports**

The enhanced report viewing experience that provides for end users is the same as the report preview experience in Microsoft Visual Studio. You no longer use a separate design preview in Visual Studio. Instead, just press Ctrl+F5 to build and preview the report in an Internet Explorer window. The report appears exactly as it would appear in the client. Even the user's parameter experience is the same. The following screen shot shows an example of a report preview that is opened from Visual Studio.



## Service administration prerequisites

The following table compares the service administration prerequisites for Microsoft Dynamics AX 2012 and the current version of the application.

AX 2012	THE CURRENT VERSION OF THE APPLICATION
<p>A report development environment has the following prerequisites:</p> <ul style="list-style-type: none"> <li>• SSRS must be installed.</li> <li>• SSRS must be configured by using Reporting Services Configuration Manager.</li> <li>• SSRS extensions for the application must be installed.</li> </ul>	<p>Reporting services run in the Azure compute emulator, together with the application server. Therefore, there are no SSRS service administration prerequisites. After reports have been deployed to the local reporting services, they can be accessed from the client.</p>

## Developing application reports

The process for developing a report in the current version is easier than it is in AX 2012, because you can create and validate a reporting solution entirely in Visual Studio. The following table describes how the application simplifies the basic procedure for adding an automatic design report that is based on a query.

AX 2012	THE CURRENT VERSION OF THE APPLICATION
<ol style="list-style-type: none"> <li>1. In the application, create a query in the Application Object Tree (AOT).</li> <li>2. In Visual Studio, create a reporting project, and add the query to it.</li> <li>3. Edit the report in the Visual Studio model editor.</li> <li>4. Preview the report design in Visual Studio by using the model editor toolbar.</li> <li>5. Use Visual Studio to add the report to the AOT.</li> <li>6. Use the AOT in the client to create a menu item for the report, and add the menu item to a menu.</li> <li>7. Use the AOT to deploy the report to the report server.</li> <li>8. Verify the report in the client.</li> </ol>	<ol style="list-style-type: none"> <li>1. In Visual Studio, create a reporting project and the query.</li> <li>2. Edit the report in Visual Studio.</li> <li>3. In Visual Studio, add the report to a menu item, and set the menu item as a startup object.</li> <li>4. Use the AOT to deploy the report to the report server.</li> <li>5. Press Ctrl+F5 to verify the report in the application.</li> </ol> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p>[!NOTE] There is no longer a separate preview of the report design from the model editor.</p> </div> <ol style="list-style-type: none"> <li>6. When the whole solution is completed, deploy it to the cloud in one package.</li> </ol>

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Supported fonts

2/18/2021 • 13 minutes to read • [Edit Online](#)

Finance and Operations apps include access to hundreds of standard, business-ready fonts available for documents rendered by the cloud-hosted service.

This portfolio will continue to grow as the service expands into new regions and industries. However, the service no longer supports the installation of custom fonts in customer environments. Requests to expand the collection of fonts supported by the service will be considered on a case-by-case basis.

The following list captures the portfolio of font families available for documents produced using SQL Server Reporting Services (SSRS) services in Finance and Operations apps.

## Supported fonts for application version 8.3 with Platform update 32 or later

FONT FAMILY	SOURCE
Agency FB	Office font
Algerian	Office font
Arial	Finance and Operations font
Arial Black	Finance and Operations font
Arial Narrow	Finance and Operations font
Arial Rounded MT Bold	Office font
Baskerville Old Face	Office font
Bauhaus 93	Office font
BC C128 HD Medium	Finance and Operations font
BC C128 HD Narrow	Finance and Operations font
BC C128 HD Wide	Finance and Operations font
BC C128 Medium	Finance and Operations font
BC C128 Narrow	Finance and Operations font
BC C128 Wide	Finance and Operations font
BC C39 2 to 1 HD Medium	Finance and Operations font
BC C39 2 to 1 HD Narrow	Finance and Operations font

FONT FAMILY	SOURCE
BC C39 2 to 1 HD Wide	Finance and Operations font
BC C39 2 to 1 Medium	Finance and Operations font
BC C39 2 to 1 Narrow	Finance and Operations font
BC C39 2 to 1 Wide	Finance and Operations font
BC C39 3 to 1 HD Medium	Finance and Operations font
BC C39 3 to 1 HD Narrow	Finance and Operations font
BC C39 3 to 1 HD Wide	Finance and Operations font
BC C39 3 to 1 Medium	Finance and Operations font
BC C39 3 to 1 Narrow	Finance and Operations font
BC C39 3 to 1 Wide	Finance and Operations font
BC CBar 2 to 1 HD Medium	Finance and Operations font
BC CBar 2 to 1 HD Narrow	Finance and Operations font
BC CBar 2 to 1 HD Wide	Finance and Operations font
BC CBar 2 to 1 Medium	Finance and Operations font
BC CBar 2 to 1 Narrow	Finance and Operations font
BC CBar 2 to 1 Wide	Finance and Operations font
BC CBar 3 to 1 HD Medium	Finance and Operations font
BC CBar 3 to 1 HD Narrow	Finance and Operations font
BC CBar 3 to 1 HD Wide	Finance and Operations font
BC CBar 3 to 1 Medium	Finance and Operations font
BC CBar 3 to 1 Narrow	Finance and Operations font
BC CBar 3 to 1 Wide	Finance and Operations font
BC I25 HD Medium	Finance and Operations font
BC I25 HD Narrow	Finance and Operations font
BC I25 HD Wide	Finance and Operations font

FONT FAMILY	SOURCE
BC I25 Medium	Finance and Operations font
BC I25 Narrow	Finance and Operations font
BC I25 Wide	Finance and Operations font
BC Postnet	Finance and Operations font
BC UPC HD Medium	Finance and Operations font
BC UPC HD Narrow	Finance and Operations font
BC UPC HD Wide	Finance and Operations font
BC UPC Medium	Finance and Operations font
BC UPC Narrow	Finance and Operations font
BC UPC Wide	Finance and Operations font
Bell MT	Office font
Berlin Sans FB	Office font
Berlin Sans FB Demi	Office font
Bernard MT Condensed	Office font
Blackadder ITC	Office font
Bodoni MT	Office font
Bodoni MT Black	Office font
Bodoni MT Condensed	Office font
Bodoni MT Poster Compres	Office font
Book Antiqua	Office font
Bookman Old Style	Office font
Bookshelf Symbol 7	Office font
Bradley Hand ITC	Office font
Britannic Bold	Office font
Broadway	Office font

FONT FAMILY	SOURCE
Brush Script MT	Office font
Buxton Sketch	Office font
Calibri	Finance and Operations font
Calibri Light	Finance and Operations font
Californian FB	Office font
Calisto MT	Office font
Cambria	Finance and Operations font
Cambria Math	Finance and Operations font
Candara	Finance and Operations font
Castellar	Office font
Centaur	Office font
Century	Office font
Century Gothic	Office font
Century Schoolbook	Office font
Chiller	Office font
Colonna MT	Office font
Comic Sans MS	Finance and Operations font
Consolas	Finance and Operations font
Constantia	Finance and Operations font
Cooper Black	Office font
Copperplate Gothic Bold	Office font
Copperplate Gothic Light	Office font
Corbel	Finance and Operations font
Courier New	Finance and Operations font
Curlz MT	Office font

FONT FAMILY	SOURCE
DengXian	Office font
Dotum	Finance and Operations font
DotumChe	Finance and Operations font
Ebrima	Finance and Operations font
Edwardian Script ITC	Office font
Elephant	Office font
Engravers MT	Office font
Eras Bold ITC	Office font
Eras Demi ITC	Office font
Eras Light ITC	Office font
Eras Medium ITC	Office font
Felix Titling	Office font
Footlight MT Light	Office font
Forte	Office font
Franklin Gothic Book	Office font
Franklin Gothic Demi	Office font
Franklin Gothic Demi Con	Office font
Franklin Gothic Heavy	Office font
Franklin Gothic Medium	Finance and Operations font
Franklin Gothic Medium C	Office font
Freestyle Script	Office font
French Script MT	Office font
Gabriola	Finance and Operations font
Gadugi	Finance and Operations font
Garamond	Office font

FONT FAMILY	SOURCE
Georgia	Finance and Operations font
Gigi	Office font
Gill Sans MT	Office font
Gill Sans MT Condensed	Office font
Gill Sans MT Ext Condens	Office font
Gill Sans Ultra Bold	Office font
Gill Sans Ultra Bold Con	Office font
Gloucester MT Extra Cond	Office font
Goudy Old Style	Office font
Goudy Stout	Office font
Gulim	Finance and Operations font
GulimChe	Finance and Operations font
Haettenschweiler	Office font
Harlow Solid Italic	Office font
Harrington	Office font
High Tower Text	Office font
IDAutomationMICR	Finance and Operations font
Impact	Finance and Operations font
Imprint MT Shadow	Office font
Informal Roman	Office font
Javanese Text	Finance and Operations font
Jokerman	Office font
Juice ITC	Office font
Kristen ITC	Office font
Kunstler Script	Office font

FONT FAMILY	SOURCE
Leelawadee	Office font
Leelawadee UI	Finance and Operations font
Leelawadee UI Semilight	Finance and Operations font
Lucida Bright	Office font
Lucida Calligraphy	Office font
Lucida Console	Finance and Operations font
Lucida Fax	Office font
Lucida Handwriting	Office font
Lucida Sans	Office font
Lucida Sans Typewriter	Office font
Lucida Sans Unicode	Finance and Operations font
Magneto	Office font
Maiandra GD	Office font
Malgun Gothic	Finance and Operations font
Malgun Gothic Semilight	Finance and Operations font
Marlett	Finance and Operations font
Matura MT Script Capital	Office font
MICR E13B 2.1	Finance and Operations font
Microsoft Himalaya	Finance and Operations font
Microsoft JhengHei	Finance and Operations font
Microsoft JhengHei Light	Finance and Operations font
Microsoft JhengHei UI	Finance and Operations font
Microsoft JhengHei UI Li	Office font
Microsoft MHei	Office font
Microsoft NeoGothic	Office font

FONT FAMILY	SOURCE
Microsoft New Tai Lue	Finance and Operations font
Microsoft PhagsPa	Finance and Operations font
Microsoft Sans Serif	Finance and Operations font
Microsoft Tai Le	Finance and Operations font
Microsoft Uighur	Office font
Microsoft YaHei	Finance and Operations font
Microsoft YaHei Light	Finance and Operations font
Microsoft YaHei UI	Finance and Operations font
Microsoft YaHei UI Light	Finance and Operations font
Microsoft Yi Baiti	Finance and Operations font
MingLiU	Finance and Operations font
MingLiU-ExtB	Finance and Operations font
MingLiU_HKSCS	Finance and Operations font
MingLiU_HKSCS-ExtB	Finance and Operations font
Mistral	Office font
Modern No. 20	Office font
Mongolian Baiti	Finance and Operations font
Monotype Corsiva	Office font
MS Gothic	Finance and Operations font
MS Outlook	Office font
MS PGothic	Finance and Operations font
MS Reference Sans Serif	Office font
MS Reference Specialty	Office font
MS UI Gothic	Finance and Operations font
MT Extra	Office font



FONT FAMILY	SOURCE
MV Boli	Finance and Operations font
Myanmar Text	Finance and Operations font
Niagara Engraved	Office font
Niagara Solid	Office font
Nirmala UI	Finance and Operations font
Nirmala UI Semilight	Finance and Operations font
NSimSun	Finance and Operations font
OCR A Extended	Office font
OCRB	Finance and Operations font
Old English Text MT	Office font
Onyx	Office font
Palace Script MT	Office font
Palatino Linotype	Finance and Operations font
Papyrus	Office font
Parchment	Office font
Perpetua	Office font
Perpetua Titling MT	Office font
Playbill	Office font
PMingLiU	Finance and Operations font
PMingLiU-ExtB	Finance and Operations font
Poor Richard	Office font
Pristina	Office font
Rage Italic	Office font
Ravie	Office font
Rockwell	Office font

FONT FAMILY	SOURCE
Rockwell Condensed	Office font
Rockwell Extra Bold	Office font
Script MT Bold	Office font
Segoe Marker	Office font
Segoe MDL2 Assets	Finance and Operations font
Segoe Print	Finance and Operations font
Segoe Script	Finance and Operations font
Segoe UI	Finance and Operations font
Segoe UI Black	Finance and Operations font
Segoe UI Emoji	Finance and Operations font
Segoe UI Historic	Finance and Operations font
Segoe UI Light	Finance and Operations font
Segoe UI Semibold	Finance and Operations font
Segoe UI Semilight	Finance and Operations font
Segoe UI Symbol	Finance and Operations font
Segoe WP	Office font
Segoe WP Black	Office font
Segoe WP Light	Office font
Segoe WP Semibold	Office font
Segoe WP SemiLight	Office font
Showcard Gothic	Office font
SimSun	Finance and Operations font
SimSun-ExtB	Finance and Operations font
Sitka Banner	Finance and Operations font
Sitka Display	Finance and Operations font

FONT FAMILY	SOURCE
Sitka Heading	Finance and Operations font
Sitka Small	Finance and Operations font
Sitka Subheading	Finance and Operations font
Sitka Text	Finance and Operations font
SketchFlow Print	Office font
Snap ITC	Office font
Stencil	Office font
Sylfaen	Finance and Operations font
Symbol	Finance and Operations font
Tahoma	Finance and Operations font
Tempus Sans ITC	Office font
Times New Roman	Finance and Operations font
Trebuchet MS	Finance and Operations font
Tw Cen MT	Office font
Tw Cen MT Condensed	Office font
Tw Cen MT Condensed Extr	Office font
Verdana	Finance and Operations font
Viner Hand ITC	Office font
Vivaldi	Office font
Vladimir Script	Office font
Webdings	Finance and Operations font
Wide Latin	Office font
Wingdings	Finance and Operations font
Wingdings 2	Office font
Wingdings 3	Office font

FONT FAMILY	SOURCE
Yu Gothic	Finance and Operations font
Yu Gothic Light	Finance and Operations font
Yu Gothic Medium	Finance and Operations font
Yu Gothic UI	Finance and Operations font
Yu Gothic UI Light	Finance and Operations font
Yu Gothic UI Semibold	Finance and Operations font
Yu Gothic UI Semilight	Finance and Operations font

## Supported fonts for application version 8.0 with Platform update 23

NUMBER	FONT FAMILY
1	Agency FB
2	Algerian
3	Arial
4	Arial Black
5	Arial Narrow
6	Arial Rounded MT Bold
7	Baskerville Old Face
8	Bauhaus 93
9	BC C128 HD Medium
10	BC C128 HD Narrow
11	BC C128 HD Wide
12	BC C128 Medium
13	BC C128 Narrow
14	BC C128 Wide
15	BC C39 2 to 1 HD Medium
16	BC C39 2 to 1 HD Narrow

NUMBER	FONT FAMILY
17	BC C39 2 to 1 HD Wide
18	BC C39 2 to 1 Medium
19	BC C39 2 to 1 Narrow
20	BC C39 2 to 1 Wide
21	BC C39 3 to 1 HD Medium
22	BC C39 3 to 1 HD Narrow
23	BC C39 3 to 1 HD Wide
24	BC C39 3 to 1 Medium
25	BC C39 3 to 1 Narrow
26	BC C39 3 to 1 Wide
27	BC CBar 2 to 1 HD Medium
28	BC CBar 2 to 1 HD Narrow
29	BC CBar 2 to 1 HD Wide
30	BC CBar 2 to 1 Medium
31	BC CBar 2 to 1 Narrow
32	BC CBar 2 to 1 Wide
33	BC CBar 3 to 1 HD Medium
34	BC CBar 3 to 1 HD Narrow
35	BC CBar 3 to 1 HD Wide
36	BC CBar 3 to 1 Medium
37	BC CBar 3 to 1 Narrow
38	BC CBar 3 to 1 Wide
39	BC I25 HD Medium
40	BC I25 HD Narrow
41	BC I25 HD Wide

NUMBER	FONT FAMILY
42	BC I25 Medium
43	BC I25 Narrow
44	BC I25 Wide
45	BC Postnet
46	BC UPC HD Medium
47	BC UPC HD Narrow
48	BC UPC HD Wide
49	BC UPC Medium
50	BC UPC Narrow
51	BC UPC Wide
52	Bell MT
53	Berlin Sans FB
54	Berlin Sans FB Demi
55	Bernard MT Condensed
56	Blackadder ITC
57	Bodoni MT
58	Bodoni MT Black
59	Bodoni MT Condensed
60	Bodoni MT Poster Compres
61	Book Antiqua
62	Bookman Old Style
63	Bookshelf Symbol 7
64	Bradley Hand ITC
65	Britannic Bold
66	Broadway

NUMBER	FONT FAMILY
67	Brush Script MT
68	Buxton Sketch
69	Calibri
70	Calibri Light
71	Californian FB
72	Calisto MT
73	Cambria
74	Cambria Math
75	Candara
76	Castellar
77	Centaur
78	Century
79	Century Gothic
80	Century Schoolbook
81	Chiller
82	Colonna MT
83	Comic Sans MS
84	Consolas
85	Constantia
86	Cooper Black
87	Copperplate Gothic Bold
88	Copperplate Gothic Light
89	Corbel
90	Courier New
91	Curlz MT

NUMBER	FONT FAMILY
92	DengXian
93	Dotum
94	DotumChe
95	Ebrima
96	Edwardian Script ITC
97	Elephant
98	Engravers MT
99	Eras Bold ITC
100	Eras Demi ITC
101	Eras Light ITC
102	Eras Medium ITC
103	Felix Titling
104	Footlight MT Light
105	Forte
106	Franklin Gothic Book
107	Franklin Gothic Demi
108	Franklin Gothic Demi Con
109	Franklin Gothic Heavy
110	Franklin Gothic Medium
111	Franklin Gothic Medium C
112	Freestyle Script
113	French Script MT
114	Gabriola
115	Gadugi
116	Garamond



NUMBER	FONT FAMILY
117	Georgia
118	Gigi
119	Gill Sans MT
120	Gill Sans MT Condensed
121	Gill Sans MT Ext Condens
122	Gill Sans Ultra Bold
123	Gill Sans Ultra Bold Con
124	Gloucester MT Extra Cond
125	Goudy Old Style
126	Goudy Stout
127	Gulim
128	GulimChe
129	Haettenschweiler
130	Harlow Solid Italic
131	Harrington
132	High Tower Text
133	IDAutomationMICR
134	Impact
135	Imprint MT Shadow
136	Informal Roman
137	Javanese Text
138	Jokerman
139	Juice ITC
140	Kristen ITC
141	Kunstler Script

NUMBER	FONT FAMILY
142	Leelawadee
143	Leelawadee UI
144	Leelawadee UI Semilight
145	Lucida Bright
146	Lucida Calligraphy
147	Lucida Console
148	Lucida Fax
149	Lucida Handwriting
150	Lucida Sans
151	Lucida Sans Typewriter
152	Lucida Sans Unicode
153	Magneto
154	Maiandra GD
155	Malgun Gothic
156	Malgun Gothic Semilight
157	Marlett
158	Matura MT Script Capital
159	MICR E13B 2.1
160	Microsoft Himalaya
161	Microsoft JhengHei
162	Microsoft JhengHei Light
163	Microsoft JhengHei UI
164	Microsoft JhengHei UI Li
165	Microsoft MHei
166	Microsoft NeoGothic

NUMBER	FONT FAMILY
167	Microsoft New Tai Lue
168	Microsoft PhagsPa
169	Microsoft Sans Serif
170	Microsoft Tai Le
171	Microsoft Uighur
172	Microsoft YaHei
173	Microsoft YaHei Light
174	Microsoft YaHei UI
175	Microsoft YaHei UI Light
176	Microsoft Yi Baiti
177	MingLiU
178	MingLiU-ExtB
179	MingLiU_HKSCS
180	MingLiU_HKSCS-ExtB
181	Mistral
182	Modern No. 20
183	Mongolian Baiti
184	Monotype Corsiva
185	MS Gothic
186	MS Outlook
187	MS PGothic
188	MS Reference Sans Serif
189	MS Reference Specialty
190	MS UI Gothic
191	MT Extra

NUMBER	FONT FAMILY
192	MV Boli
193	Myanmar Text
194	Niagara Engraved
195	Niagara Solid
196	Nirmala UI
197	Nirmala UI Semilight
198	NSimSun
199	OCR A Extended
200	OCRB
201	Old English Text MT
202	Onyx
203	Palace Script MT
204	Palatino Linotype
205	Papyrus
206	Parchment
207	Perpetua
208	Perpetua Titling MT
209	Playbill
210	PMingLiU
211	PMingLiU-ExtB
212	Poor Richard
213	Pristina
214	Rage Italic
215	Ravie
216	Rockwell

NUMBER	FONT FAMILY
217	Rockwell Condensed
218	Rockwell Extra Bold
219	Script MT Bold
220	Segoe Marker
221	Segoe MDL2 Assets
222	Segoe Print
223	Segoe Script
224	Segoe UI
225	Segoe UI Black
226	Segoe UI Emoji
227	Segoe UI Historic
228	Segoe UI Light
229	Segoe UI Semibold
230	Segoe UI Semilight
231	Segoe UI Symbol
232	Segoe WP
233	Segoe WP Black
234	Segoe WP Light
235	Segoe WP Semibold
236	Segoe WP SemiLight
237	Showcard Gothic
238	SimSun
239	SimSun-ExtB
240	Sitka Banner
241	Sitka Display

NUMBER	FONT FAMILY
242	Sitka Heading
243	Sitka Small
244	Sitka Subheading
245	Sitka Text
246	SketchFlow Print
247	Snap ITC
248	Stencil
249	Sylfaen
250	Symbol
251	Tahoma
252	Tempus Sans ITC
253	Times New Roman
254	Trebuchet MS
255	Tw Cen MT
256	Tw Cen MT Condensed
257	Tw Cen MT Condensed Extr
258	Verdana
259	Viner Hand ITC
260	Vivaldi
261	Vladimir Script
262	Webdings
263	Wide Latin
264	Wingdings
265	Wingdings 2
266	Wingdings 3

NUMBER	FONT FAMILY
267	Yu Gothic
268	Yu Gothic Light
269	Yu Gothic Medium
270	Yu Gothic UI
271	Yu Gothic UI Light
272	Yu Gothic UI Semibold
273	Yu Gothic UI Semilight

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

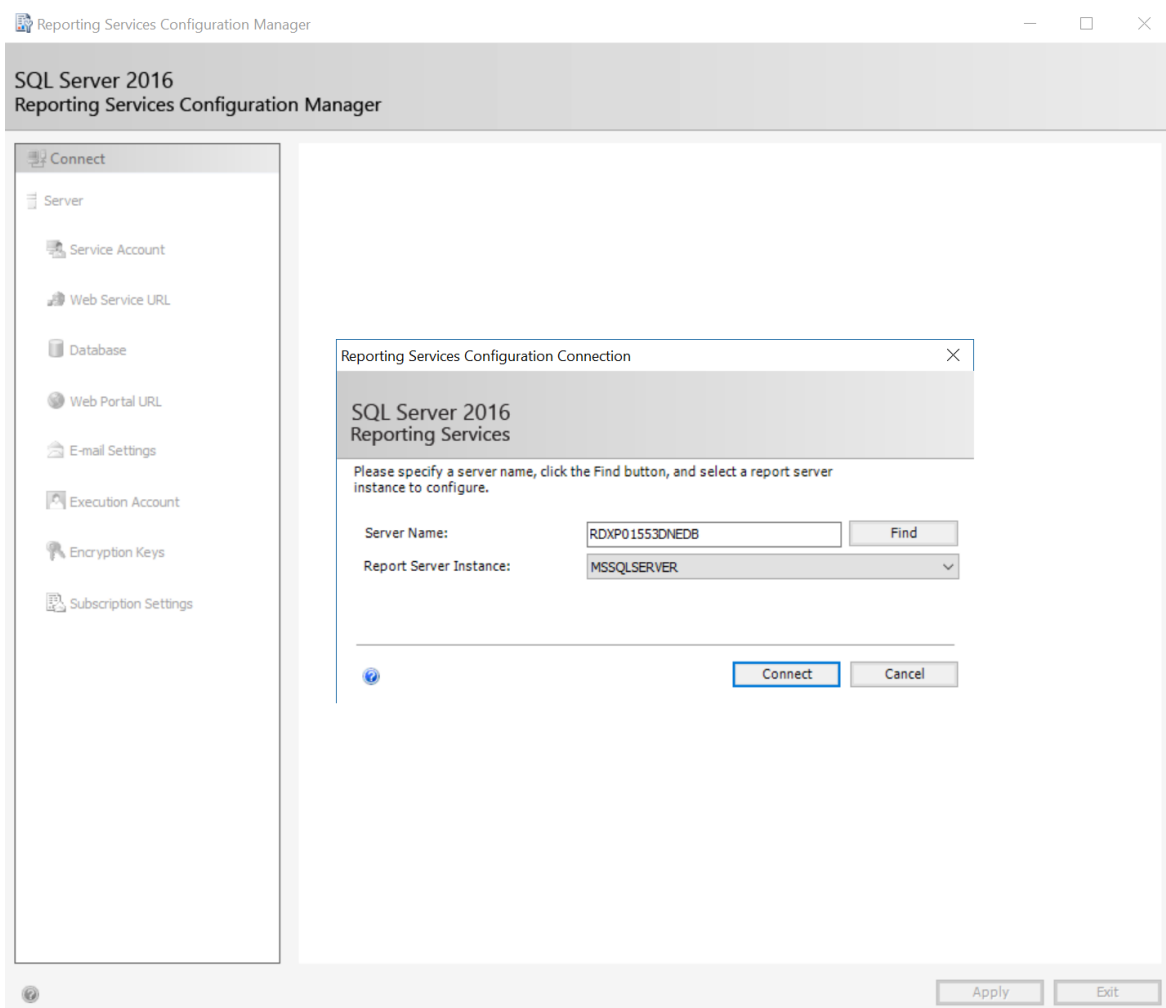
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure SQL Server Reporting Services for on-premises deployments

2/18/2021 • 2 minutes to read • [Edit Online](#)

Use the steps in this topic to configure SQL Server Reporting Services (SSRS) for your Microsoft Dynamics 365 Finance + Operations (on-premises) deployment.

1. Open the Reporting Services Configuration Manager application.
2. Leave the default **Server name**, which should be the name of the current machine, and the **Report Server Instance**, **MSSQLSERVER**.
3. Click **Connect**.



4. Click the **Service Account** tab and verify that the settings match the following graphic.

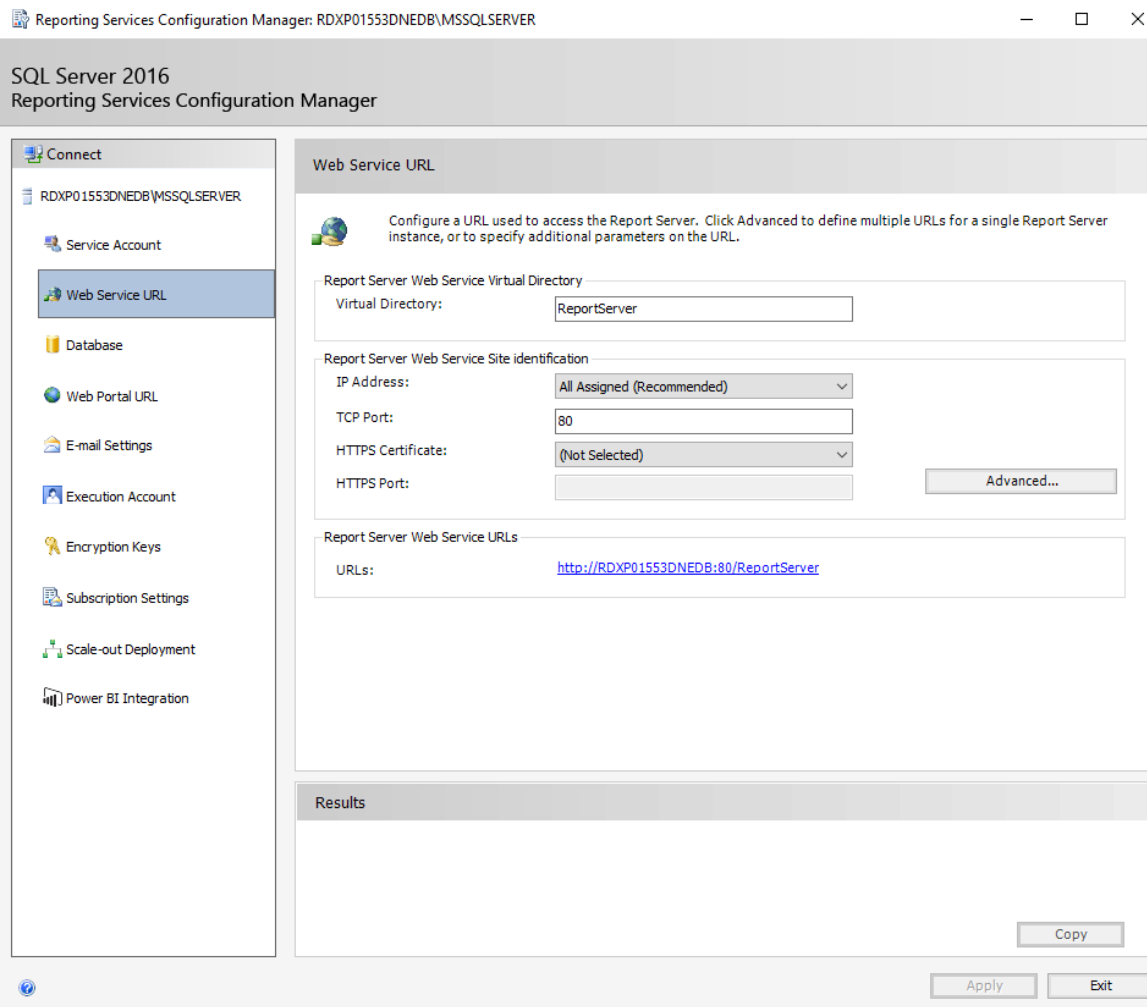


SQL Server 2016  
Reporting Services Configuration Manager

The screenshot shows the 'Service Account' configuration window in the Reporting Services Configuration Manager. The left-hand navigation pane lists various configuration options, with 'Service Account' currently selected. The main area is titled 'Service Account' and contains the following elements:

- A header section with a user icon and the text: "Specify a built-in account or Windows domain user account to run the report server service."
- A section titled "Report Server Service Account" with the instruction: "Choose an option to set the service account and then click Apply."
- Two radio button options:
  - Use built-in account: A dropdown menu is set to "Local System".
  - Use another account: Two text input fields are provided for "Account (Domain\user):" and "Password:".
- A "Results" section at the bottom, which is currently empty.
- A "Copy" button located at the bottom right of the main configuration area.
- At the very bottom of the window, there are "Apply" and "Exit" buttons.

5. Click the **Web Service URL** tab and verify that the settings match the following graphic.



6. Click the **Database** tab and verify that the **Database Name** and **Credential settings** match the following graphic.

**NOTE**

You will need to create a new database. To do this, click **Change Database**, and then verify that the new database name is: **DynamicsAxReportServer**.

SQL Server 2016  
Reporting Services Configuration Manager

**Connect**

- RDXP01553DNEDB\MSSQLSERVER
- Service Account
- Web Service URL
- Database**
- Web Portal URL
- E-mail Settings
- Execution Account
- Encryption Keys
- Subscription Settings
- Scale-out Deployment
- Power BI Integration

### Report Server Database

Reporting Services stores all report server content and application data in a database. Use this page to create or change the report server database or update database connection credentials.

**Current Report Server Database**

Click: Change database to select a different database or create a new database in native or SharePoint integrated mode.

SQL Server Name:	rdxp01553Dnedb
Database Name:	DynamicsAxReportServer
Report Server Mode:	Native

**Current Report Server Database Credential**

The following credentials are used by the report server to connect to the report server database. Use the options below to choose a different account or update a password.

Credential:	Service Account
Login:	LocalSystem
Password:	*****

### Results

7. Click the **Web Portal URL** tab and verify that the settings match the following graphic.

**NOTE**

You must click **Apply** to create and properly configure the Portal.

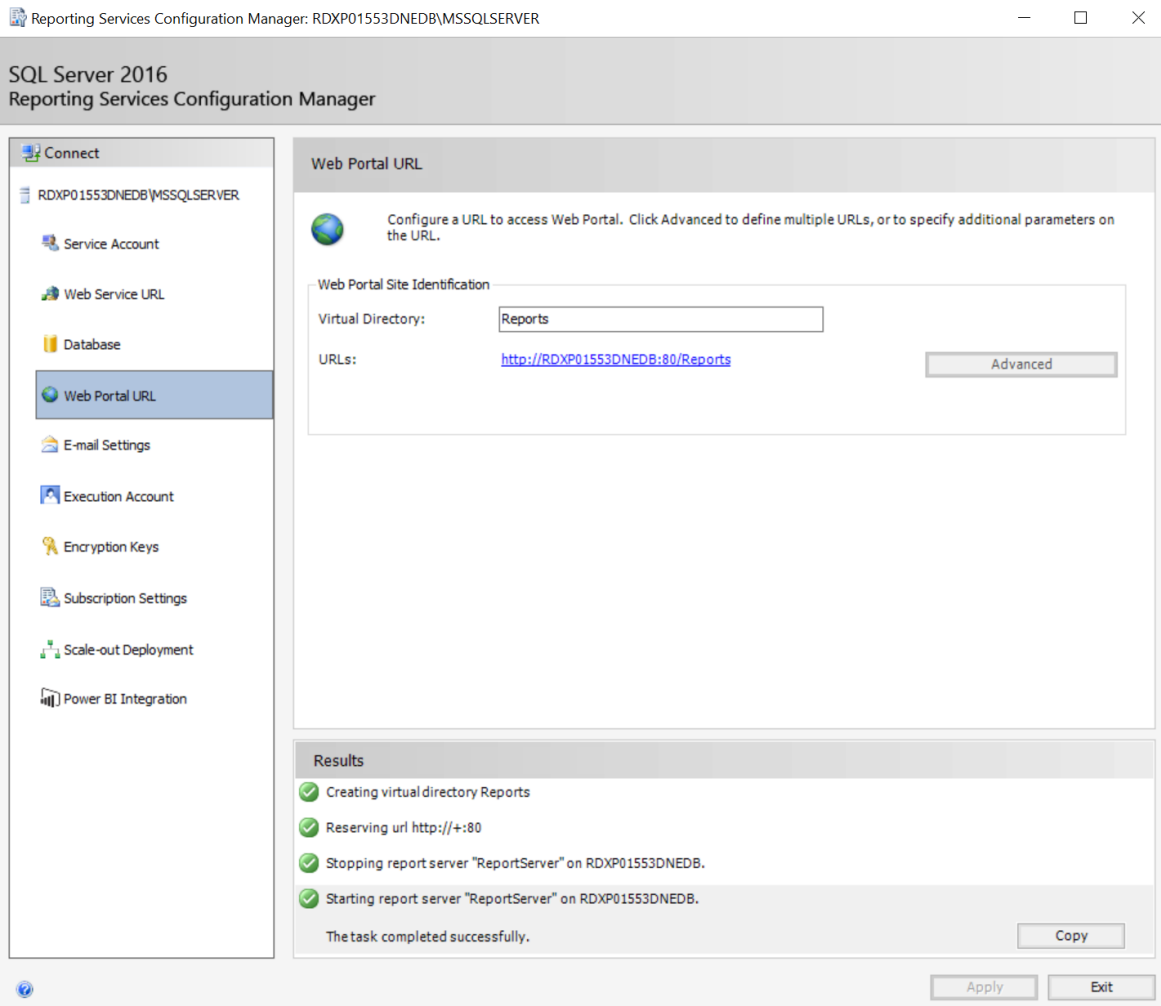
SQL Server 2016  
Reporting Services Configuration Manager

The screenshot shows the 'Web Portal URL' configuration window in the Reporting Services Configuration Manager. The left-hand navigation pane is open to the 'Web Portal URL' tab. The main area contains the following elements:

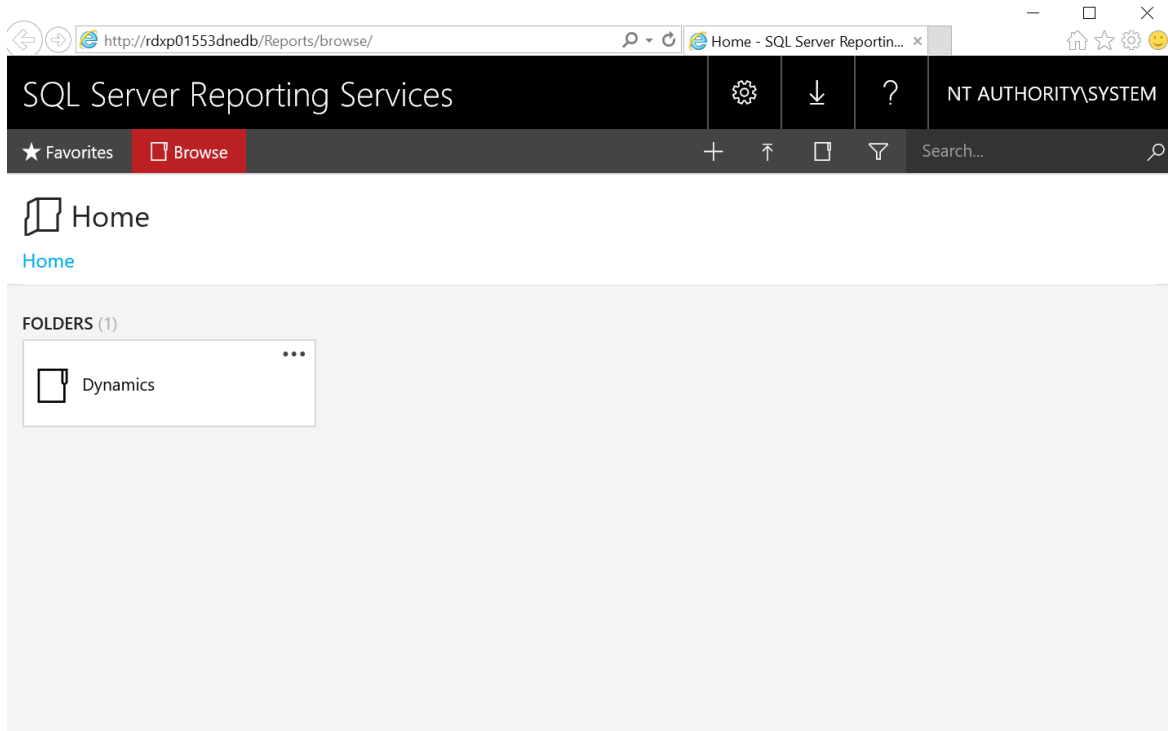
- Web Portal URL** header.
- Instructional text: "Configure a URL to access Web Portal. Click Advanced to define multiple URLs, or to specify additional parameters on the URL."
- Warning icon and text: "The Web Portal virtual directory name is not configured. To configure the directory, enter a name or use the default value that is provided, and then click Apply."
- Web Portal Site Identification** section with a scrollable area containing:
  - Virtual Directory:** A text box containing the value "Reports".
  - URLs:** A text box containing the value "http://RDXP01553DNEDB:80/Reports".
  - An **Advanced** button to the right of the URL text box.
- A horizontal scrollbar below the configuration area.
- Results** section, which is currently empty.
- A **Copy** button at the bottom right of the Results section.

At the bottom of the window, there are **Apply** and **Exit** buttons.

After the Portal is configured, the **Web Portal** tab will match the following graphic.



8. Click the reports URL to view the SQL Server Reporting Services web portal.
9. When you are in the portal, create a new folder named **Dynamics**.



10. In the **Reporting Services Configuration Manager**, click the **E-mail Settings** tab and verify that the settings match the following graphic.

SQL Server 2016  
Reporting Services Configuration Manager

The screenshot shows the 'E-mail Settings' configuration page in the Reporting Services Configuration Manager. The left-hand navigation pane lists various configuration options, with 'E-mail Settings' currently selected. The main content area is titled 'E-mail Settings' and contains a descriptive icon and text: 'To use report server e-mail, specify an existing SMTP server and an e-mail account that can send e-mail from that server.' Below this is a section for 'SMTP Settings' with a sub-instruction: 'To edit, change the fields and click the Apply button.' The settings are as follows:

Field	Value
Sender Address:	<input type="text"/>
Current SMTP Delivery Method:	Use SMTP server
SMTP Server:	<input type="text"/>
Authentication:	No authentication

At the bottom of the configuration area is a 'Copy' button. Below the configuration area is a 'Results' section, which is currently empty. At the very bottom of the window are 'Apply' and 'Exit' buttons.

11. Click the **Execution Account** tab and verify that the settings match the following graphic.

SQL Server 2016  
Reporting Services Configuration Manager

The screenshot shows the 'Execution Account' configuration window in the Reporting Services Configuration Manager. The left-hand navigation pane lists various configuration options, with 'Execution Account' currently selected. The main area contains instructions on how to specify an execution account, including a checkbox and three text input fields for account name, password, and confirmation. A 'Results' section is visible below, and 'Apply' and 'Exit' buttons are at the bottom right.

**Connect**

- Connect
- RDXP01553DNEDB\MSSQLSERVER
- Service Account
- Web Service URL
- Database
- Web Portal URL
- E-mail Settings
- Execution Account**
- Encryption Keys
- Subscription Settings
- Scale-out Deployment
- Power BI Integration

**Execution Account**

Specify this account to enable the use of report data sources that do not require credentials or to connect to remote servers that store external images used in reports. Be sure to specify a domain user account with minimal permissions for performing read-only operations. Avoid using an account that has more permissions than you actually need. The account you specify should be different from the service account to ensure you do not compromise security on your report server instance.

**Execution Account**

Use the following options to set the account, then click Apply.

Specify an execution account

Account:

Password:

Confirm Password:

**Results**

Copy

Apply Exit

Don't change the default settings on the **Encryption Keys** tab.

SQL Server 2016  
Reporting Services Configuration Manager

The screenshot shows the Reporting Services Configuration Manager interface. On the left is a navigation pane with the following items: Connect, RDXP01553DNEDB\MSSQLSERVER, Service Account, Web Service URL, Database, Web Portal URL, E-mail Settings, Execution Account, Encryption Keys (selected), Subscription Settings, Scale-out Deployment, and Power BI Integration. The main area is titled 'Encryption Keys' and contains the following text: 'Reporting Services uses a symmetric key to encrypt credentials, connection strings, and other sensitive data that is stored in the report server database. You can manage this key by creating a backup. If you migrate or move the report server installation to another computer, you can restore the key to regain access to encrypted content.' Below this text are four sections, each with a button: 'Backup' (Backup the key to a password protected file for report server recovery in case of emergency. Backup), 'Restore' (To restore the encryption key, click the Restore button. You must know the password that was used to protect the encryption key file. Restore), 'Change' (This operation replaces the encryption key with a newer version. Change), and 'Delete Encrypted Content' (All stored connection strings, credentials, and encrypted values in a subscription will be deleted. After you delete this content, you must redefine all data source connections and subscriptions used on the report server. Delete). At the bottom right of the main area is a 'Copy' button. At the very bottom of the window are 'Apply' and 'Exit' buttons.

12. Click the **Subscription Settings** tab, and verify that the settings match the following graphic.



SQL Server 2016  
Reporting Services Configuration Manager

**Connect**

- RDXP01553DNEDB\MSSQLSERVER
- Service Account
- Web Service URL
- Database
- Web Portal URL
- E-mail Settings
- Execution Account
- Encryption Keys
- Subscription Settings**
- Scale-out Deployment
- Power BI Integration

**File Share Account**

Configure an account to be used by subscriptions to access file shares. Use an account with as minimum permissions as possible and an account that is different from the account used for the Reporting Services service account.

**File Share Account**

Use the following options to set the account, then click Apply.

Specify a file share account

Account:

Password:

Confirm Password:

**Results**

Copy

Apply Exit

Don't change the default settings on the **Scale-out Deployment** tab.

SQL Server 2016  
Reporting Services Configuration Manager

The screenshot shows the 'Scale-out Deployment' page in the Reporting Services Configuration Manager. The left-hand navigation pane is visible, with 'Scale-out Deployment' selected. The main content area displays the 'Scale-out Deployment Status' section, which includes a table of server instances and two buttons: 'Add Server' and 'Remove Server'. Below this is a 'Results' section with a 'Copy' button. At the bottom of the window are 'Apply' and 'Exit' buttons.

**Scale-out Deployment**

Use this page to view information about a scale-out deployment. Report Servers that are joined to the scale-out can store encrypted data in a common Report Server database. Servers that are waiting to join the scale-out deployment must be added by a Report Server instance that is already part of the deployment.

**Scale-out Deployment Status**

SQL Server Name: rdxp01553Dnedb  
Database Name: DynamicsAxReportServer  
Report Server Mode: Native

Server	Instance	Status
RDXP01553DNEDB	MSSQLSERVER	Joined

**Results**

Don't change the default settings on the **Power BI Integration** tab.

## SQL Server 2016 Reporting Services Configuration Manager

The screenshot shows the Reporting Services Configuration Manager interface. On the left is a 'Connect' sidebar with various configuration options. The main area is titled 'Power BI Integration' and contains instructions for registering the report server with Power BI. A 'Register with Power BI' button is visible, and the status is 'Not Registered'. At the bottom right, there are 'Apply' and 'Exit' buttons.

**Connect**

- RDXP01553DNEDB\MSSQLSERVER
- Service Account
- Web Service URL
- Database
- Web Portal URL
- E-mail Settings
- Execution Account
- Encryption Keys
- Subscription Settings
- Scale-out Deployment
- Power BI Integration**

**Power BI Integration**

When the report server is registered with Power BI, users can pin report items to their Power BI dashboards. Use this page to register or unregister the report server with Power BI.

**Power BI Registration**

Click Register with Power BI to register the Report server with Power BI.

Power BI Registration: Not Registered

**Results**

13. Click Exit to close the Reporting Services Configuration Manager.

## SQL Server 2016 Reporting Services Configuration Manager

The screenshot displays the Reporting Services Configuration Manager interface. On the left is a navigation pane with the following options: Connect, RDXP01553DNEDB\MSSQLSERVER (selected), Service Account, Web Service URL, Database, Web Portal URL, E-mail Settings, Execution Account, Encryption Keys, Subscription Settings, Scale-out Deployment, and Power BI Integration. The main area is titled 'Report Server Status' and contains a warning icon and text: 'Use the Reporting Services Configuration Manager tool to define or modify settings for the Report Server and Report Manager. If you installed Reporting Services in files-only mode, you must configure the Web service URL, the database, and the Report Manager URL.' Below this is a section for 'Current Report Server' with the following details:

SQL Server Instance:	MSSQLSERVER
Instance ID:	MSRS13.MSSQLSERVER
Edition:	ENTERPRISE EDITION
Product Version:	13.0.4422.0
Report Server Database Name:	DynamicsAxReportServer
Report Server Mode:	Native
Report Service Status:	Started

At the bottom of this section are 'Start' and 'Stop' buttons. Below the status information is a 'Results' section, which is currently empty. At the bottom right of the Results section is a 'Copy' button. At the very bottom of the window are 'Apply' and 'Exit' buttons.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Document printing overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

You can print documents by using either a local printer or a network-connected device. This article provides an overview of how documents are printed.

## Printing overview

The application provides integrated services and client applications that make it easy to generate, store, and distribute documents that support business activity. You can print documents by using either a local printer or a network-connected device. In addition, you can export pages and reports directly from the client, as PDF files or Microsoft Office documents. Finally, the distributed workload lets you print business documents directly from a mobile device by using network resources. Although printing requirements might vary, all industries typically must create hard copies of business documents by using the application. Printing documents on network devices from hosted applications presents a unique set of challenges. Here are some examples:

- Print drivers might not be available on the user's device.
- The user's device might not be connected to the corporate network.

By using a dedicated host and following a few easy steps, system administrators can configure deployments so that users can print directly from business applications on network devices.

### Application printing scenarios

The following table describes the three primary printing scenarios.

SCENARIO	GOAL	SOLUTION
1. Printing what you see	Print what is currently shown in the browser.	A "print-friendly" version of the webpage is generated for the browser.
2. Interactive printing	Print a precision document on a locally connected device.	You can export a PDF version of the report and download it to the browser.
3. Printing on a network device	Send a precision document to a domain printer device.	A precision document is sent to a client application that runs on a server that is hosted in the customer's domain.

Because the solution varies, depending on the scenario, applications provide built-in services and tooling to help users accomplish their goals:

- **Scenario 1** is supported by the browser's rendering of the HTML5 client.
- **Scenario 2** uses client applications and Microsoft 365 services.
- **Scenario 3** requires support from client applications and from services that are hosted in Microsoft Azure.

In addition to the platform that is deployed to the Azure subscription, Finance and Operations applications provide customers with an integrated, first-party Azure application that helps them more easily use domain-hosted devices to print documents.

## Service overview

While documents that are produced by the hosted applications are waiting to be printed on a network-

connected device, they are stored in Azure blob storage. The [Install the Document Routing Agent to enable network printing](#) uses Azure authentication to establish a secure channel to the Azure services.

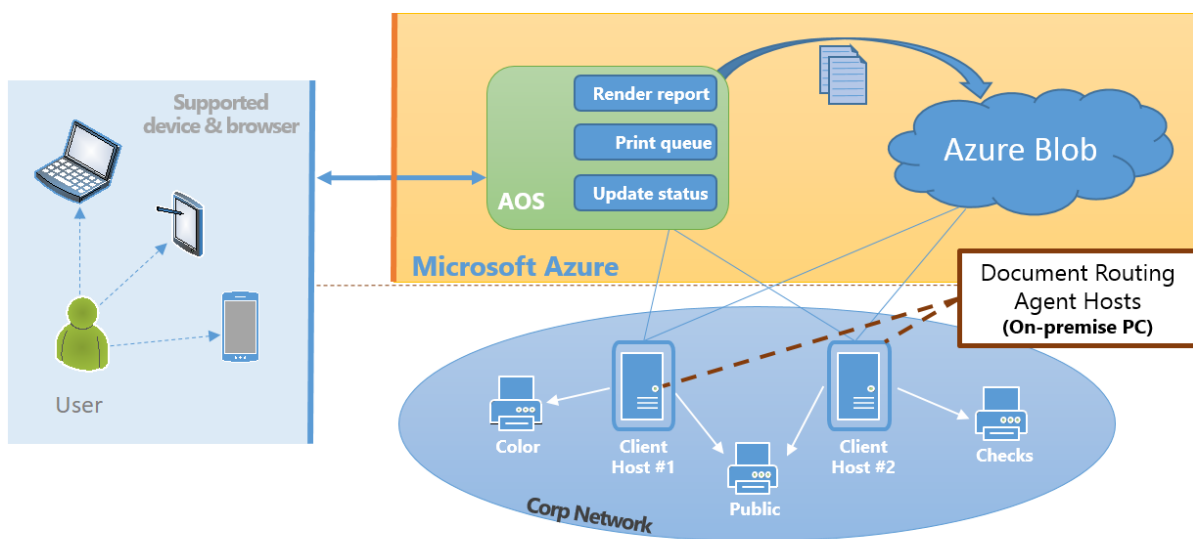
### Execution sequence

1. The report is generated by Microsoft SQL Server Reporting Services (SSRS) and stored in Azure blob storage. Attached printer settings are stored together with the document.
2. The Document Routing Agent queries the Azure Service Bus queue for active jobs.
3. The document is downloaded by the Document Routing Agent and spooled to the network printer.

The client-based solution lets customers manage the scale of their printing needs. Customers who have heavy-volume printing workloads can install many Document Routing Agents to increase the number of concurrent printing operations. Alternatively, some customers require very few installations of the Document Routing Agent to handle their anticipated printing needs.

### Service components for network printing

The following diagram shows the basic components that help support network printing operations.



Note that a single printer can be registered with multiple Document Routing Agents. To resolve the printer preferences, the hosted service uses the network path that uniquely identifies every network printer. As a result, even when a printer is registered by multiple clients, it appears as a single selection in the list of printers available in applications.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Install the Document Routing Agent to enable network printing

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to install and configure the Document Routing Agent (DRA). The DRA is a downloadable application that you can use to enable network printing scenarios. You can enable network printers for specific companies by using in-client administrative pages.

## Preparing to install the Document Routing Agent

- Supported on Windows 8.1, Windows 10, Microsoft Windows Server 2012 R2, or Microsoft Windows Server 2016.
- Access to network printing resources requires Active Directory Domain Services (AD DS) authentication.
- When installing the DRA, make sure you are logged in as the Admin user.
- The Microsoft Azure Active Directory (Azure AD) account that is used to configure the DRA must share the same domain as the Azure tenant.
- The DRA requires .NET 4.62 or later and Adobe Acrobat Reader on the client.
- Configure Adobe client print settings to prevent document scaling.

Network printers that are registered for applications can be used by all legal entities (also known as companies) that are defined in the environment. Network printer settings are company-specific. Therefore, administrators can restrict access, based on the user's active company. For example, users in the active company might have access to all the network printers that are registered by the Document Routing Agent. However, users in another company won't have access to those printers until access is explicitly enabled for that company.

## Key concepts

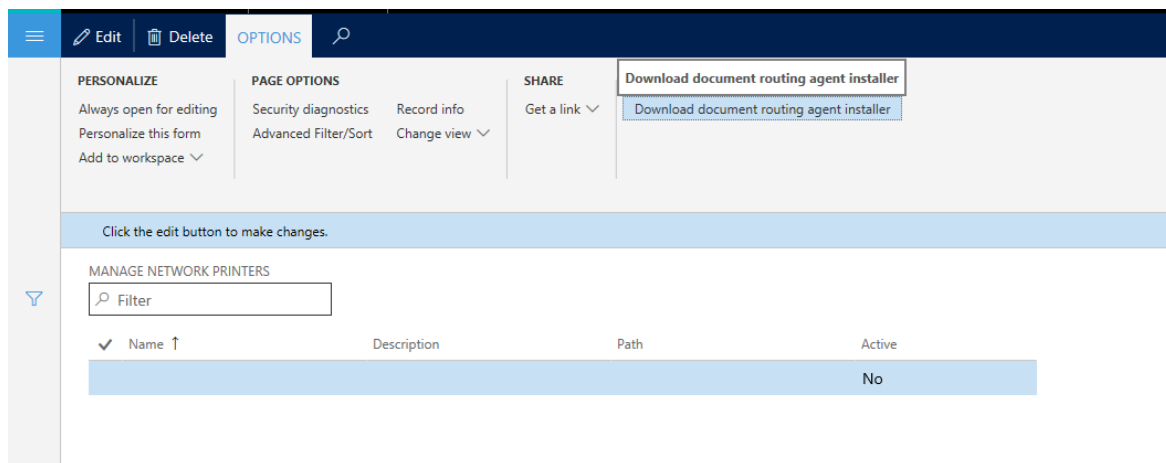
This topic will help you with the following tasks:

- Identify the key components that are involved in the support for network printing in applications.
- Learn about the function of the Document Routing Agent.
- Configure the Document Routing Agent to work against an existing application.
- Use administration pages to manage access to network printers.

## Install the Document Routing Agent

Applications use the Document Routing Agent to manage the spooling of documents to network printer devices. You can obtain the client by using direct links that are embedded in the web application. Use the following procedure to download the application to your local computer. You will then be able to access both local and network printers that are connected to your computer, from a single deployment.

1. Open the **Manage network printers** page (**Organization administration > Setup > Network printers**).
2. On the **Options** tab, in the **Application** group, click **Download document routing agent installer**.



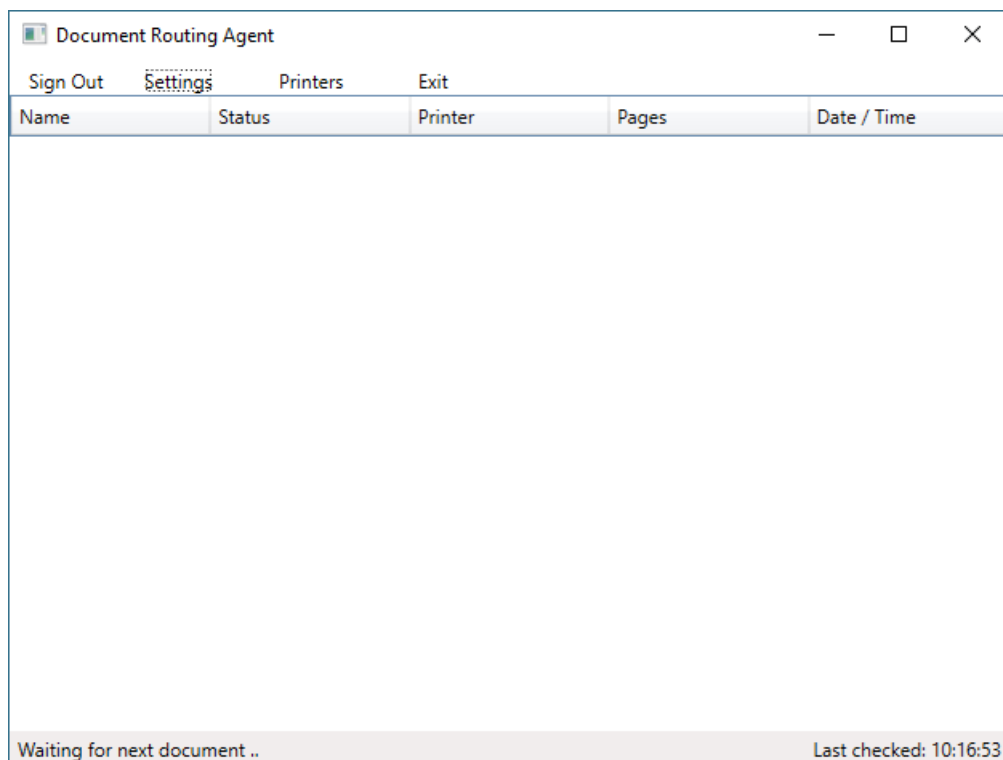
3. Run the downloaded file to begin the installation process.
4. Complete the setup process.

After the application is installed, you can begin to register local printers as network printers for the applications.

## Configure the Document Routing Agent

Use the following procedure to configure the client application so that it can communicate with the Azure services that host the documents that are in-flight.

1. Close all browser instances that are running the application. This resets the local Azure authentication tokens.
2. On your desktop, run the Document Routing Agent.
3. On the toolbar, click **Settings**.



4. Add the following settings:
  - **Application ID** – The ID that is unique to the application and should be entered automatically.
  - **Finance and Operations URL** – The base URL of the application.
  - **Azure AD tenant** – The domain name of the Azure AD.



5. Click **OK**.
6. Click **Sign In** to sign in to your account.

#### NOTE

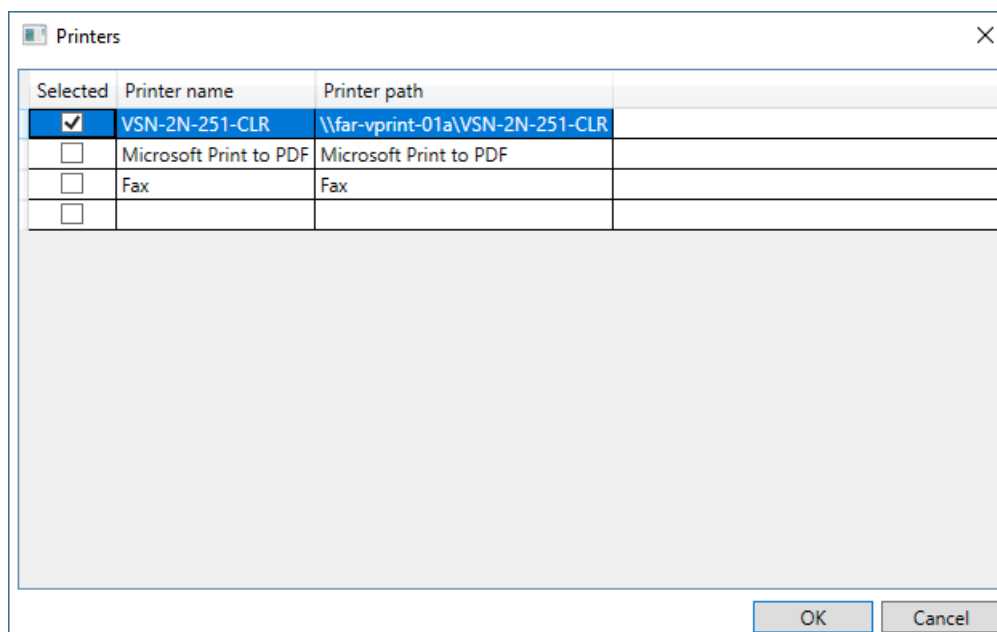
The account must share the same domain as the Azure AD that is associated with the application. The Document Routing Agent is now ready to process documents.

After you've successfully signed in, the **Printers** button becomes available on the toolbar.

## Register network printers

Before you complete this procedure, make sure that you've installed all the network printers on the local host computer. All the printer devices that are installed will be available for service registration. Be sure to select only the printers that you want to expose in the applications.

1. On the toolbar, click **Printers**.
2. Select the printers to make available in the applications.



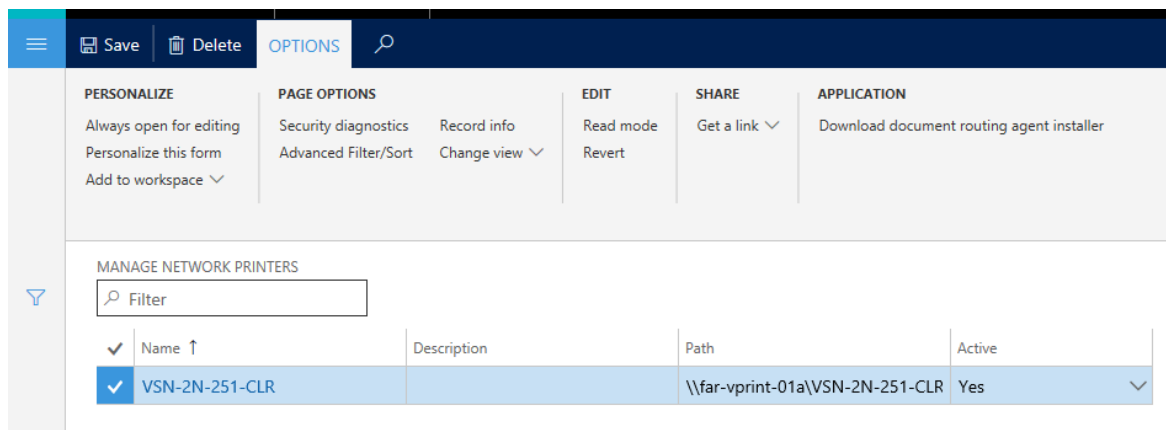
3. Specify a default name for the printer.
4. Click **OK**.

After you've completed this procedure, the selected printer devices are registered in the application's network printer catalog. System administrators can now enable the printers for access from within the application.

## Administer network printers

Use client pages to manage access to the network printers that have been registered by one or more Document Routing Agents. Network printers are uniquely identified by their path. Therefore, printers are listed one time, even if they have been registered by more than one Document Routing Agent. Use the following procedure to activate the Application Object Server (AOS) network printers.

1. Open the **Manage network printers** page (**Organization administration > Setup > Network printers**).



2. Edit the existing entries that are mapped to each network printer. As part of your changes, edit the connection path.
3. To include a printer as an option in the **Print Destinations** field, set the **Active** field to **Yes**.

The network printers can now be used in the application.

## Frequently asked questions

### Does the Document Routing Agent have to be installed on each computer where a user connects by using a browser?

No. Client installations of the Document Routing Agent can be shared by individuals who access the provisioned environment. We recommend that you install agents on one or more Print Servers or other domain-hosted clients that have access to network printers.

### If the Document Routing Agent belongs on a network Print Server, why doesn't the client run as a service?

The Document Routing Agent now supports running in the background as a service. You need to ensure that you have downloaded the latest version of the client. For more information, see [Run the Document Routing Agent as a Windows service](#).

### Do I need to update credentials or refresh Azure authentication tokens on a recurring basis?

Yes. The Azure Active Directory token must be refreshed every 90 days. Failing to do so will prevent the DRA from being able to authenticate and retrieve printing instructions applications.

### Is the Document Routing Agent supported on Microsoft Windows Server 2019?

Yes. The Document Routing Agent is supported on Microsoft Windows Server 2019.

#### NOTE

If the server is configured to prevent background service, the Document Routing Agent client will not be able to run as a service. For more information, see [Run the Document Routing Agent as a Windows service](#).

### Will Microsoft add support for Microsoft Windows Server 2008 servers?

No, not at this time. There are several dependencies on Azure capabilities that are available only in Microsoft Windows Server 2012 R2 and Microsoft Windows Server 2016.

### Does the user who installs the Document Routing Agent have to be part of a Finance and Operations apps security group?

Yes. To access the agent installation links, the user must be part of the **Document routing client** security role.

### How many network printers can the Document Routing Agent support?

The number of supported network printers depends on the number of legal entities and the number of network printers deployed. If you have fifty printers and one legal entity, a single Document Routing Agent can handle

the load (although you'd want more than one to ensure high availability). If you have a large number of printers and legal entities, we recommend that you do some performance testing to determine the number of Document Routing Agents that you'll need.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Update the Document Routing Agent

2/18/2021 • 2 minutes to read • [Edit Online](#)

The solution for managing the print job queue is designed to allow customers to properly scale Dynamics 365 Finance and Operations apps to satisfy high-volume printing requirements. Although public service endpoints used to manage print jobs are backward-compatible, we strongly recommend that customers update **all** existing Document Routing Agent (DRA) clients.

If you don't update existing installations of the DRA the most current version, you might experience issues such as:

- Observable performance degradation in applications
- Document loss that is associated with orphaned print jobs
- Inconsistent handling of printed documents that have custom margins

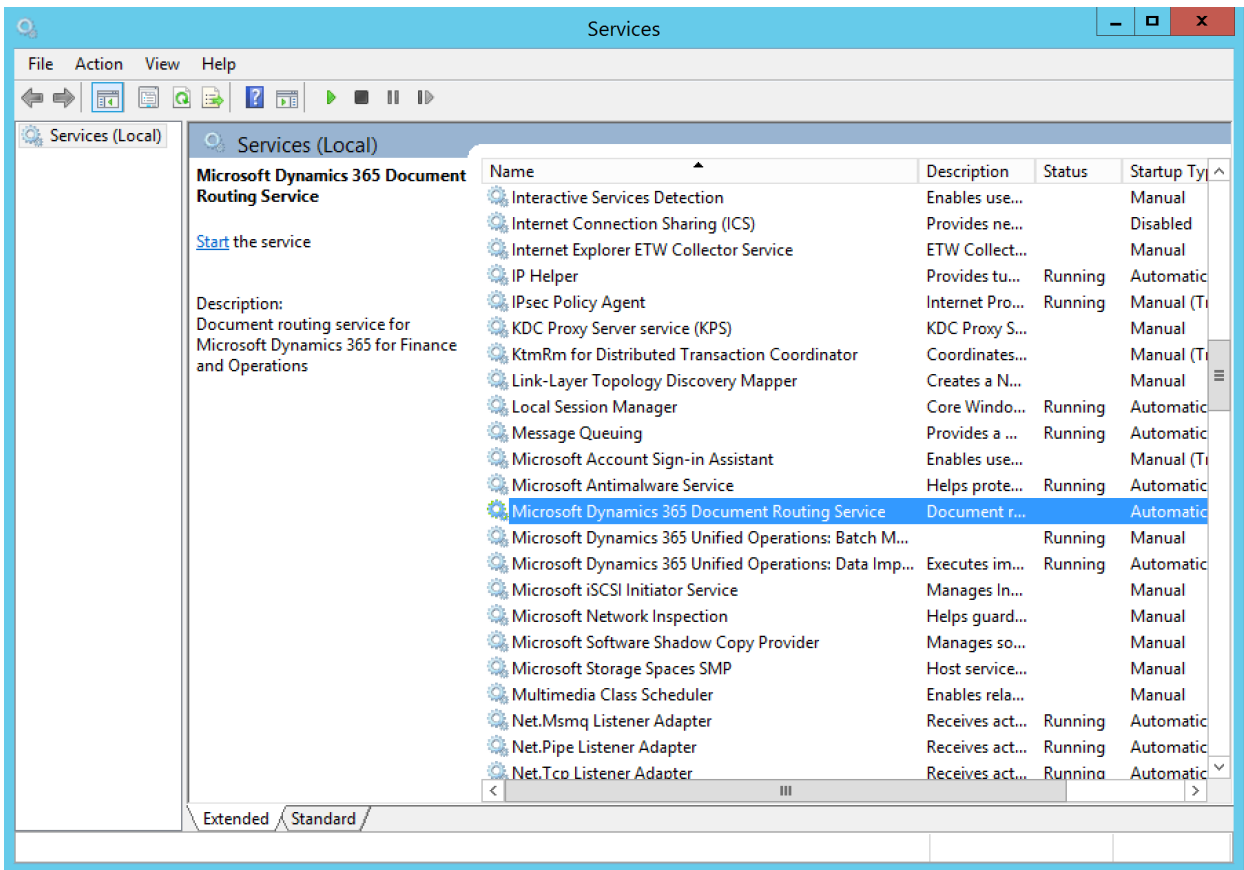
IT administrators must perform the following procedures on each domain resource that is used to host a DRA.

## NOTE

When you complete a DRA update, IT administrators should register any printers that are connected through the host server. For network printers that are identified by their network paths, if the paths have not changed, updates are not required.

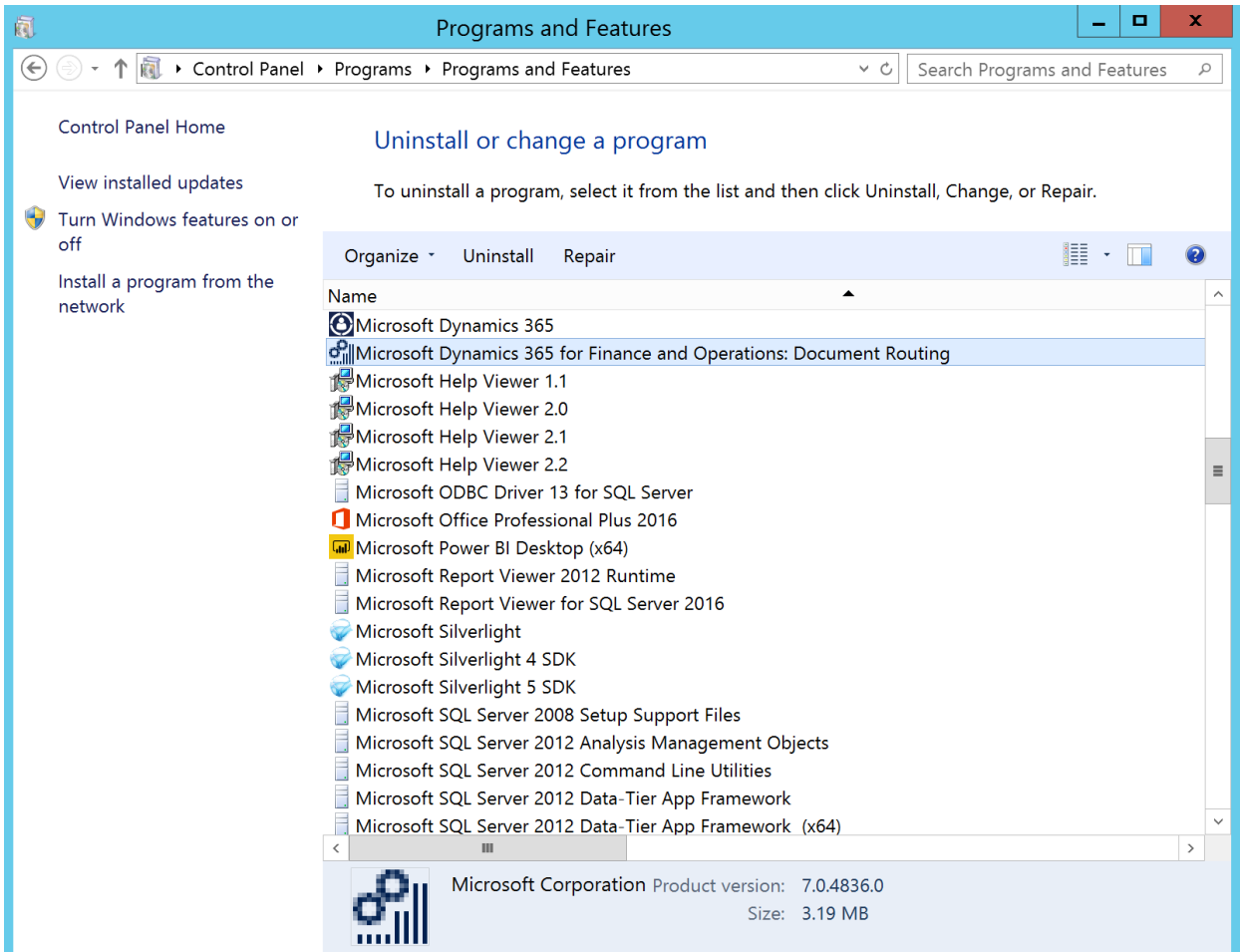
## Get started

To continue to run the DRA as a Microsoft Windows service, you must have both the user name and the password of the domain account that is used to run the service. This information must be available after the update is completed. To find the information for the active service account, start the Microsoft Management Console (MMC) Services snap-in, and select **Microsoft Dynamics 365 Document Routing Service** in the list.

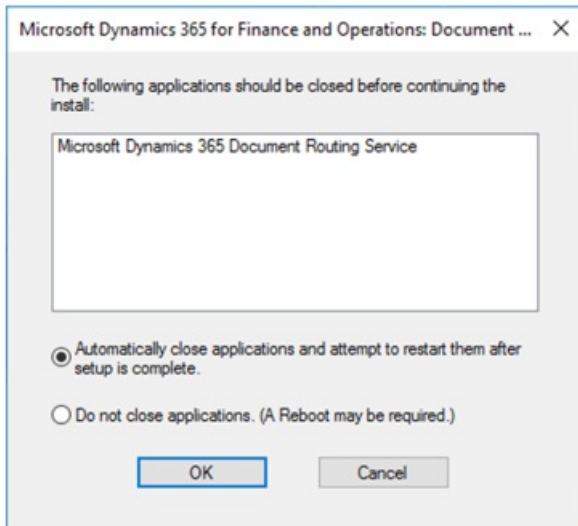


## Uninstall an existing Document Routing Agent

Open **Programs and Features**, and then find and uninstall **Microsoft Dynamics 365 for Finance and Operations: Document Routing**.



During the uninstallation process, if you're prompted to close the Microsoft Dynamics 365 Document Routing Service application, select **Automatically close applications and attempt to restart them after setup is complete**.



## Install the latest Document Routing Agent

For information about how to install the latest DRA that is available with your subscription, see [Install the Document Routing Agent to enable network printing](#).

### NOTE

Be sure to open the DRA client after upgrading to refresh network user credentials.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Install network printer devices in on-premises environments

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to connect an on-premises deployment of Microsoft Dynamics 365 Finance + Operations (on-premises) to existing network printer devices. Network printing in the on-premises application is supported by the [Print and Document Services](#) feature in Microsoft Windows Server 2016. This feature lets you centralize tasks that are related to printer management. To install and configure Print and Document Services, you must have administrative access to the server that hosts the primary instance of Application Object Server (AOS).

Two roles are associated with the configuration of network printing services:

- **Service Administrator** – The person who has this role is responsible for installing and configuring components of the platform infrastructure. Traditionally, this role is an Active Directory account that has elevated domain privileges. It has enough privileges to install components of Microsoft Windows Server.
- **Organization Administrator** – The person who has this role manages application security privileges. This Active Directory account is assigned to the **System Administrator** role.

Before the organization administrator can begin to add network printers, the service administrator must install and configure Print and Document Services on the server that hosts the primary AOS instance. The organization administrator can then begin to use built-in tools to configure network printer devices.

## Install and configure Print and Document Services

The environment administrator uses the information in this section to enable network printing services.

1. Install Print and Document Services by following the instructions in [Install Print and Document Services](#).
2. Configure Print and Document Services by following the instructions in [Configure Print and Document Services](#).
3. Follow these steps for each server that is used to host the AXService application:
  - a. On the local server, start the **Local Users and Groups** manager.
  - b. Select the **Groups** node.
  - c. Right-click **Print Operators**, and then select **Add to Group**.
  - d. Add the network Active Directory account that is used to run the AXService application to the group.
  - e. Install network printers by using the AXService user account. This step helps guarantee that the printer driver is available to the AXService user account.
  - f. Print a test page on the installed printers to make sure that all the connections are correctly configured.
  - g. Restart the AXService application to help guarantee that the user's profile is correctly loaded so that it can look up the printer driver.

## Manage network printers

The system administrator uses the information in this section to define network printers.

1. Go to **Organization administration > Setup > Network printers**.

2. On the **Network printers** page, add new printers. For each printer, specify a name, description, path, and status. Make sure that the printer path matches the network path of the installed printer.

Items that are marked **Active** immediately become available to application users, so that they can begin to print document-style reports on network printer devices.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Document generation, publishing, and printing in on-premises deployments

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the capabilities for generating, publishing, and printing documents in on-premises deployments of Microsoft Dynamics 365 Finance + Operations (on-premises). The application provides a fully integrated experience for enterprise document generation that is powered by Microsoft SQL Server Reporting Services (SSRS). From any supported device, users can produce standard industry documents that are linked to business processes. These documents include sales invoices, checks, and packing slips. Built-in tools let administrators configure the service so that users can securely connect to network printers.

You can upgrade solutions that are built on the Microsoft Dynamics AX 2012 SQL Reporting Services framework, or you can take advantage of the modern solutions that are available in [Microsoft Dynamics Lifecycle Services \(LCS\)](#).

## Document publishing services: secure, reliable, and convenient

Employees spend lots of time on the go. Therefore, businesses depend on their employees' ability to stay productive while they work remotely. However, even today, documents remain critical for business transactions and record keeping.

From their mobile devices, users can print documents on network printers. Users can also automate the creation of business documents and use built-in tools to configure instructions for routing documents to multiple recipients.

The following options are available for document publishing:

- **Email** – Distribute mail via a server, and link reports as attachments.
- **Archive** – Store reports for future reference and regulatory compliance.
- **File** – Produce a PDF file that is downloaded directly to the browser for local printing.
- **Print** – Send documents directly to network printers from all supported platforms. These platforms include mobile devices.

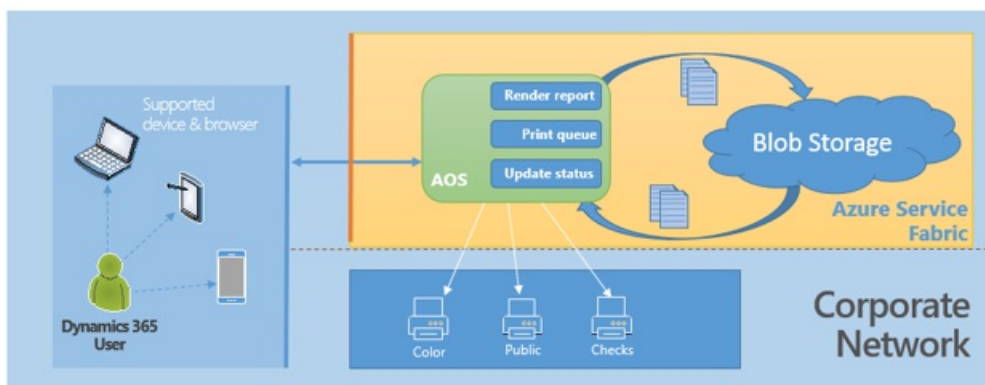


For a high-level summary of the options for information access that are available in the cloud-hosted solution, see [Document printing overview](#).

## Comparing the cloud-hosted and on-premises services

Unlike the cloud-hosted service, the on-premises publishing service produces documents as PDF files that are automatically downloaded to the browser. Therefore, users can save documents or print hard copies by using local connected devices. Administrators can manage access to network printers directly from the application, by using built-in administrative pages. Users can interact with reports on demand, or they can schedule automatic jobs to securely generate and distribute documents on a recurring basis.

The following illustration shows the components that are involved in document printing.

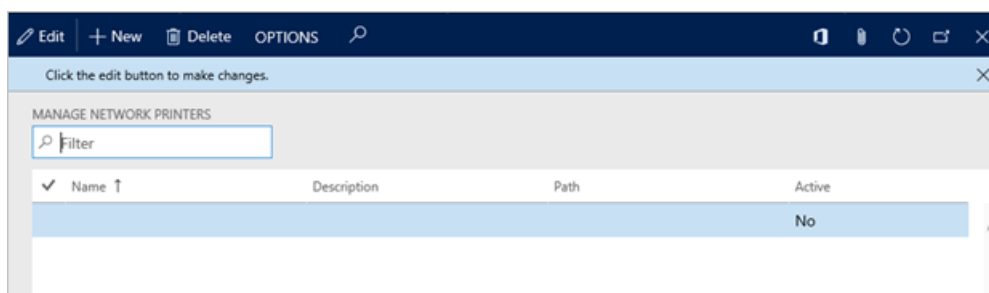


For information about how to use extensions to manage availability of the embedded drill-through links in application reports, see the Appendix.

## Managing access to network printers

Administrators can use built-in administrative pages to manage access to network printers. Network printers are secured per company and shared by users of the application. Documents are then printed by using a privileged domain account, based on settings that the user provides. In on-premises deployments, you don't have to install an adapter to connect to domain resources such as printers and fax machines.

The following illustration shows the page that is used to manage network printers.



## Appendix

### Turning on embedded links in business documents

Here is the code that you can use to make embedded drill-through links available in PDF documents.

```
class Controller extends SrsReportRunController
{
    protected void preRunModifyContract()
    {
        this.parmReportContract().parmRdlContract().parmEnableFileDrillThrough(true);
        super();
    }
    static void main(Args _args)
    {
        ...
    }
}
```

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Run the Document Routing Agent as a Windows service

2/18/2021 • 5 minutes to read • [Edit Online](#)

The Document Routing Agent includes an option that lets you select the mode of execution. The process can run as either a desktop application or a Microsoft Windows service. When the application runs as a Windows service, it can be started automatically after a computer restart. It can also be configured to run under the security context of a specific user account. This enhancement lets customers host the Document Routing Agent on secured domain resources such as network print servers.

This topic provides important information that will help you select the correct execution mode.

## Service applications

An application is a program that a user interacts with on the desktop. A service is a process that runs in the background and doesn't have an active window. The Document Routing Agent now supports both execution modes. It's important that you understand why you might select one mode instead of the other and the steps that are involved in running the process as a service. For more information about Windows services, see [Introduction to Windows Service Applications](#). Here are some of the main benefits of running the Document Routing Agent as a background service:

- The service can be configured to start automatically after a computer restart. No user intervention is required.
- The service runs in the background. No active application runs in the notification area.
- The service routes documents without requiring that a user sign in by using cached credentials.

Although there are many benefits of running the Document Routing Agent as a Windows service, there are also limitations. The next section discusses an issue that affects the handling of document reports, such as checks, that require custom page margins.

## Documents that require custom margins

When the Document Routing Agent runs as a Windows service, document reports, such as checks, that require custom margins can't be printed directly to network printers. Instead, the Document Routing Agent automatically routes those documents to a target folder. New configuration properties in the application's **Settings** dialog box let you define the target location for document reports that require custom margins.

When the Document Routing Agent runs as a desktop application, it continues to take advantage of Adobe Reader to spool the document to the shared printer device that is selected in Finance and Operations. To handle scenarios where documents that have custom margins must be printed, we recommend that you install the Document Routing Agent in multiple locations. Then install the printers that will handle those documents only on the Document Routing Agents that will run in desktop application mode. Alternatively, use a post-execution process to pick up the files in the target directory and direct them in the appropriate manner.

## Install the latest build

1. Save a copy of the current Document Routing Agent configuration file. This file is located at C:\Users\\AppData\Local\Microsoft\Microsoft Dynamics 365 for Finance and Operations Document Routing\Microsoft.Dynamics.AX.Framework.DocumentRouting.config. In this path, <UserID> is the Active Directory Domain Services (AD DS) user name that the Document Routing Agent was installed under.

2. Uninstall the current version of the Document Routing Agent.
3. Install the latest version of the Document Routing Agent by following the instructions in [Install the Document Routing Agent to enable network printing](#).

#### NOTE

Although you install the application at this point, don't run it yet.

4. Copy the configuration file from step 1, and paste it into the following directory:  
C:\ProgramData\Microsoft\Microsoft Dynamics 365 for Finance and Operations Document Routing. This step helps guarantee that all your previous configuration settings are used for the new version of the Document Routing Agent application.
5. Run the Document Routing Agent.
6. Sign in by using your Microsoft Azure Active Directory (Azure AD) or your credentials for your Finance and Operations apps.
7. View and verify your settings and printers by clicking the appropriate menu items.

The next section provides detailed instructions for selecting the Windows service execution mode.

## Change the default execution mode

By default, the Document Routing Agent runs as a desktop application. To run the process as a Windows service, make sure that you're familiar with the process for [installing the Document Routing Agent to enable network printer devices](#). Then complete the following tasks.

### Update the execution mode for the Document Routing Agent

1. Start the Document Routing Agent by using the desktop icon.
2. Select the **Sign In** option, and sign in by using your Azure AD credentials.
3. On the ribbon, select **Settings**.
4. Enable the **Run as a Windows Service** option.
5. Provide a target folder for PDF files that are produced for documents that have custom margins.
6. Select **OK**, and close the Document Routing Agent window.

### Configure and start the Windows service

1. In Windows, start Service Manager.
2. In the list, select **Microsoft Dynamics 365 for Finance and Operations Document Routing Service**.
3. Right-click the name, and then select **Properties**.
4. On the **Log On** tab, select the **This account** option, and then supply the AD DS credentials that are used to run the service.

#### NOTE

The selected account must have access to the shared network devices. The windows domain account ( or local machine account) used to run the windows service must be same as the account that starts the Document Routing Agent desktop app.

5. Select **OK**.

6. Start the service.

The Document Routing Agent is now running as a Windows service.

## Troubleshooting tips

### Verify the network printer connection

- Verify that the active account has enough access rights to the network printer.
- Verify that the user account can successfully print to the network device by using Notepad or another local application.

### Verify security roles

- To access the application links that are used to install the Document Routing Agent, the user must be part of the **Document routing client** security role.

### Review the service account's access rights

- Verify that the **DocumentRoutingService** service is running as a domain account that has access to the network devices.

### Refresh the Azure service token

- Azure authentication tokens must be **refreshed every 90 days** while the Document Routing Agent is running as a Windows service. To refresh the service token, start the client, and then sign out and sign back in by using the menu items.

### Disable shared printers for remote access

- When you connect to the host machine by using Microsoft Remote Desktop, make sure that you clear the **Printers** option in the **Local devices and resources** section on the **Local Resources** tab.

### Review the event logs

1. On the host machine, start Event Viewer.
2. Review the logs at **Application and Services Logs > Microsoft > Dynamics > AX-DocumentRouting**.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SQL Server Reporting Services (SSRS) reports that are available

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic points you to a report that lists the SQL Server Reporting Services (SSRS) reports available.

Reports can be defined simply as any visualization of a structured data set. This may include transactional data presented in a tabular layout and advanced graphical views of aggregate information. To account for this broad definition, the application offers several tools to produce reports to satisfy complex business requirements. One of these tools is SQL Server Reporting Services. SSRS reports provide the following advantages:

- Back office document management capabilities including email support, scheduled executions via batches, and print archive functionality.
- Parameterized views with drill-through navigation to application pages and other reports.
- Used to produce precision documents for compliance with local regulatory business practices.

For more information, see [Create reporting solutions](#).

## To view the report

The **SQL Server Reporting Services Reports report**, included with the [Technical reference reports](#), lists each SSRS report that is available. The report indicates the data set used for each report, as well as the filters and fields available on each report.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Install modern report design templates

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how to install the modern report design templates in the application suite. You can use these samples to create graphically rich business documents that have flexible branding in the header and footer.

## Introduction

A new set of developer tools is available that takes the form of report designs for several core business documents in the application suite. These report designs have been re-imagined so that flexible branding appears in the header and footer of public-facing documents when transactions are generated in the application. The following illustration shows how an earlier design for a sales invoice differs from a modern sales invoice design.

The image shows two side-by-side sales invoice reports. The left report, titled 'LEGACY DESIGNS', is a 'SalesInvoice Report' for Contoso Entertainment Systems (West). It features a dense layout with multiple contact addresses, a detailed item list with columns for item number, description, quantity, unit, unit price, discount percent, discount, and amount, and a 'SalesInvoice Report' footer. The right report is a modern Microsoft invoice for Owl Wholesale. It features a clean layout with the Microsoft logo, a clear header with the invoice number (CIV-000676) and date (30 November 2012), a total amount of \$315,479.00, and a table of items with columns for item, description, quantity, sales price, discount, and amount. It also includes a 'METHODS OF PAYMENT' section and 'OTHER INFORMATION'.

After you complete the installation, you can use the built-in brand management tools to define brand settings that should be applied to the modern designs for application business documents. The brand management tools are available at [Organization administration > Setup > Document branding > Branding details](#).

## Why aren't these designs the default designs for the application suite reports?

We are maintaining the legacy solutions for two primary reasons:

- **Modern designs don't include code.** Although the legacy solutions use embedded Microsoft Visual Basic (VB) code to recognize configuration keys and honor regulatory requirements that vary by region, the modern report designs offer much less flexibility. The benefit of a simple design that has minimal code behind it comes at the expense of reusability across regions.
- **Modern designs aren't available for all business documents.** There is a gap between the supported business documents and the availability of modern report designs. Although the legacy designs aren't as



aesthetically pleasing, they provide a sense of consistency.

#### IMPORTANT

The simple modern designs are **not** recommended for all types of deployments. They are intended for cases where the customer doesn't require runtime control over the layout of the document through existing application configuration settings.

## Apply the modern designs

The modern report designs have been bundled into a model file and posted to Microsoft Dynamics Lifecycle Services (LCS). Therefore, you can easily access them from your existing subscription. Use the following procedure to obtain the modern report design solutions and install them in your local development environment. You must then apply some customizations to incorporate the modern report designs into the appropriate scenarios.

Follow these steps to install the modern report designs for the application suite.

1. Sign in to [LCS](#) to access the deployment dashboard. Then, on the **Shared asset library** page, select the **Model** asset type, and download the **ApplicationSuiteModernDesigns** model file. Save the model file to a location that is accessible from the development environment.

#### NOTE

Be sure to select the appropriate model file for the version of the application that you're using.

2. Import the model file into your local development environment. To install a model file in a development environment, use the ModelUtil.exe tool and the **-import** directive. Here is an example.

```
ModelUtil.exe -import -metadastorepath=[path of the metadata store] -file=[full path of the file to import]
```

3. Navigate to the `J:\AOSService\PackagesLocalDirectory\bin` folder.
4. Run the following command.

```
ModelUtil.exe -import -metadastorepath=J:\AOSService\PackagesLocalDirectory -file="E:\Test\AppSuiteModernDesigns.axmodel"
```

For more information about how to import model files, see [Export and import models](#). After you've imported the model file, start Microsoft Visual Studio. In Application Explorer, verify that the **Application Suite - Modern Designs** collection appears under the **AOT** node. For more information about how to use the Application Explorer, see [Development tools tutorial](#)

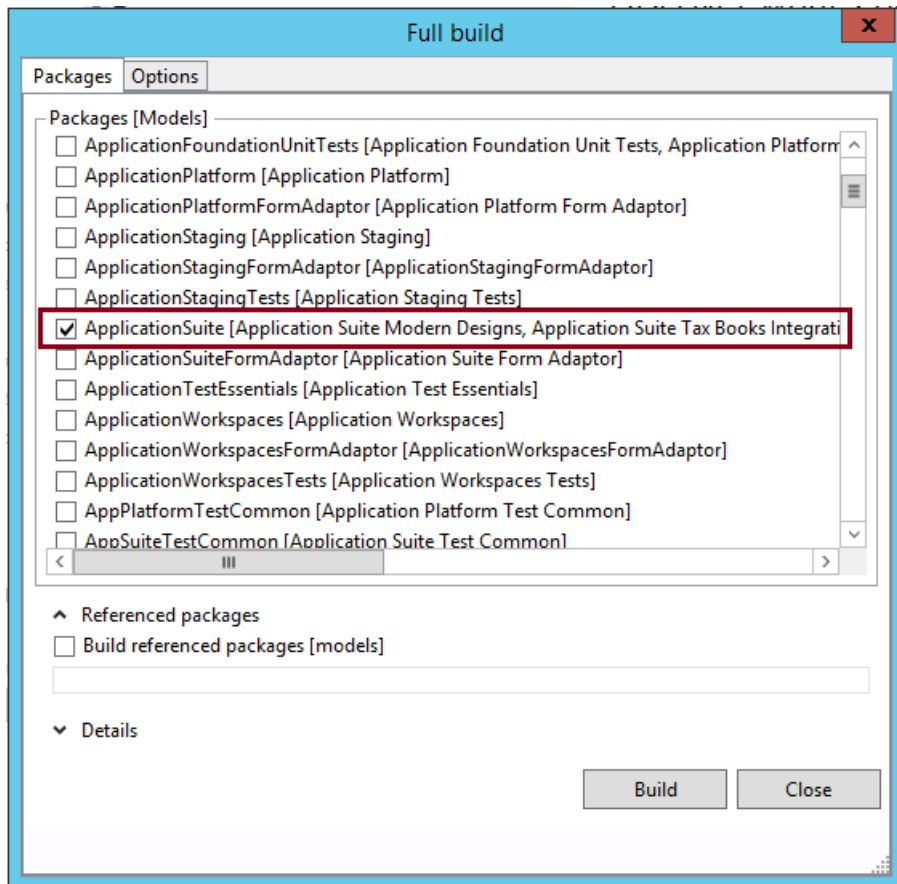
Now that you've successfully imported the Application Suite Modern Designs model, you must to rebuild the application suite to update the metadata elements.

## Build the application suite

The Application Suite Modern Designs model is an extension of the Application Suite model. To help guarantee that all application references are updated so that they target the model extensions, you must build the Application Suite model by using Microsoft Visual Studio.

1. Start Visual Studio, or use the existing instance.

2. On the **Dynamics 365** menu, select **Build models**.
3. In the list, select the check box for the **ApplicationSuite** package.



#### NOTE

You will see that the Application Suite Modern Designs model is included in the package definition.

4. Select **Build** to do a full build of the application suite.

This process may take up to 20 minutes, depending on the size of your machine.

## Deploy the modern designs (one-box environments)

After you've compiled the application suite that includes the modern report design templates, you should verify the changes locally. To verify the changes, you must deploy the new modern report design solutions to the instance of Microsoft SQL Server Reporting Services (SSRS) that is running locally.

Follow these steps to incorporate the modern report design into an existing application suite report.

1. Create a project that contains the application suite report. In Application Explorer, under the **Application Suite Modern Designs** model, expand the **Reports** node, and then expand the **Reports** subnode. Select all the items in the folder, right-click, and then select **Add to new project**.
2. Complete the **New Project** wizard. Accept all default values.
3. In Solution Explorer, select the project, right-click, and then select **Deploy reports** to deploy the build and deploy the reports locally.

When you add the modern report design to an existing report, you can reuse both the parameter handling and the data provider that the out-of-box solution uses.

# Update Print management settings

At this point, you should be able to access the modern report designs from the application. Make sure that you do thorough test validations on the modern report design templates before you deploy them to production environments. To do test validations, you must activate the modern report designs for the application business process.

Follow these steps to update the Print management settings for customer sales orders by selecting the modern report design solution as the default report design.

1. Open the **Form setup** page for the module. For example, for Accounts receivable, select **Accounts receivable > Setup > Forms > Form Setup**.
2. Select **Print management** to open the **Print management setup** page.
3. Expand the tree, and find the settings for the **Sales order confirmation** document.
4. Select **Original <Default>** to begin to modify the default document routing.
5. In the **Report format** list, select **SalesConfirmModern.Report** to enable the modern report design solution.

## Print management setup

The screenshot shows the 'Print management setup' page for the 'Sales order confirmation' document. On the left, a tree view shows the navigation path: 'Module - accounts receivable' > 'Documents' > 'Sales order confirmation' > 'Original <Default>'. The 'Original <Default>' option is selected. On the right, the 'ORIGINAL OR COPY IDENTIFICATION' section is visible. It includes a dropdown for 'Original / copy' set to 'Original', a 'Name' field with the value 'Suspend printing for this document', and a 'Report format' dropdown set to 'All countries/regions'. Below the dropdown, a list of report formats is shown: 'SalesConfirm.Report' and 'SalesConfirmModern.Report'. The 'SalesConfirmModern.Report' option is highlighted with a blue background and a red border. At the bottom of the page, the text 'SalesConfirm.Report' is visible.

6. Open another page. This step forces a save operation to occur.
7. Post a sales order to view the modern design in the application.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Power BI integration with Entity store

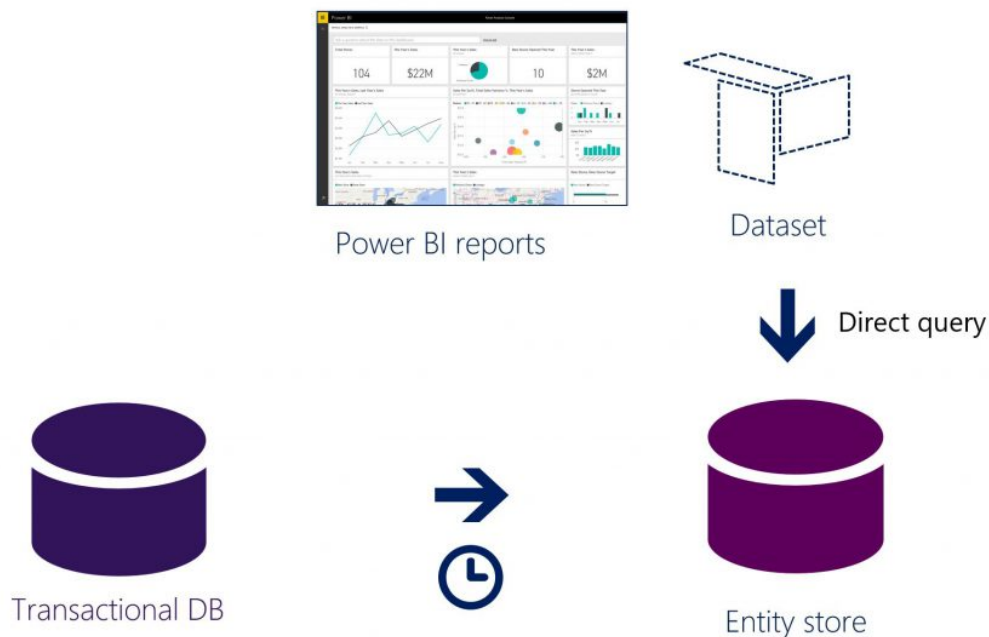
2/18/2021 • 4 minutes to read • [Edit Online](#)

Entity store is an operational data store that is included with Microsoft Dynamics 365 Finance. This topic describes how Entity store enables Power BI integration.

Entity store is an operational data store that is included with the application. The Entity store feature was introduced in the Microsoft Dynamics AX platform update 1 (May 2016) release. This feature lets an administrator or power user stage aggregate measurements in a dedicated data store for reporting and analytics. (Aggregate measurements are a star schema that is modeled by using entities.) We call this data store Entity store. It's a database that is optimized for reporting purposes. Entity store uses the in-memory, clustered columnstore index (CCI) functionality that is built into Microsoft SQL Server to optimize reporting and queries. Customers can use Microsoft Power BI DirectQuery models together with Entity store to enable high-volume, near-real-time analytical reporting over large volumes of data.

## Power BI DirectQuery mode

In the February 2016 release of Microsoft Dynamics AX, you could create Power BI reports by using OData endpoints that are exposed via data entities (both aggregate data entities and detailed or regular data entities). Although this approach is still supported, Entity store also lets power users create Power BI DirectQuery reports.



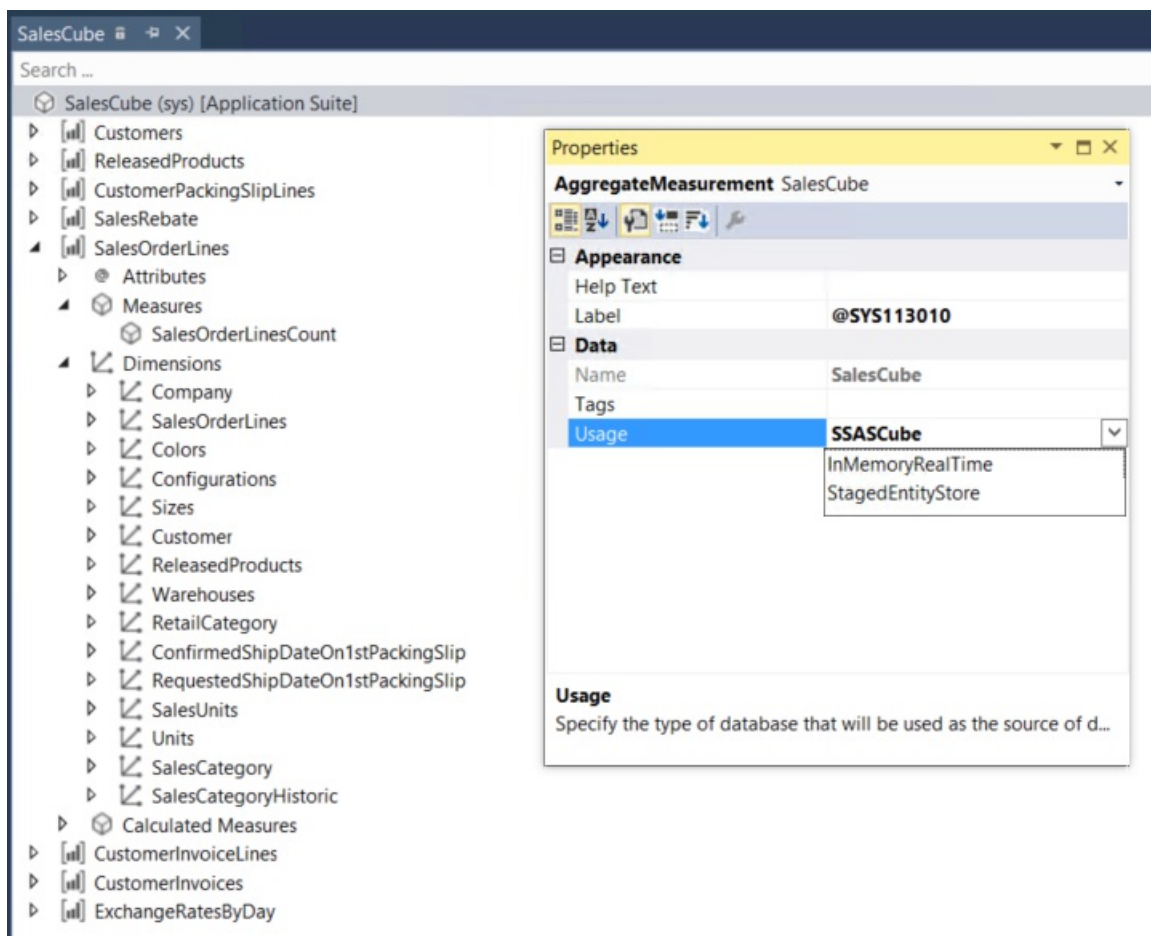
As the preceding illustration shows, DirectQuery is a reporting mode that runs reports directly on Entity store. In this reporting mode, data isn't staged in Power BI caches. This mode provides two immediate benefits:

- You can create Power BI reports over large data volumes.
- Reports don't have to be updated by using Power BI. When Entity store is updated, reports reflect the latest data.

Additionally, data doesn't leave your environment, because no data is cached in the Power BI service.

## Stage aggregate measurements in Entity store

Aggregate measurements are a star schema that is modeled for analytical scenarios. In the February 2016 release, we enabled real-time, in-memory aggregate measurements. By using real-time aggregate measurements, you can enable embedded charts and key performance indicators (KPIs) that react to real-time operations on data. For information, see [Transition from Analysis Services cubes to aggregate models](#). Real-time aggregate measurements take advantage of the in-memory, non-clustered columnstore index (NCCI) technology. Visuals and aggregate calculations that are built over real-time aggregate measurements reflect transactions within seconds. In the platform update 1 (May 2016) release, we enabled aggregate measurements that can be staged in Entity store. Aggregate measurements that are staged in Entity store can be used for near-real-time analytical scenarios where large volumes of data must be explored by using Power BI. As a developer, you learned how to model an aggregate measurement for real-time analytics in [Model aggregate data](#). In the platform update 1 (May 2016) release, we also added the capability to model aggregate measurements that can be staged in Entity store. In Microsoft Visual Studio, you can specify **StagedEntityStore** as the usage property of an aggregate measurement. This new property was added in May 2016. Previously, **InMemoryRealTime** was available as the usage property.



However, you might wonder why you would model an aggregate measurement so that it can be staged? Why wouldn't you use in-memory real-time aggregate measurements all the time? There are several reasons for using the **StagedEntityStore** pattern:

- There might be large amounts of data that must be explored and analyzed.
- You might have Analysis projects (that is, cubes) that you migrated from Microsoft Dynamics AX 2012 R3 as part of the code upgrade process. Because of the complex views and joins that are present in the schema, query response times might not be acceptable for embedded visuals. However, you might not want to refactor the visuals to take advantage of NCCI technology immediately.
- Unlike the operational database schema, the schema in Entity store is modeled specifically for reporting. Therefore, it's much easier to build new reports from the schema in Entity store.
- Your scenario might not require that analytical data be updated within seconds of an operation. Most Power BI reports that are built to enable data exploration fall into this category. If data freshness of approximately

10 minutes is acceptable for your scenario, you might want to use the staged pattern.

If one of the preceding reasons covers your situation, you should stage your aggregate measurement in Entity store and it use for Power BI integration.

## Update Entity store

Entity store refresh is automated and managed by the system. In the client, you can find the **Entity Store** page at **Systems administration > Setup > Entity Store**. For more information, see [Automated Entity store refresh](#).

### Connecting to the Entity store database

For troubleshooting and diagnostics, you can connect to the Entity store database directly from a related sandbox environment. To connect:

1. Use Remote Desktop to access the sandbox. The RDP file can be downloaded from the **Environment Details** page after you have included your IP address in a safe list.
2. Open SQL Server Management Studio, and connect to the server specified on the **Environment Details** page.
  - Find the section titled **Database Accounts**. Locate the entry for the user with the name **axdwadmin**.
  - The server name is the first portion of the **SQL Server\Database Name** field. This should be used in the format of **SQLServerName.database.windows.net** where **SQLServerName** is the value from **LCS**.
  - The authentication type should be changed from Windows Authentication to SQL Server Authentication.
  - The login will be **axdwadmin** and the password will be the value from **LCS**.
3. Using the **Options** button or by browsing to the **Connection Properties** tab, change the **Connect to database** property from the default value to your **Database Name** value from **LCS**.
4. Click **Connect** to access the database.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Resolve issues after entity store maintenance

2/18/2021 • 2 minutes to read • [Edit Online](#)

When maintenance is performed on the entity store, it impacts the following components:

- Application analytical workspaces.
- Entity store-based reports that have been deployed to PowerBI.com.

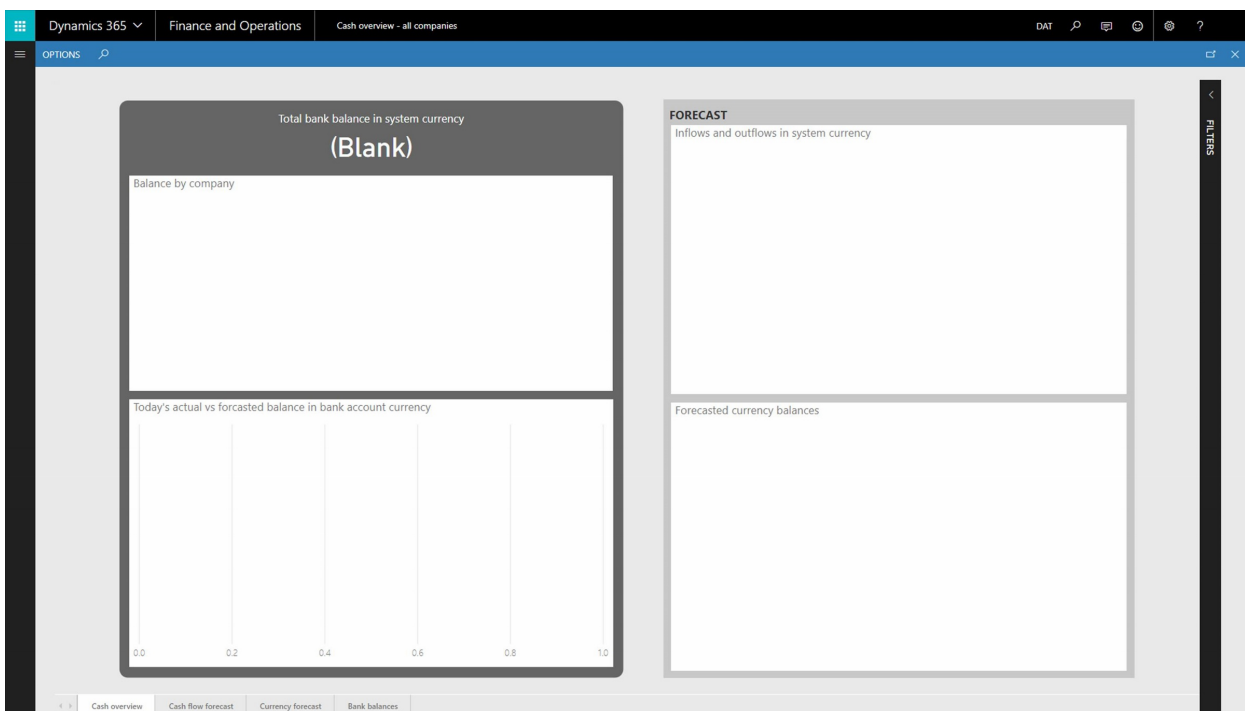
To resolve issues with these components, complete the procedures in this topic.

## NOTE

There will be **no impact** to the normal operation of your application.

## If you are using application analytical workspaces

Application analytical workspaces and reports may not render data after certain maintenance operations are completed. The following screenshot shows an example of this.



To resolve this issue:

1. Sign in to the application.
2. Go to the **Batch jobs** page (**System administration** > **Inquiries** > **Batch jobs**).
3. Delete all pending batch jobs associated with the entity store. These batch jobs:
  - Will have a status of **Waiting**.
  - Will typically have a description of **Deploy measurement**.

#### NOTE

The default description is **Deploy measurement**. If the description has been customized, you can verify whether a batch job is associated with the entity store by looking at the class name. Batch jobs associated with the entity store will have a class name of **BIMeasurementDeployManagementEntityBatchJob**.

4. Go to the **Entity store** page (**System Administration > Setup > Entity Store**).
5. Select all measurements that need to be refreshed.
6. Click **Refresh**, and then click **OK**.

After the refresh completes, the application analytical workspaces and reports will render data.

## If you have deployed entity store-based reports to PowerBI.com and are using the reports within PowerBI.com

After refreshing the entity store (as described above), redeploy the reports using the **Deploy Power BI report files** page by selecting **System Administration > Setup > Deploy Power BI files**.

#### NOTE

Reports that were previously deployed to PowerBI.com may produce errors. If this occurs, you may need to delete and redeploy the report after the maintenance activity is completed.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure PowerBI.com integration

2/18/2021 • 5 minutes to read • [Edit Online](#)

## Overview

Users can pin tiles, dashboards, and reports from their own PowerBI.com account to application workspace.

This functionality requires a one-time configuration of your environment. An administrator must do this step to enable Microsoft Power BI to communicate and authenticate correctly.

For a workspace to show a Power BI tile, the server must contact the Power BI service on behalf of a user and access the visualization. It must then redraw the visualization in the application workspace. The fact that the server contacts the Power BI service "on behalf of a user" is important. When a user, such as `D365User@contoso.com`, contacts the PowerBI.com service, Power BI should show only tiles and reports from the user's PowerBI.com subscription.

By completing this configuration step, you enable to contact the PowerBI.com service.

## Things you must know before you start

- You must be a system administrator in the application. This option is available on the **System administration** menu.
- You must have a PowerBI.com account. You can create a trial account if you don't have an account. (A Pro license isn't required for this configuration step.)
- You must have at least one dashboard and one report in your Power BI account. Although the dashboard and report aren't required for this configuration step, you might not be able to validate the configuration if you don't have any content in your PowerBI.com account.
- You must be an administrator for your Microsoft Azure Active Directory (Azure AD) account. If you aren't the administrator, an administrative user must perform this configuration step for you.
- The Azure AD domain that is configured must be the same domain that you used for your PowerBI.com account. For example, if you provisioned the application in the Contoso.com domain, you must have Power BI accounts in that domain, such as `Tim@ContosoAX7.onmicrosoft.com`.
- Users navigating to these pinned tiles must have a valid Power BI.com license.

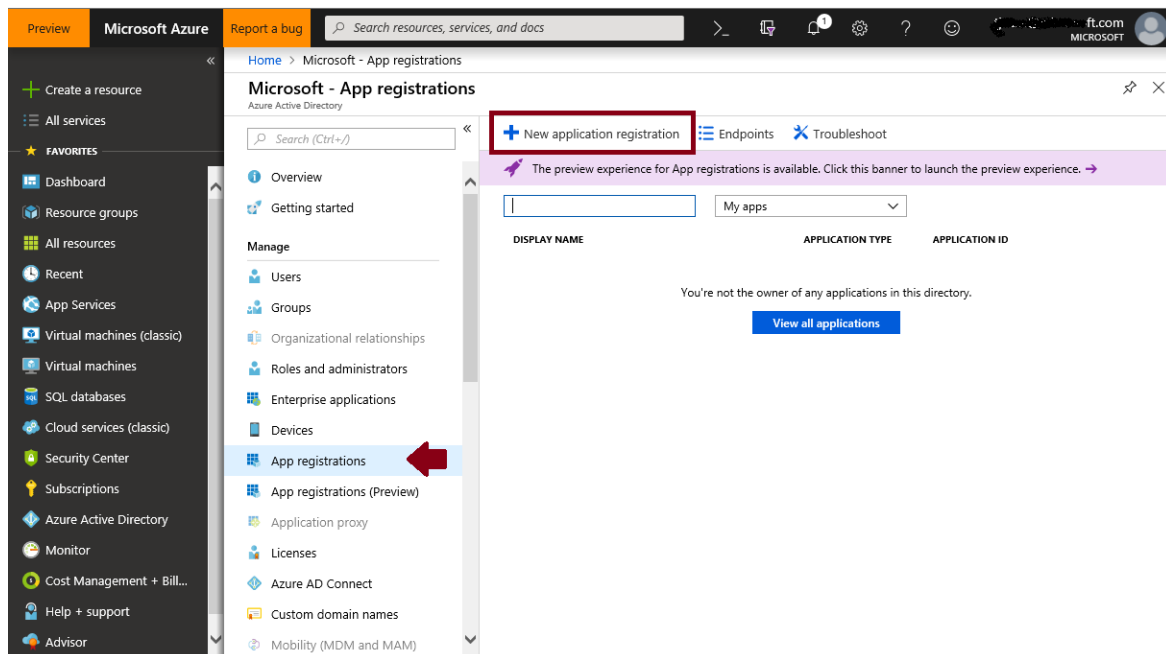
## Registration process

1. Sign in to <https://portal.azure.com/> using an Azure tenant admin account.

### NOTE

The user who completes this procedure must have Admin rights for the tenant to register applications.

2. Go to **Azure Active Directory > App registrations > New application registration**.



3. Enter the following values:

- **Name** - Your app name.
- **Application type** - Web app/API
- **Sign-on URL** - The base URL of your client. For example, `https://contosoax7.cloud.dynamics.com` .

#### NOTE

Depending on your version, you may need to add /oauth as a suffix to the URL, or use http instead of https as the protocol, such as: `https://contosoax7.cloud.dynamics.com/oauth` or `http://contosoax7.cloud.dynamics.com/oauth` .

4. Click **Create**.
5. Copy the **Application ID**. This will be used to connect to the PowerBI.com service.
6. Click **Settings** > **Required permissions** > **Add** > **Select an API** > **Power BI Service (Power BI)**.
7. Click **Select**.
8. Enable Access and click **Select**.

## Enable Access

<input type="checkbox"/>	APPLICATION PERMISSIONS	↑↓	REQUIRES ADMIN	↑↓
	View all content in tenant		✔ Yes	
<input checked="" type="checkbox"/>	DELEGATED PERMISSIONS	↑↓	REQUIRES ADMIN	↑↓
	View all datapools		✘ No	
	Read and write all datapools		✘ No	
	Read and Write all Reports		✘ No	
	View users Groups		✘ No	
<input checked="" type="checkbox"/>	View all Groups		✘ No	
<input checked="" type="checkbox"/>	View all Reports (preview)		✘ No	
<input checked="" type="checkbox"/>	Create content (preview)		✘ No	
<input checked="" type="checkbox"/>	View content properties (preview)		✘ No	
<input checked="" type="checkbox"/>	Read and Write all Datasets		✘ No	
<input checked="" type="checkbox"/>	View all Datasets		✘ No	
<input checked="" type="checkbox"/>	View all Dashboards (preview)		✘ No	
<input checked="" type="checkbox"/>	Add data to a user's dataset (preview)		✘ No	
	Read and Write all Dashboards		✘ No	
	View all content in tenant		✔ Yes	
	Read and write all workspaces		✘ No	
	View all workspaces		✘ No	

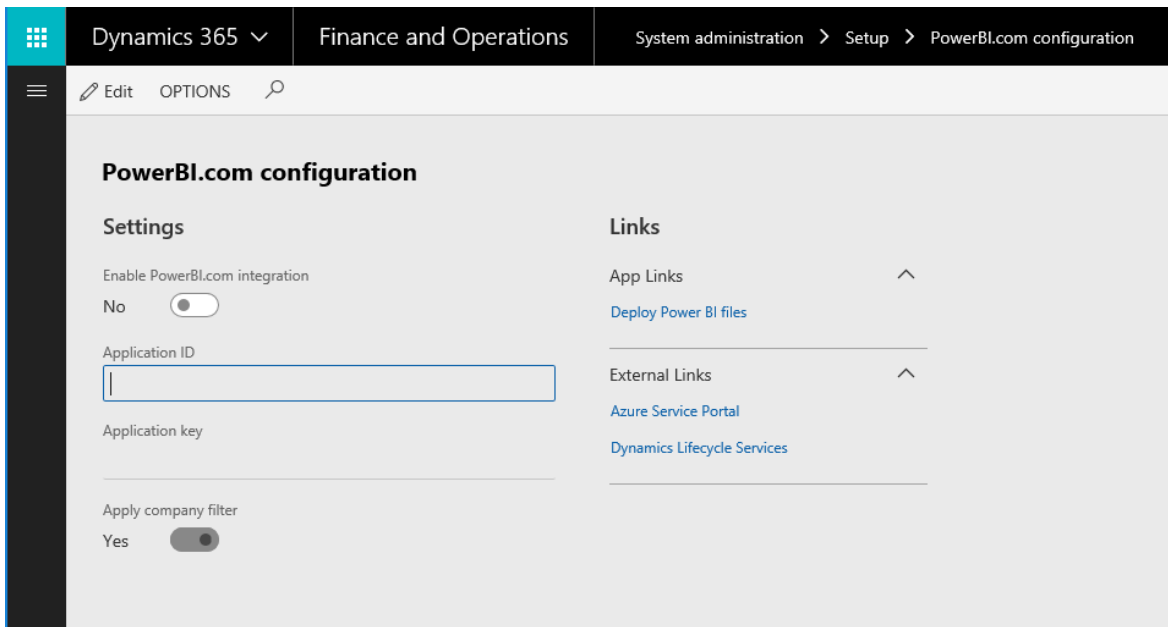
Select

9. Click **Done** and then click **Grant Permissions**.
10. Click **Settings > Keys**.
11. Enter a value for **Key description** and **Expires**, and then click **Save**.

Make a note of the **Application ID** and **Application Key**. You will use these values in the next procedure.

## Specify Power BI settings in Finance and Operations

1. In the client, open the **Power BI configuration** page.



2. Select **Edit**.
3. Set the **Enabled** option to **Yes**.
4. In the **Application ID** field, enter the **Application ID** value that you got from Power BI in the previous procedure.
5. In the **Application Key** field, enter the **Application Key** value that you got from Power BI in the previous procedure.

You can apply the company filter only if your Power BI content has a table that is named **Company** and a column that is named **ID**. Ready-made Power BI content that is released uses this convention.

6. Click **Save**.

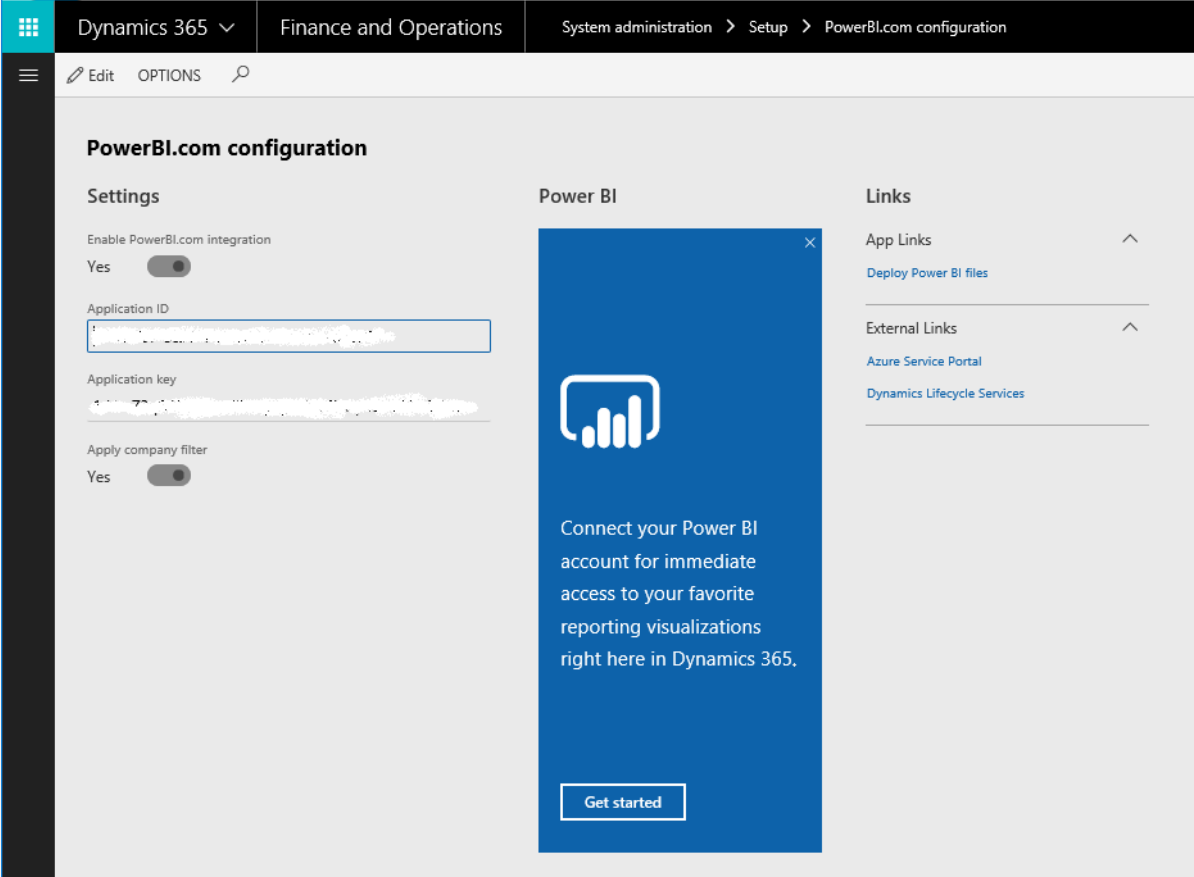
Complete the steps in the next section to verify the changes and enable PowerBI.com integrations.

## Pin tiles to a workspace

1. To validate the PowerBI.com configuration, click **Get started**.

## NOTE

You may need to refresh the browser to apply the changes.

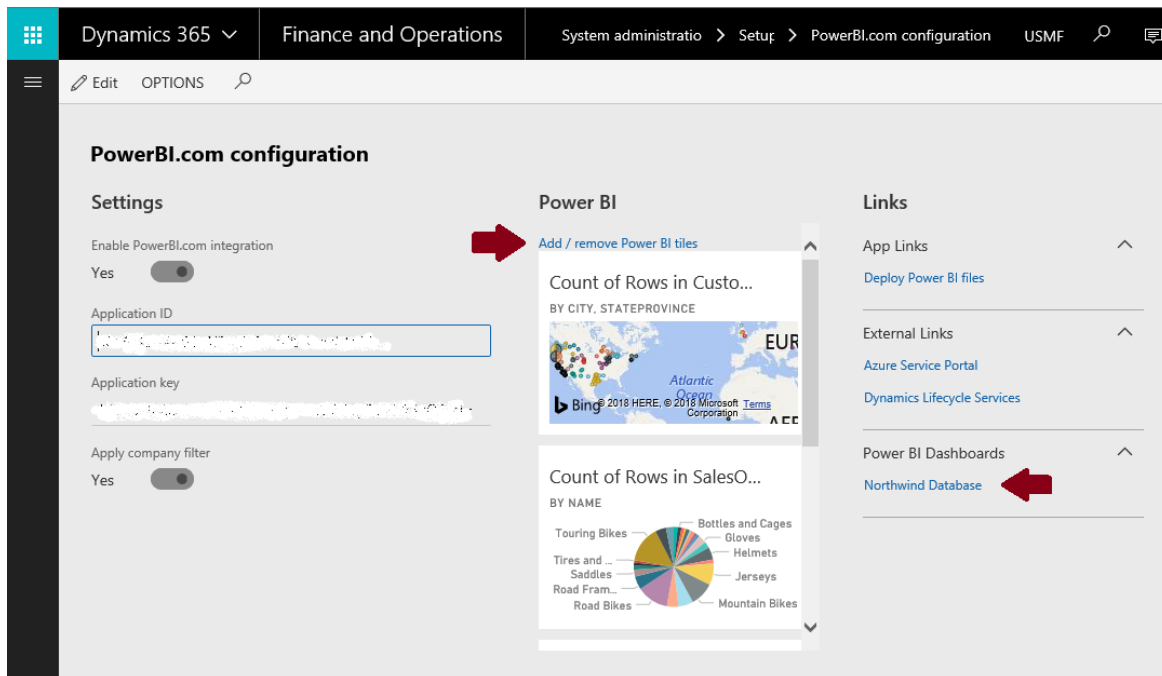


The screenshot shows the Dynamics 365 configuration interface for PowerBI.com. The breadcrumb trail is 'System administration > Setup > PowerBI.com configuration'. The main content area is divided into three sections: 'Settings', 'Power BI', and 'Links'. The 'Settings' section includes a toggle for 'Enable PowerBI.com integration' (set to 'Yes'), an 'Application ID' field, an 'Application key' field, and another toggle for 'Apply company filter' (set to 'Yes'). The 'Power BI' section features a blue modal window with a bar chart icon and the text: 'Connect your Power BI account for immediate access to your favorite reporting visualizations right here in Dynamics 365.' and a 'Get started' button. The 'Links' section on the right has two expandable sections: 'App Links' with a link to 'Deploy Power BI files', and 'External Links' with links to 'Azure Service Portal' and 'Dynamics Lifecycle Services'.

If you're starting Power BI from the application for the first time, you're prompted to authorize sign-in to Power BI from the client. Select **Click here to provide authorization to Power BI.**

Users must complete this step the first time they pin Power BI content.

2. The Azure AD consent page asks for your consent. User consent is required for the application to access PowerBI.com on behalf of the user. Select **Accept.**
3. Because you're already signed in to Azure AD, you don't have to enter your credentials again. A new tab appears, where you're prompted to authorize the connection between the application and Power BI. Authorize the connection, and then return to the original tab.
4. A list of tiles from your PowerBI.com account appears. Select one or more tiles to pin to the selected workspace.



## Troubleshooting common errors

In the procedure above, after you click **Accept**, you might receive the following error message if the process is unsuccessful. Note that the details of the error appear at the bottom of the message. Additional technical information provides clues that can help you determine what went wrong.

### Some common issues and the resolution steps

ERROR	RESOLUTION
The Power BI service is unavailable.	This issue doesn't occur very often, but the Power BI service might sometimes be unreachable. You don't have to re-register. Try to pin a tile to a workspace later.
You can't access the application.	You probably didn't select all the check boxes under <b>Step 8 in the Registration process</b> during the registration process. Start Power BI, and re-run the registration process.
The Power BI tiles page is empty (no content is shown).	Your PowerBI.com account might not have a dashboard or any tiles. Add a dashboard, such as a sample dashboard, and try to pin a tile again.
Error when authorizing Power BI	On the Azure Admin dashboard, under <b>Users and Groups &gt; User settings</b> , make sure that the <b>Users can consent to apps accessing company data on their behalf</b> option is set to <b>Yes</b> .
Sorry, something went wrong. The authentication process was not successful. Please contact your system administrator.	This may occur in cases where service keys have expired. We recommend performing the registration process above, starting with step 3. When complete, be sure to update the PowerBI.com settings at <b>AX client &gt; System administration &gt; Set up &gt; PowerBI.com integration</b> .

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Bring your own database (BYOD)

2/18/2021 • 13 minutes to read • [Edit Online](#)

This topic explains how administrators can export data entities from the application into their own Microsoft Azure SQL database. This feature is also known as *bring your own database* (BYOD). The BYOD feature was released in Microsoft Dynamics AX with platform update 2 (August 2016). Minor improvements and bug fixes have been included in subsequent platform updates.

The BYOD feature lets administrators configure their own database, and then export one or more data entities that are available in the application into the database. (Currently, more than 1,700 data entities are available.) Specifically, this feature lets you complete these tasks:

- Define one or more SQL databases that you can export entity data into.
- Export either all the records (*full push*) or only the records that have changed or been deleted (*incremental push*).
- Use the rich scheduling capabilities of the batch framework to enable periodic exports.
- Access the entity database by using Transact-SQL (T-SQL), and even extend the database by adding more tables.

## Entity store or BYOD?

If you followed the series of [blog posts about Microsoft Power BI integration](#), you will be familiar with Entity store. Entity store is the operational data warehouse. Entity store provides built-in integration of operational reports with Power BI. Ready-made reports and analytical workspaces use Entity store. If you write Power BI reports by using data in your application environment, you should use Entity store.

However, the BYOD feature is recommended for the following scenarios:

- You must export data into your own data warehouse.
- You use analytical tools other than Power BI, and those tools require T-SQL access to data.
- You must perform batch integration with other systems.

### NOTE

The application doesn't allow T-SQL connections to the production database. If you're upgrading from a previous version of Finance and Operations, and you have integration solutions that require direct T-SQL access to the database, BYOD is the recommended upgrade path.

You can use either Entity store or BYOD. The default operational reports that are available take advantage of embedded Power BI and Entity store. We recommend that you use our default operational reports as your first choice. You can also extend the ready-made operational reports to meet your requirements. You should consider BYOD a complementary option that you use as you require.

## Creating a SQL database

Before you can configure the entity export option and use the BYOD feature, you must create a SQL database by using Azure portal.

For one-box development environments, you can create a database in the local Microsoft SQL Server database. However, this database should be used only for development and testing purposes. For production



environments, you must create an Azure SQL database.

You should also create a SQL user account for sign-in to the database. Write down the server name, database name, and the SQL user ID and password. You will use this information when you configure the entity export option in the next section.

If you're using the BYOD feature for integration with a business intelligence (BI) tool, you should consider using clustered columnstore indexes (CCIs). CCIs are in-memory indexes that improve the performance of read queries that are typical in analytical and reporting workloads.

#### NOTE

Your BYOD database must be accessible to Finance and Operations apps. If you encounter issues where you are unable access to access BYOD, you must ensure firewall rules in your BYOD are configured appropriately.

## Configuring the entity export option

1. Start the client, and then, in the **Data management** workspace, select the **Configure Entity export to database** tile.
2. If you've configured any databases, a list is shown. Otherwise, you must configure a new database. In this case, select **New**, and then enter a unique name and a description for the new database. Note that you can export entities into multiple databases.
3. Enter the connection string in the following format:

```
Data Source=<logical server name>,1433; Initial Catalog=<your DB name>; Integrated Security=False; User ID=<SQL user ID>; Password=<password>
```

In this connection string, the logical server name should resemble **nnnn.database.windows.net**. You should be able to find the logical server name in Azure portal. The following illustration shows an example of a connection string.

#### NOTE

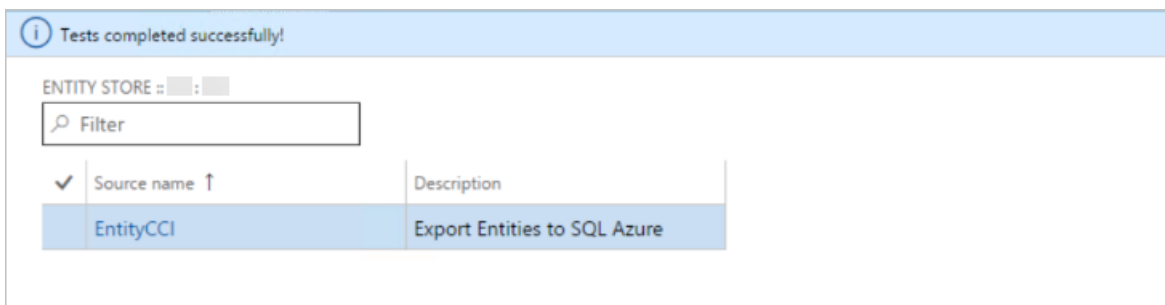
The default extension field shown in the image above does not apply to BYOD.

4. Select **Validate**, and make sure that the connection is successful.

- The **Create clustered column store indexes** option optimizes the destination database for selected queries by defining CCI for entities that are copied. However, CCI are currently supported only on SQL premium databases. Therefore, to enable this option, you must create a SQL premium database.
- The **Enable triggers in target database** option sets export jobs to enable SQL triggers in the target database. This option lets you hook downstream processes into the trigger to orchestrate actions that must be started after records have been inserted. One trigger is supported per bulk insert operation. The size of the bulk insert is determined by the **Maximum insert commit size** parameter in the Data management framework.

For scenarios in which reporting systems read data from BYOD, there is always the challenge of ensuring that the reporting systems get consistent data from BYOD while the sync is in progress. You can achieve this result by not having the reporting systems read directly from the staging tables created by the BYOD process. The staging tables hold the data while data is being synced from the instance and hence will be constantly changing. Use the SQL trigger feature to determine when the data sync has been completed, and then hydrate the downstream reporting systems.

When the validation is passed, the database that you configured for entity export appears in lists of databases, as shown in the following illustration.



You can now publish one or more entities to the new database by selecting the **Publish** option on the menu.

### **Publishing the entity schema to the database**

The **Publish** page enables several scenarios:

- Publish new entities to the database.
- Delete previously published entities from the database. (For example, you might want to re-create the schema.)
- Compare published entities with the entity schema. (For example, if new fields are added later, you can compare the fields with your database schema.)
- Configure change tracking functionality that enables incremental updates of your data.

The following sections discuss each option.

#### **Publish**

The **Publish** option defines the entity database schema on the destination database. When you select one or more entities, and then select the **Publish** option, a batch job is started. This job creates the entities in the destination database. When the database definition job is completed, you receive a message, which you can access by using the bell symbol in the upper right.

The actual data update occurs when you export data. At this point, you're just creating the schema.

#### **Drop entity**

The **Drop entity** option deletes the data and the entity definition from the destination database.

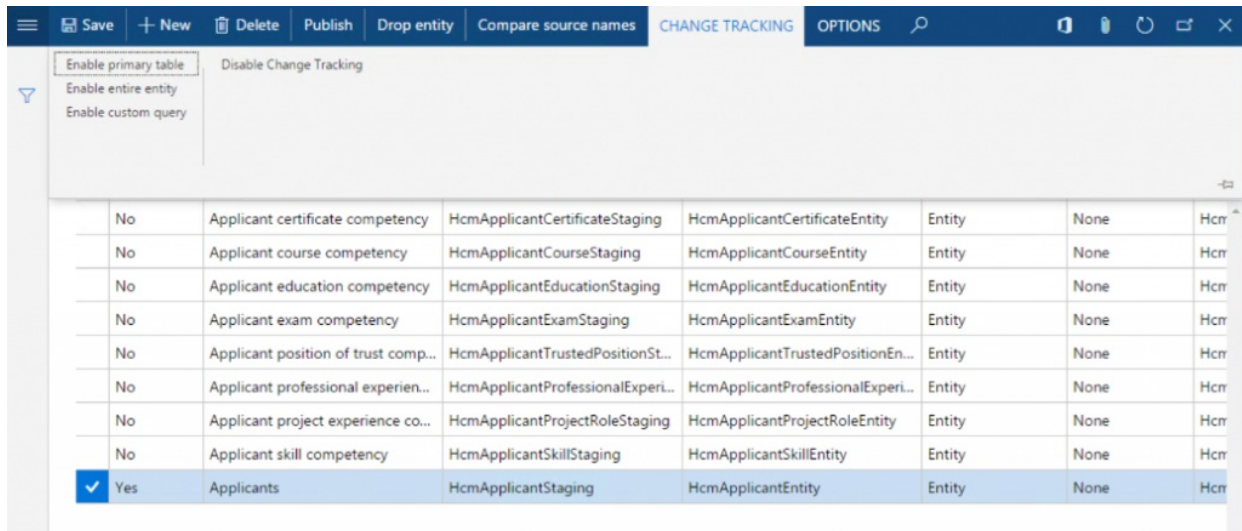
#### **Compare source names**

The **Compare source names** option lets you compare the entity schema in the destination with the entity schema in the application. This option is used for version management. You can also use this option to remove any unwanted columns from the destination table.

## Configure change tracking

Change tracking is a feature that is provided in SQL Server and SQL Database. Change tracking enables the database to track changes including deletes that are made on tables. The system uses change tracking to identify changes that are made to tables as transactions. However, because the application must track changes at the data entity level, there is additional logic on top of SQL change tracking to make this functionality work. The steps to enable change tracking are explained later in this section.

The **Change tracking** option on the **Publish** page lets you configure how changes are tracked on the underlying entity.



Change Tracking	Entity Name	Staging Table	Entity Table	Type	Other	Other
No	Applicant certificate competency	HcmApplicantCertificateStaging	HcmApplicantCertificateEntity	Entity	None	Hcr
No	Applicant course competency	HcmApplicantCourseStaging	HcmApplicantCourseEntity	Entity	None	Hcr
No	Applicant education competency	HcmApplicantEducationStaging	HcmApplicantEducationEntity	Entity	None	Hcr
No	Applicant exam competency	HcmApplicantExamStaging	HcmApplicantExamEntity	Entity	None	Hcr
No	Applicant position of trust comp...	HcmApplicantTrustedPositionSt...	HcmApplicantTrustedPositionEn...	Entity	None	Hcr
No	Applicant professional experien...	HcmApplicantProfessionalExperi...	HcmApplicantProfessionalExperi...	Entity	None	Hcr
No	Applicant project experience co...	HcmApplicantProjectRoleStaging	HcmApplicantProjectRoleEntity	Entity	None	Hcr
No	Applicant skill competency	HcmApplicantSkillStaging	HcmApplicantSkillEntity	Entity	None	Hcr
<input checked="" type="checkbox"/>	Applicants	HcmApplicantStaging	HcmApplicantEntity	Entity	None	Hcr

The following table describes the change tracking options that are available.

OPTION	DESCRIPTION
Enable primary table	An entity consists of several tables. Select this option to track all changes that are made to the primary table of the entity. When changes are made to the primary table, the corresponding record is inserted into or updated in the destination database. Although data from the whole entity is written to the destination table, the system triggers the insert or update option only when the primary table is modified.
Enable entire entity	Select this option to track all changes to the entity. (These changes include changes to all the tables that make up the entity.) When changes are made to the entity, corresponding updates are made to the destination.
Enable custom query	This option lets a developer provide a custom query that the system runs to evaluate changes. This option is useful when you have a complex requirement to track changes from only a selected set of fields. You can also select this option when the entities that will be exported were built by using a hierarchy of nested views. For more information, see <a href="#">Enable change tracking for entities</a> .

To use change tracking, you must enable the **Change tracking** option as shown above in data management. This action is available on the **Data entities** list page, by going to **Data management > Data entities**. You need to select an entity and select from one of the options listed above to enable change tracking on the data entity.

If you republish an entity that exists in the destination database, the system warns you that existing data will be

deleted because of the new operation.

When you confirm the publish operation, the system publishes the schema to the database, and you're notified when the operation is completed.

By selecting the **Show published only** option on the **Publish** page, you can show only the entities that were published to a given destination database. The Publish function creates the entity schema in the database. You can navigate to the database and see the table schemas that were created, together with corresponding indexes.

#### NOTE

Currently, you can't use BYOD to export composite entities into a database. You must export each entity in the composite entity.

## Exporting data into your database

After entities are published to the destination database, you can use the Export function in the **Data management** workspace to move data. The Export function lets you define a Data movement job that contains one or more entities.

You can use the **Export** page to export data into many target data formats, such as a comma-separated values (CSV) file. This page also supports SQL databases as another destination.

### Export

The screenshot shows the 'Export' configuration page. On the left, under 'JOB DETAILS', there are several fields: 'Name' (OneTimeExport), 'Target data format' (EntityCCI), 'Entity name' (empty), 'Use sample file' (No), 'Skip staging' (Yes), 'Default refresh type' (Incremental push only), and 'Generate data package' (No). On the right, under 'SELECTED FILES AND ENTITIES', there is a search filter, a trash icon, and up/down arrows. A blue card for 'Applicants' is selected, with a checkmark and a 'View map' button. An 'Add entity' button is located at the bottom right of the configuration area.

You can create a data project that has multiple entities. You can schedule this data project to run by using the batch framework. You also schedule the data export job to run on a periodic basis by selecting the **Export in batch** option.

The same job can also be used to export data from all companies. In prior to Platform update 27, this feature can be enabled by enabling the flight DMFEnableAllCompanyExport as explained in [Data management overview](#). Starting in Platform update 27, this feature can be enabled in data management framework parameters. After the feature is enabled, a new option will appear when adding an entity to a data project. This option can be enabled to export data from all companies for the specific entity.

#### NOTE

Adding multiple entities to an export project for BYOD must be done carefully to ensure the overall reliability of the BYOD export is not compromised. Different parameters must be taken into consideration when deciding the number of entities that are added to the same project. Some of these parameters should be the degree of complexity of the entities, data volume per entity that is expected, and the overall time for export to complete at the job level. Adding hundreds of entities must be avoided, therefore creating multiple jobs with smaller number of entities is recommended.

Use of recurring exports in **Manage > Manage recurring data jobs** for BYOD is discouraged. You must use the **Export in batch** option.

### Incremental export

When you add an entity for data export, you can select to do an incremental export (which is also known as incremental push) or a full push. For incremental push to work, you must enable the **Change tracking** option in the database and specify an appropriate change tracking option, as described earlier in this topic.

#### NOTE

A full push deletes all existing records from an entity and then inserts the current set of records from the selected entity.

If you select an incremental push, the first push is always going to be a full push. This is because SQL needs to know which records have been 'tracked' in order to be able to track subsequent changes. Whenever a new record is inserted, or a record is added or deleted, the corresponding change will be reflected in the destination entity.

Because the first push is always a full push, we do not recommend that you do an explicit full push before you enable change tracking.

We recommend that you first enable change tracking and schedule a export job with incremental push. This will take care of the first full push and the subsequent incremental exports.

### Timeouts

The default timeouts for BYOD exports are set to ten minutes for truncation operations and one hour for actual bulk insert operations. When volumes are high, these timeout settings may not be sufficient and must be updated. Starting with the release of Platform update 18, you can update the timeout settings by navigating to **Data management > Framework parameters > Bring your own database**. These timeouts are company specific and must be set separately for each company.

### Known limitations

The BYOD feature has the following limitations.

#### **There should be no active locks on your database during synchronization**

Because BYOD is your own database, you must ensure that there are no active locks on your Azure SQL database when data is being synced. Having active locks on your database during synchronization can result in slow writes or even failure to export to your Azure SQL database.

#### **You can't export composite entities into your own database**

Currently, composite entities aren't supported. You must export individual entities that make up the composite entity. However, you can export both the entities in the same data project.

#### **Entities that don't have unique keys can't be exported by using incremental push**

You might face this limitation especially when you try to incrementally export records from a few ready-made entities. Because these entities were designed to enable the import of data, they don't have a unique key. However, you can enable change tracking only for entities that have a unique key. Therefore, there is a limitation on incremental push. One workaround is to extend the required entity and define a unique key.

# Troubleshooting

## Incremental push not working correctly

**Issue** - When a full push occurs for some entity then a large set of records can be seen in BYOD using a select statement. However, an incremental push results in only a few records in BYOD. It seems as if the incremental push deleted all the records and added only the changed records in BYOD.

**Solution** - In cases like this it is recommended to disable and re-enable change tracking for the entity in question. The state of the SQL change tracking tables might not be in the expected state. Also verify that there are no other incremental exports that cover the same tables (DMF, MR, Retail).

## SSIS Error Code DTS\_E\_OLEDBERROR. An OLE DB error has occurred. Error code: 0x80004005

**Issue** - Export to BYOD fails with an SSIS exception shown below.

```
An OLE DB error has occurred. Error code: 0x80004005.
```

```
An OLE DB record is available. Source: "Microsoft SQL Server Native Client 11.0" Hresult: 0x80004005  
Description: "Communication link failure".
```

```
An OLE DB record is available. Source: "Microsoft SQL Server Native Client 11.0" Hresult: 0x80004005  
Description: "TCP Provider: An existing connection was forcibly closed by the remote host."
```

```
Failed to open a fastload rowset for <entityStaging>. Check that the object exists in the database.
```

```
OLE DB Destination failed the pre-execute phase and returned error code 0xC0202040.
```

**Solution** - This can occur if the connection policy on the Azure SQL BYOD server is set to Proxy. This must be changed to 'Redirect' as explained in [SQL DB Connectivity Architecture](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Preview PDF documents using a PDF viewer

2/18/2021 • 2 minutes to read • [Edit Online](#)

Streamline application experiences that result in the production of business documents by taking advantage of the embedded PDF Preview option. Finance and Operations applications deliver a modern experience to preview business documents that are produced by the service. You can use the built-in toolbar to navigate and download the document or to print to locally connected devices.

The embedded viewer offers consistency between the screen presentation and the printed output. In addition, report viewing times are drastically reduced when compared to the legacy experience. The Preview option is available on all supported devices and does not require any additional third-party software. Documents can be easily downloaded and navigated by using the built-in viewer toolbar options.

The following illustration shows a preview of the experience with a modern business document.

The screenshot shows a browser window titled "Finance and Operations Preview" displaying a PDF document. The document is a sales order confirmation for Contoso Entertainment System USA. It includes a company logo, address, shipping information, a confirmation number (000783-1), date (17 August 2017), and a total amount of \$110.00. A table lists items: Professional speaker cable. An orange callout box states: "PDF formatted document is presented to the user within hosted viewer control".

ITEM	DESCRIPTION	QUANTITY	SALES PRICE	DISCOUNT	AMOUNT
M0030	Professional speaker cable	3 ea	20.00	0.00	60.00
M0030	Professional speaker cable	2 ea	25.00	0.00	50.00
NET AMOUNT					110.00
SALES TAX					0.00
<b>USD TOTAL</b>					<b>\$110.00</b>

The legacy HTML-based preview experience is being replaced by a true document preview experience. There are several key advantages in the modern PDF preview experience. These advantages include:

- A fidelity between the screen presentation and the printed output.
- A consistent document report preview experience across devices and platforms, including on-premises deployments.
- The server-side rendering improves the performance when producing the document.
- A built-in tooling that allows users to quickly navigate the contents of the business document.

## Accessing the PDF preview experience (Platform update 36 or later)

The PDF preview experience is enabled by default in [Self-Service deployments](#) and in environments hosted on Platform update 39 or later. To use the PDF preview experience in cloud-hosted and Microsoft-managed environments running Platform updates 36 thru 38, use Feature Management to enable the **Report PDF viewer** feature.

## Additional feature information

- Expandable/collapsible sections are available by default. These interactive operations do not function after the PDF document has been created.
- The printer drop-down menu allows users to choose from locally connected devices. This list does not include network printers connected through the service.
- Documents are downloaded to the local device using the built-in toolbar actions.
- Use the **Print destination** options to produce documents in formats other than PDF.

## Feature limitations

The Embedded PDF viewer experience delivers a closed document that exactly matches the printed output of the document. These documents cannot be modified by the recipient making the format ideal for business operations. However, as a closed format, the documents are far less interactive on the screen when compared to HTML presentations. The following end-user capabilities are not supported when previewing documents using the embedded PDF viewer.

- By default, embedded drill-thru navigation links are only available while previewing PDF documents.
- PDF documents do not support expandable and collapsible sections.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Analytics, aggregate measurements, and KPI modeling

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic discusses the embedded business intelligence (BI), aggregate measurements, dimensions, and data entities, and aggregate programming model.

## Embedded business intelligence

The term embedded business intelligence (BI) refers to experiences that use highly intuitive and fluid visualizations to provide insights that are relevant to a task, so that user is more informed and can make better choices. For example, a rental clerk in a car rental organization views the previous rental preferences of a customer when the customer makes a reservation. In this case, the clerk can see the vehicles and colors that the customer has selected in the past, and therefore can offer options that are likely to please the customer. Embedded BI is used throughout the user interface. At a technical level, building rich visualizations requires a powerful charting framework, and also an efficient way to access aggregated data that enables the display of fluid visualizations. Microsoft Dynamics 365 Finance and Operations meets both of these requirements, so that application developers can build rich and deep embedded BI-enabled scenarios.

## Where are the perspectives?

Perspectives were a concept that was designed to present data from a reporting point of view. Perspectives have evolved over the past three releases for analytical scenarios.

VERSION	DESCRIPTION
Microsoft Dynamics AX 4.0	Perspectives provided the ability to model ad-hoc reporting models.
Microsoft Dynamics AX 2009	Perspectives added support for modeling analysis cubes.
Microsoft Dynamics AX 2012	Perspective modeling capabilities were improved through richer modeling support and deeper integration with the Application Object Tree (AOT).
Finance and Operations apps	Perspectives are a first-class citizen in the data access framework. They can be consumed via X++ or C# code, and also in a model-driven way.

Perspectives reside within the analytics collection in the Application Explorer. Perspectives have undergone a major upgrade and now incorporate the following improvements:

- You can model new aggregate models and customize existing aggregate models as a star schema within Application Explorer.
- Modeling for key performance indicators (KPIs) in Application Explorer is supported.
- You can model data entities by referencing aggregate models. Data entities can be exposed to external reporting tools, such as PowerBI, as OData endpoints. Data entities can also be consumed.
- You can consume aggregate models directly in the programming model by using X++ or C# code. You no longer have to write MDX code to consume aggregate data.

- Aggregate data is a first-class citizen within application data access. Its behavior is similar to the behavior of detailed data. For example, aggregate data can be enriched with extended data types (EDTs) and enumerations, and you can help secure them by using application security concepts.
- The aggregate data infrastructure is maintained completely within the environment. By default, aggregate measurements are real-time. As a system administrator, you can manage the latency of aggregate data and controls based on available resources and business needs, without having to deal with the complexity of scheduling and external tools.
- Developers can reuse existing business models, making the modeling process quicker and easier.

Projects that were generated by using perspectives from Dynamics AX 2012 and later can be upgraded to Finance and Operations metadata equivalents.

## Aggregate measurements and aggregate dimensions

An aggregate measurement is a model that contains a collection of measures together with their corresponding dimensions. Measures are aggregate numbers, such as Total Sales or Number of Orders. Dimensions are slicers, such as Product, Vendor, or Customer, that help you analyze the measure. For example, the measure of Total Sales isn't useful unless it can be sliced by Product, Region, and Customer. Aggregate measurements are the evolution of AX 2012 analysis cubes. Whereas a cube was based on a multidimensional online analytical processing (OLAP) technology, an aggregate measurement abstracts the underlying technology. Therefore, you no longer have to know a lot about the underlying implementation technology. Additionally, the underlying technology infrastructure can take advantage of improvements in in-memory real-time technology without requiring the developer to rewrite the program. Aggregate dimensions are shared across an implementation. Aggregate measurements associate themselves with relevant aggregate dimensions. For example, Total Sales can be associated with Customer, Product, and Sales Region dimensions. However, Total Sales can't be associated with Vendor and Warehouse dimensions. Aggregate dimensions and aggregate measurements are modeled by using Visual Studio tools. The Upgrade tool lets customers and partners migrate existing Dynamics AX 2012 cubes to newer versions of the product.

## Aggregate data entities

By using the model-driven approach, you can create data entities by directly referencing aggregate measurements and aggregate dimensions. These are known as aggregate data entities. Aggregate data entities are read-only data entities that are used for reporting purposes. To consume aggregate data when you build charts and other client controls, add the aggregate data to a form as a data source. You can also consume aggregate data entities programmatically in C# or X++ code.

## Aggregate programming model

The Aggregate programming model lets a developer consume aggregate data programmatically by using either X++ or C# code. Data that you retrieve by using the Aggregate programming model can be used as a data source in forms and reports. A developer can add aggregate data that is modeled in perspectives to an **AXQuery** object. The developer can also use an existing aggregate data entity to create a query that can be extended by adding filters and additional columns that aren't present in the aggregate data entity. Bulk Move is a capability that is associated with the Aggregate programming model. When a query is run by the kernel, the developer can move all the records to a temporary or regular table without iterating row by row. Bulk Move provides a very efficient way to move data from aggregate models to temporary tables.

## Method expressions

Method expressions are a programming model for constructing rich calculations that are used to define fields in a data entity. Method expressions enhance the computed column capability that was introduced in AX 2012 views. Method expressions let you build expressions by using the C# or X++ programming language. You can

also create calculations on aggregate data that was previously coded by using MDX. Method expressions can be shared across the program.

## In-memory, near-real-time aggregate measurements

Aggregate measurements are deployed to Microsoft SQL Server non-clustered column store indexes (NCCI). Therefore, they can take advantage of the in-memory computing (IMC) engine that is built into Microsoft SQL Server 2016 as Azure DB. Aggregate measurements that have the IMC engine as their destination are referred to as in-memory, near-real-time (IM-NRT) aggregate measurements. These aggregate measurements don't require that a Microsoft SQL Server Analysis Services (SSAS) server be used. Because these models don't involve data updates, the queries that are sent to them reflect the latest state of data in the operational database. That is why it's referred to as near-real-time.

## KPI modeling and customization

In AX 2012 and earlier versions, KPIs and business indicators had to be modeled by using native SQL Server development tools. Although users could pin a KPI or business indicator to a Role Center by using the **Business Overview** Web Part, they could not modify a KPI definition, such as the goal. Users can use a rich client form to modify a KPI definition that was built and shipped by a developer. Users can also define new KPIs by using the aggregate data that is contained in aggregate measurements. A developer can model a KPI definition in Microsoft Visual Studio and ship it to a customer, either as a project or together with an independent software vendor (ISV) solution. After a KPI is defined, users can customize it at run time.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Model aggregate data

2/18/2021 • 15 minutes to read • [Edit Online](#)

This tutorial will walk you through the process of modeling aggregate data.

## Prerequisites

This tutorial requires you to access the environment using Remote Desktop, and be provisioned as an administrator. For more information, see the topic named [Deploy and access development environments](#).

## Key concepts

- **Aggregate measurements**, similar to **perspectives** and **Cubes** from earlier versions, enable you to model and consume aggregate data.
- **Key Performance Indicators (KPIs)** are a form of analytic controls that track organizational performance against the current status. KPIs are represented as tiles in a **workspace** or the **dashboard**. In this tutorial, you will model a KPI in Visual Studio.
- KPIs that are modeled in Visual Studio can be modified in the client. A user also has the ability to model new KPIs using the client.
- A **workspace** is an overview page that is specific to a particular subject area. Workspaces are common to all users.
- The **dashboard** is the default home page for each user.
- **Tiles** are securable objects that can be pinned to a workspace or the dashboard. KPIs and aggregate data that are shown on the dashboard, or a workspace, can be secured by using menu items.
- Aggregate data can be consumed in building charts and other controls. Using the model driven approach, you have the ability to create data entities by directly referencing aggregate measurements and aggregate dimensions. These entities are referred to as **Aggregate data entities**. Aggregate data entities are read-only data entities used for reporting purposes.
- **The aggregate programming model** enables a developer to consume aggregate data programmatically using either X++ or C# code. Data retrieved using the aggregate programming model can be used as a data source in forms.
- **Method expressions** enable a developer to build rich expressions using aggregate data. Method expressions are an X++ class. KPIs can be modeled using method expressions, thereby eliminating the need to build MDX expressions.
- **Contextual BI** refers to providing required insights as part of the user experience such that the user has relevant insights to not only achieve the task at hand, but be highly-productive during the course of the day.
- **Embedded BI** refers to analytic content being embedded within the user experience. Contextual BI and embedded BI teams are closely related. Contextual BI implies the added notion that the context of analytic context revolves around the data or the task.
- **Self-service BI** refers to enabling a user to tweak existing and/or create new analytic content such as reports, KPIs, and dashboards.

## Set up

If this is the first tutorial you are working on, make sure you have configured the administrator user if you are running on a local VM.

### Import the tutorial project

If you have already imported the Fleet management tutorial project, skip to the next section. In Visual Studio, on the **Dynamics 365** menu, click **Import Project**.

1. Download the Fleet Management sample from <https://github.com/Microsoft/FMLab>, save it to c:\, and unzip it.
2. In the **Import Project** window, next to the **Filename** text box, click the ellipsis button.
3. In the **Select the file to import** window, browse to the location of the **FMLab** folder, click **FMTutorialDataModel.axpp**, and then click **Open**.
4. In the **Project file location** text box, enter **C:\FMLab**.
5. Select the **Overwrite Elements** check box, and then click **OK**.

### Open the tutorial project

1. In Visual Studio, open the **FMTutorial** project. On the **File** menu, point to **Open**, and then click **Project/Solution**.
2. In the **Open Project** dialog box, browse to **C:\FMLab\FMTutorial**, and then click **FMTutorial**. Click **Open**. The **FMTutorial** project appears in **Solution Explorer**.
3. Use the **FMTDataHelper** class to load data for the Fleet Management tutorial. In **Solution Explorer**, in the **FMTutorial** project, expand **Classes**. Right-click **FMTDataHelper**, and then click **Set as Startup Object**.
4. On the **Build** menu, click **Rebuild Solution**. You use the rebuild to update the timestamps of the imported artifacts. You can view the build progress in the **Output** window.
5. Press **Ctrl+F5** to run the project and load the data.

## Model an aggregate measurement for rental charges

Often, when a user asks for additional information, you get a request for one or more new reports. Imagine that the manager of a rental car company has called and asked for a report. The manager is interested in finding out how the rental business is performing. The manager wants a report that shows rental revenue by month. You soon find out that the manager is interested in a breakdown of rental revenues. The manager wants to know whether the rental revenue is high in cases where they have sold additional services, for example, car seats, GPS, re-fueling, as opposed to the base rental charge. As it turns out, the manager suspects that specific customer groups are driving revenue up, and this is why the manager wanted the report in the first place. The manager insists on adding Customer group to the report. Because the revenue must be considered in relation to the number of rentals, the manager doesn't want a few large corporate rentals to skew her analysis. You both agree that the number of rentals needs to be shown along with revenue. We could represent this requirement as a set of business questions using a matrix. Rows indicate the **measures** (or numbers) and the columns indicate the **dimensions** (or slicers). An "X" in the intersection between a measure and a dimension indicates that the measure needs to be "grouped by" the dimension.

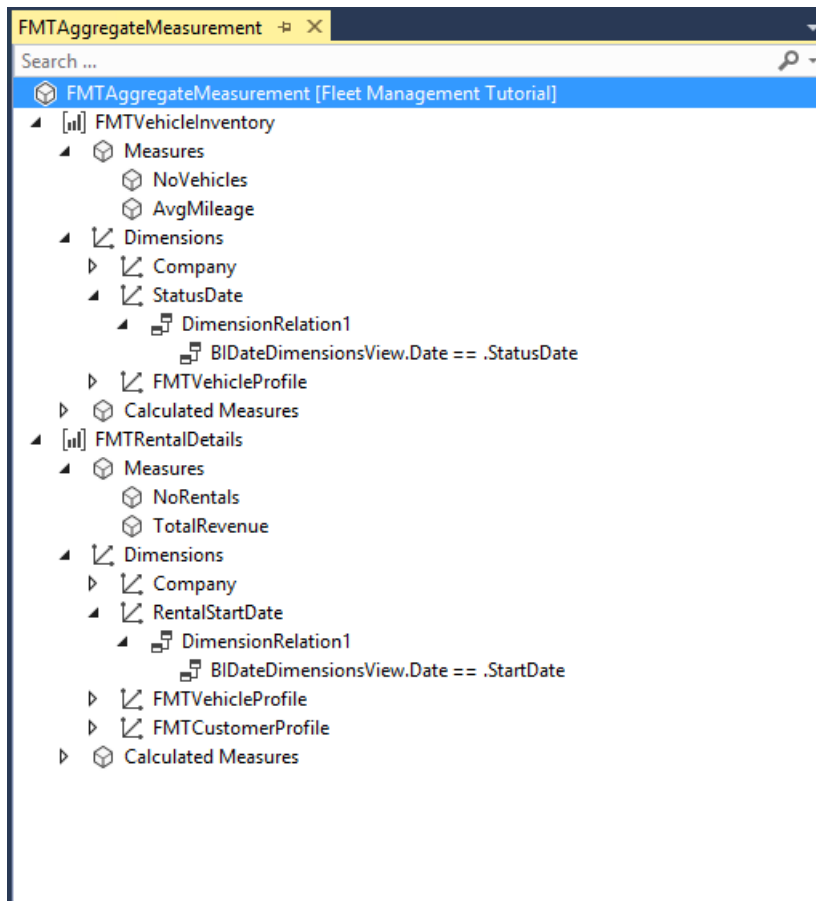
	RENTAL DATE	CUSTOMER GROUP	RENTAL CHARGE TYPE
Revenue	X	X	X
Number of rentals	X	X	X

Next, we will model an aggregate measurement to answer this business question.

### Add a measure group for rental charges by using an existing view

In this section you will add a new measure group to an existing aggregate measurement.

1. In **Solution Explorer**, expand the **Analytics** folder of the project, and then double-click the aggregate measurement, **FMTAggregateMeasurement**. The aggregate measurement will be launched in the designer. Notice that the existing aggregate measurement contains two measure groups related to vehicle inventory and rental details. You will create a new measure group related to rental charges.



- In **Solution Explorer**, expand the **Views** folder of the project, and then select the **FMTRentalChargeExtendedView** view.
- Drag-and-drop the **FMTRentalChargeExtendedView** into the root node of the **FMTAggregateMeasurement** aggregate measurement in the designer. Notice that a new measure group is created and the values of properties have been applied as follows.

PROPERTY	VALUE
Name	FMRentalChargeExtendedView
Table	FMRentalChargeExtendedView

- In **Solution Explorer**, double-click the **FMRentalChargeExtendedView** view. When the designer form opens, expand the **Fields** node.
- Select the **ExtendedAmount** and **RentalID** fields, and then drag-and-drop the two fields onto the **Measures** node of the newly created **FMTAggregateMeasurement** measure group called **FMRentalChargeExtendedView**. As you drag the fields, hover your cursor over the **FMTAggregateMeasurement** tab to access the **Measures** node. By default, when you drag-and-drop the fields, the system assumes that you want counts of the measures. In this case, you need to modify default properties for the **ExtendedAmount** and **RentalId** measures as follows:

PROPERTY (EXTENDEDAMOUNT)	VALUE
Default Aggregate	<b>Sum</b>
Field	ExtendedAmount (no change required)
Name	Revenue

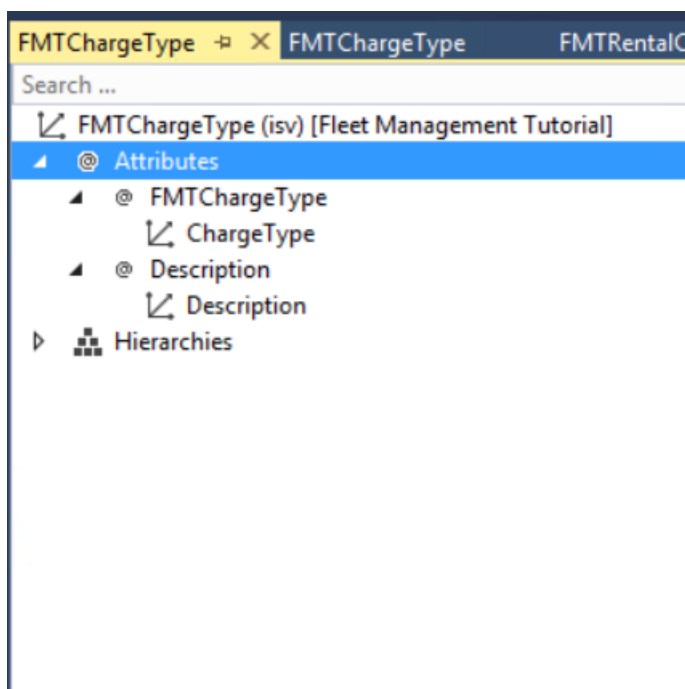
PROPERTY (RENTALID)	VALUE
Default Aggregate	DistinctCount
Field	RentalId (no change required)
Name	NumRentals

- Next let's create another measure which calculates **Revenue per Rental** by copying an existing measure. Select the **TotalRevenue** measure. Right-click and select **Copy** from the **Context** menu.
- Select the **FMRentalsDetails** measure group. Right-click and select paste. Rename the newly created measure to **RevenuePerRental**. Modify the aggregation function to **Average**.

### Model an aggregate dimension charge type

To analyze rental revenue by the different charge types, you need to be able to slice revenue by the charge type. For this purpose, you will first model a charge type dimension.

- In **Solution Explorer**, under **FMTutorial**, right-click **Analytics**, point to **Add**, and then click **New Item**.
- Select **Dynamics 365 Artifacts > Analytics > Aggregate Dimension** from the list of items.
- In the **Name** property, enter **FMTChargeType**. This is the name of the aggregate dimension that will be created. This name must be unique. Click **Add**. The new dimension will appear in Visual Studio.
- In **Application Explorer**, expand the **AOT** and click **Data Model > Tables**. Drag-and-drop the **FMTChargeType** table from **Application Explorer** onto the root node of the newly created **FMTChargeType** dimension in the designer. Notice that dimension attributes and corresponding keys have been added using the **AutoReport** field group of the table.
- Expand the **Attributes** node of the new dimension. Notice that several attributes have been created for you by default. The system has also created a dimension key based on unique indexes of the table. You can add additional fields by dragging and dropping them into the **Fields** node.



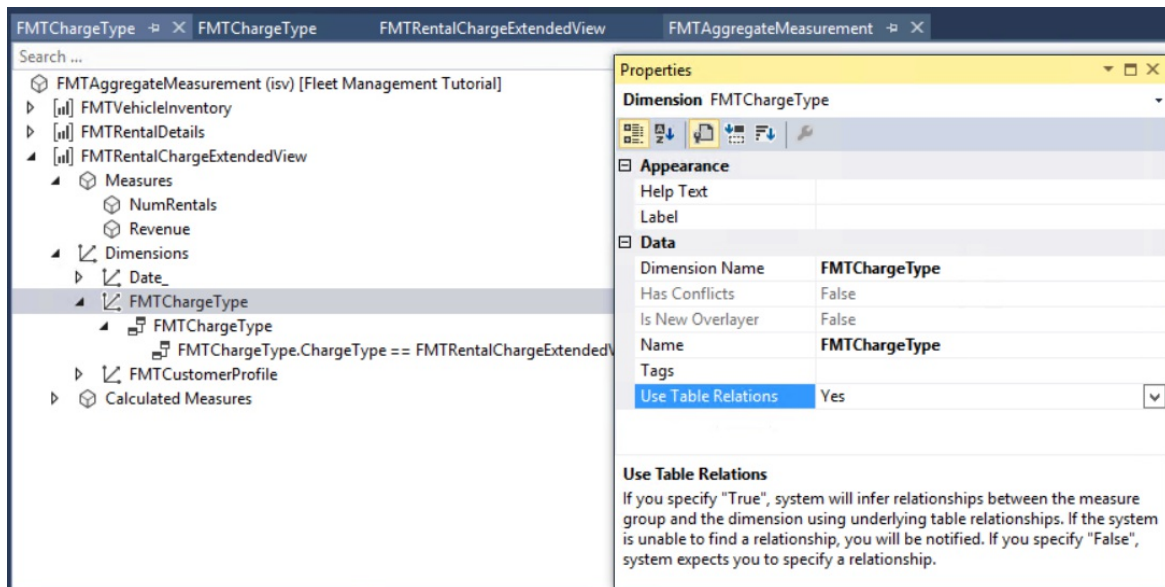
- Save the new dimension.
- You may get a warning asking you to rename the field name **Description** to avoid the MDX reserved word. Even though the aggregate dimension may not be deployed to SSAS in this aggregate

measurement, it's possible that this dimension may be used by an aggregate measurement deployed to SSAS in the future. To avoid potential issues in the future, rename the field name **Description** to **ChargeDescription**.

### Model dimension references for customer profile and charge type dimensions

Next, create dimension references to new and existing dimensions so that revenue can be sliced by customer as well as charge type.

1. In **Solution Explorer**, double-click **FMTAggregateMeasurement** or, if you have it open, navigate to it in the designer.
2. In **Solution Explorer**, select the dimensions **FMTChargeType** and **FMTCustomerProfile**.
3. Drag-and-drop them into the **Dimensions** node of the **FMTRentalChargeExtendedView** measure group. Notice that dimension references have been created along with relations.
4. Save changes to **FMTAggregateMeasurement**. Review the property sheet for the dimension relation and notice that the **Use Table relations** property is set to **Yes**. Notice that the drag-and-drop operation created relationships between the measure group dimensions **FMTRentalChargeExtendedView** and **FMTChargeType**, **FMTCustomerProfile**. Review the property sheet for the dimension relation and notice that the **Use Table relations** property is set to **Yes**.

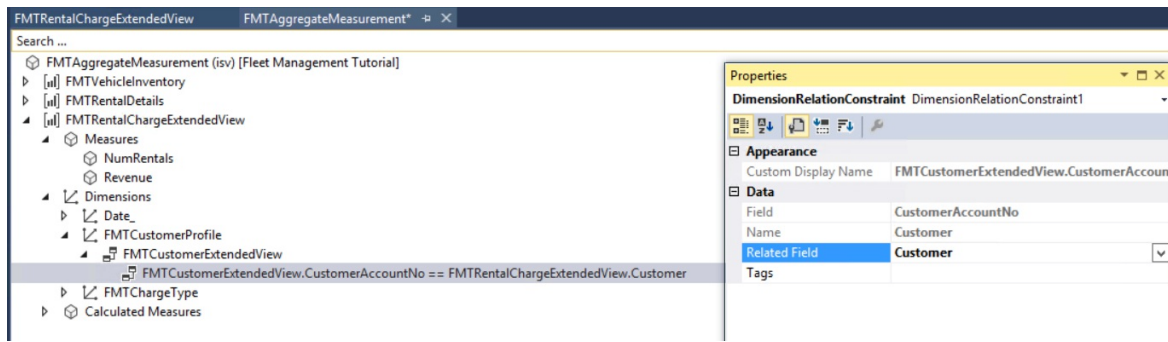


#### NOTE

In platform update 1611 and later, **UseTableRelations** property has been removed. When a new dimension reference is created, system will default existing relationships. You can continue to provide an explicit relationship by changing the relationship field that was defaulted. Providing an explicit relationship is equal to setting **UseTableRelationship** to **No**.

5. Expand the Dimension relations node for the **FMTCustomerProfile** dimension. Notice that the **UseTableRelations** property is set to **No**. In this case, the system has not been able to find a suitable relationship between the Measure group and dimension. You will need to specify one manually.
6. Expand the **FMTCustomerProfile** dimension reference if you have not done so already. Select the node **FMTCustomerExtendedView**. Right-click and see the property sheet.
7. Select **CustomerID** as the value for property **DimensionAttribute**. Select the relationship shown below. Select **Customer** for the property value **RelatedField**.
8. Save changes to **FMTAggregateMeasurement**.





- In this scenario, we specified a relationship because the system was unable to find one. You could also specify a different relationship if you want to override the system choice by setting **Use Table Relations** property to **No**.

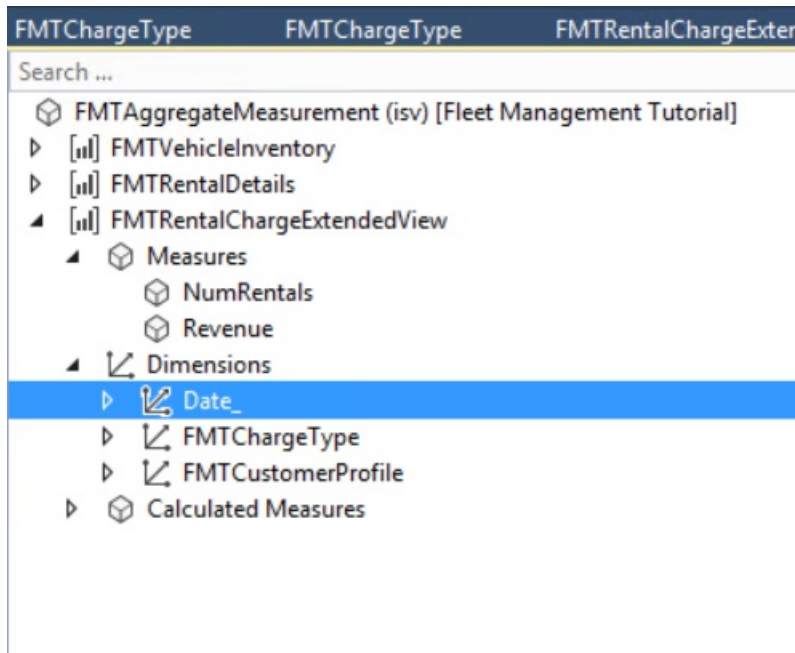
### Model dimension references for the date dimension's rental start date and transaction date

Assume that for analysis purposes, you want to enable slicing by the start date of the rental; but for accounting purposes, you want to enable slicing by the transaction date for each of the charges. To do this, you need to associate the rental charges measure group with two date dimensions. In the BI world, this pattern is known as **Role Playing dimensions**. By default, a date dimension is added to the measure group. You can rename the default name appropriately and add new date dimensions as required.

- Expand the **Dimensions** node of the **FMTRentalChargeExtendedView** measure group. Notice that the **Date\_** dimension has already been included as dimension slicers.

#### NOTE

If the table or the view used to model the measure group is a Company-specific table, for example it contains DATAAREAID as part of the key), a **Company** dimension relation will be created by default. In this case, the view we used is not a company specific one.

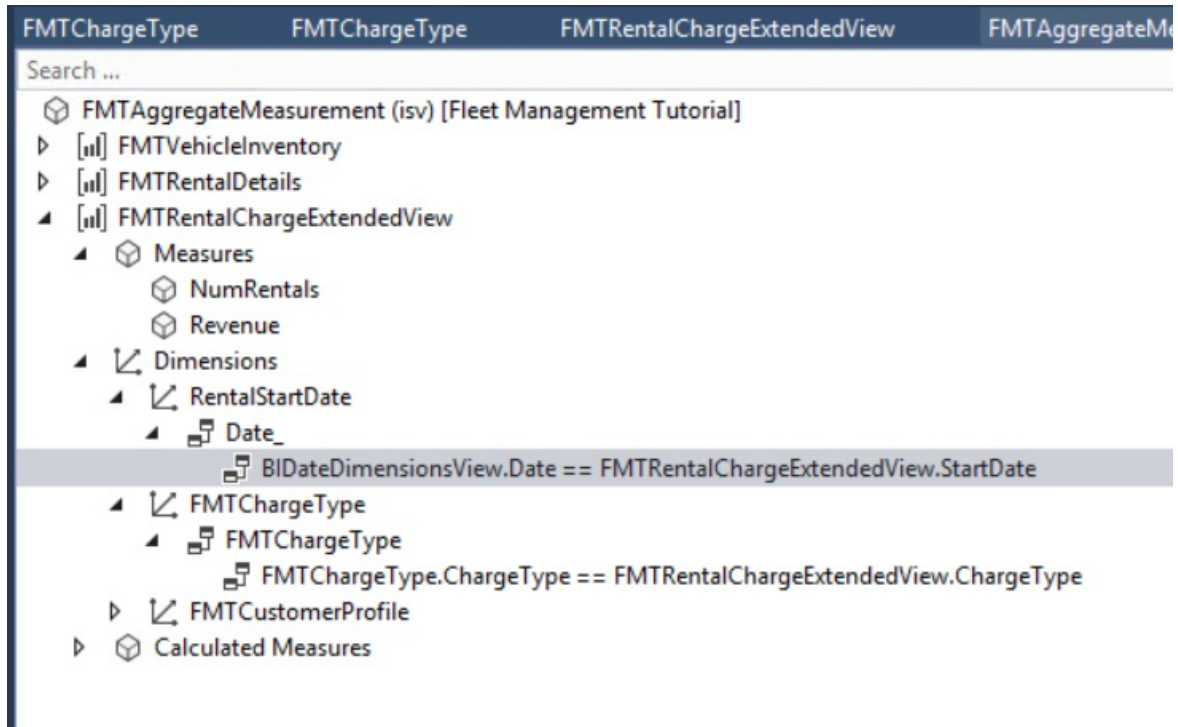


- Select the **Date** dimension and specify the following properties.

PROPERTY	VALUE
Name	RentalStartDate

PROPERTY	VALUE
Use Table Relations	No (This is the default – no need to change)

- Define the table relationship. Expand **RentalStartDate**, and then expand the **Date** node.
- Select the Relationship shown. Right-click and select the property sheet. Select **StartDate** for the value of **Related field** property.
- The relationship you defined should look like the following screenshot.



- Next, enable slicing of measures by the **TransactionDate** dimension. **TransactionDate** is also a date dimension, so you will add another reference to the date dimension and associate that with the corresponding field that contains the transaction date. When more than one date dimension is used as a slicer, each date dimension is known as a **Role playing date dimension**.
- Under the **FMTRentalChargeExtendedView** measure group, right-click the **Dimensions** node, and then click **New Dimension**. A new dimension will be added to the list of dimension references.
- Specify the following properties for the new dimension reference.

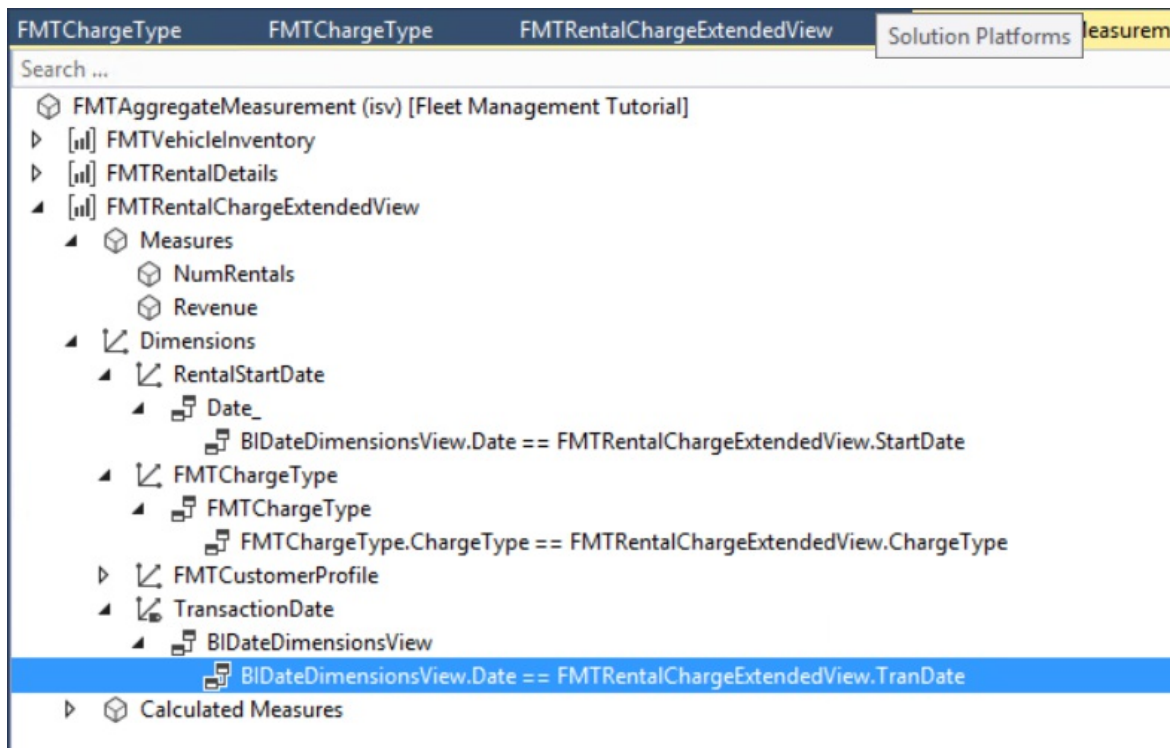
PROPERTY	VALUE
Dimension Name	Date_ <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"> <p>[!NOTE] If you select the Dimension Name in the wrong order, it will reset the other values you already set.</p> </div>
Name	TransactionDate
Use table relations	No
Tags	RolePlayingDate; Fleet

Notice the new property called **Tags**. This property enables the discovery of patterns within code and metadata from within the Visual studio environment. You can enter any number of tags and they can be searched using the hot keys or the **Dynamics 365** menu in Visual Studio.

9. Define the table relationship. Right-click **TransactionDate**, and then click **New Relation**. You do not need to specify any properties in the DimensionsRelation at this point.
10. Expand **BIDateDimensionValue**, and then select the **Relationship Constraint**. Right-click and select the property sheet.
11. Specify the following properties for the BIDateDimensionsView relation constraint.

PROPERTY	VALUE
Field	Date
Related Field	TranDate

The relationship you defined should look like the following screenshot.



12. Save the aggregate measurement.

### Deploy the newly generated aggregate measurement

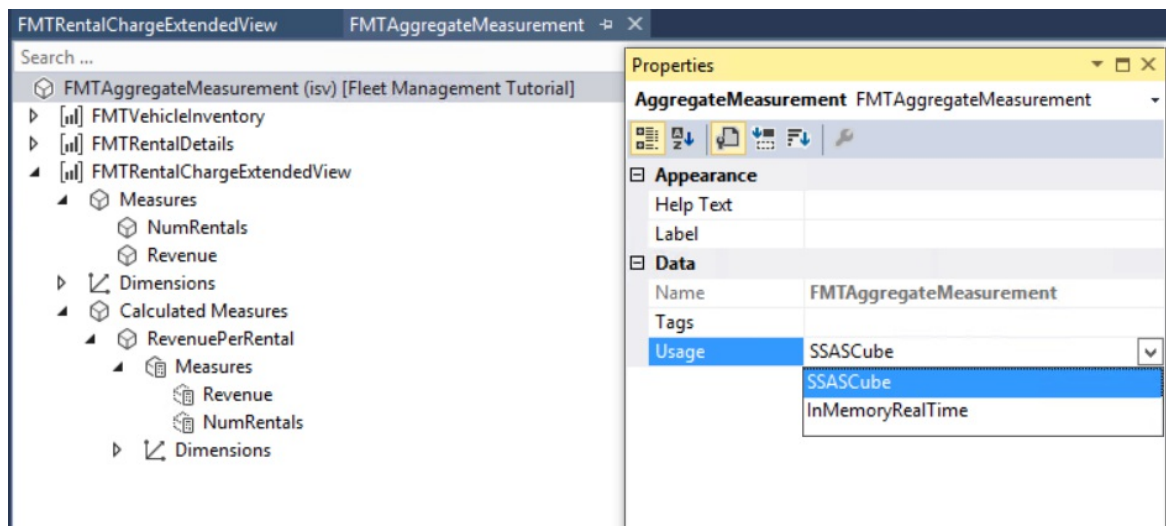
Now that you have completed modeling the aggregate measurement, you can deploy the aggregate measurement and continue with building KPIs and visualizations. You have 2 deployment choices as shown below.

OPTION	CONSIDERATIONS
In-memory real-time	This option will leverage the In-memory Column store indexes of SQL Server database to deploy Aggregate Measurements. This option is recommended when the Aggregate measurement is used for embedded analytics within the client where you need <b>real-time analytics</b> . For an overview of concepts on real-time analytics, see <a href="#">Analytics, aggregate measurements, and KPI modeling</a> .

OPTION	CONSIDERATIONS
Stage in Entity Store	This option leverages Entity store, the operational data store that enables <b>near real-time PowerBI reporting</b> . If you choose this option, Aggregate measurement can be deployed to Entity store and you can schedule the data to be refreshed periodically. For an overview of this approach, refer to the blog post here: <a href="https://blogs.msdn.microsoft.com/dynamicsaxbi/2016/06/09/power-bi-integration-with-entity-store-in-dynamics-ax-7-may-update/">https://blogs.msdn.microsoft.com/dynamicsaxbi/2016/06/09/power-bi-integration-with-entity-store-in-dynamics-ax-7-may-update/</a>

**NOTE**  
 SSAS Cube option is no longer supported when modeling aggregate measurements.

1. Select the **FMTAggregateMeasurement** node. Right-click and select **Properties**. Select **InMemoryRealTime** as the value for the property **Usage**.



2. InMemoryRealTime aggregate models are deployed to SQL Server using Non-Clustered Column Store Index (NCCI) technology. NCCIs is an in-memory technology that enables analytical and operational workloads to be served from SQL server database. NCCI indexes can be defined on tables similar to any other index. While NCCI indexes can be defined manually, framework has the ability to analyze index requirements and add them to underlying tables where necessary.
3. Right-click **FMAggregateMeasurement** in Solution Explorer, and then click **Add Column store indices option**. You will notice several new indexes being added by the system.
4. Save and build the project.
5. InMemoryRealTime aggregate models do not require data processing as the models are queried real-time. If you have not enabled database synchronization along with the build, manually synchronize the database.

## Model a KPI to show revenue per rental

### Model a KPI in Visual Studio

Model a KPI definition in Visual Studio by using the aggregate measurement you defined above.

1. In **Solution Explorer**, right-click **FMTutorial**, point to **Add**, and then click **New Item**.

- Select **Dynamics 365 Artifacts > Analytics > Key Performance Indicator**. Enter **FMTRevenuePerRental** as the name of the KPI, and then click **Add**. The name must be unique across KPIs. The KPI is created.
- Select **FMTRevenuePerRental**, and specify the **Measurement**. Leave the default values for the other properties.

PROPERTY	VALUE
Measurement	FMTAggregateMeasurement
Bad Threshold	0
Good Threshold	0
Scoring Pattern	MoreIsBetter
Show Goal	Yes
Show Status and Trend	Yes
Threshold Type	Value

- Define the expression for the KPI value. Under **FMTRevenuePerRental**, select **Value**, and specify the following properties.

**NOTE**

The values must be entered in the order they appear in the table:

PROPERTY	VALUE
Value Type	BasedOnMeasure
Measure Group	FMTRentalChargeExtendedView
Measure	RevenuePerRental

- Define the expression for the KPI Goal. Select **Goal**, and specify the following properties.

PROPERTY	VALUE
Goal Type	BasedOnValue
Value	250

**NOTE**

You could have defined a goal based on an aggregate measure as well. In this case, we will define a number as the goal.

- Save the KPI definition.

## Preview KPI in client

Next you will preview the KPI definition in the client.

1. Right-click **FMTutorial**, and then click **Re-Build**. On completion of the build, select the **Synchronize ... database** option.
2. Open Internet Explorer, and navigate to your Rainier instance base URL.
3. Navigate to the **Reservation Management workspace** under **App links > Fleet Management > Workspaces > Reservation Management**.
4. Select the KPI tile **Total Revenue**. KPI details page for **Total Revenue** KPI will be displayed.
5. To navigate to the newly defined KPI, select the **Show list** icon on top left. From the list of KPIs shown, select **FMTRevenuePerRental**.

Notice that the KPI details page for the new KPI, **FMTRevenuePerRental** is shown. Even though we did not define trend charts, the system created a set of charts based on the limited metadata defined by the developer. Users have the ability to modify KPI definitions and create new ones in the client. Next, you will modify the newly defined KPI.

6. To demonstrate this capability, you can change the KPI Goal. Click the **Edit** button on top left, and enter 900 as the Goal Value.
7. Modify the threshold properties as follows:

PROPERTY	VALUE
Threshold Type	Percentage
Red if less than	90
Green if more than	110

8. Click **Save** on the bottom left to save changes. Notice that the KPI status color has changed in the KPI tile shown.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add financial dimensions to aggregate measurements

2/18/2021 • 6 minutes to read • [Edit Online](#)

This feature lets power users include financial dimensions in ready-made Microsoft Power BI reports. Power users can also create new Power BI reports that use financial dimensions.

Financial dimensions are user-defined configuration data that enables the ledger chart of accounts to retain additional information. By adding financial dimensions, a user can configure additional data fields that will be included with the chart of accounts and financial reports. Therefore, the ledger chart of accounts can be sliced and diced based on those fields. This feature provides powerful financial reporting. For example, you can explore ledger data by using the new financial dimensions that you've added. Because no financial dimensions are included in the ready-made Power BI reports when they are released, you must include these additional fields to the ready-made reports.

After financial dimension fields are included in the report, you can share the report with other users. Those other users can modify existing visuals, such as charts, by including all or some of the financial dimension fields.

## NOTE

This feature is available in Microsoft Dynamics 365 for Finance and Operations, Enterprise edition (July 2017). Platform update 8 or later is also required.

## How does this feature work?

The Entity store is an operational data warehouse that lets power users create reports. Whenever the Entity store is updated, all available financial dimensions are included in it. We will look at an example from the **Ledger Activity** aggregate measurement that contains General ledger journal-level details.

The following list shows some of the tables are filled in the Entity store when it's updated. Tables that have changed are bold.

- LedgerActivityMeasure\_LedgerActivityMeasureGroup
- LedgerActivityMeasure\_TransactionDate
- LedgerActivityMeasure\_Currency
- LedgerActivityMeasure\_FiscalPeriodDateAggregateDimension
- LedgerActivityMeasure\_LedgerFactDimension
- LedgerActivityMeasure\_FiscalYearOffsetDimension
- LedgerActivityMeasure\_MainAccount
- **LedgerActivityMeasure\_DimensionCombination**
- LedgerActivityMeasure\_Ledger
- LedgerActivityMeasure\_MainAccountCategory
- LedgerActivityMeasure\_Company
- LedgerActivityMeasure\_MainAccountLegalEntity
- **LedgerActivityMeasure\_Agreement**
- **LedgerActivityMeasure\_BankAccount**
- **LedgerActivityMeasure\_BusinessUnit**

- LedgerActivityMeasure\_Campaign
- LedgerActivityMeasure\_Cargo
- LedgerActivityMeasure\_Cargo\_CN

Notice that, for each financial dimension that is defined in your system, you might see a corresponding table in the Entity store.

If you examine the LedgerActivityMeasure\_DimensionCombination table, you will notice that the list of fields has been expanded and now includes additional fields. Each new field corresponds to a financial dimension. For an example, here are some of the additional fields:

- Agreement\_FK
- Agreement\_Description
- Agreement\_Value
- BankAccount\_FK
- BankAccount\_Description
- BankAccount\_Value
- BusinessUnit\_FK
- BusinessUnit\_Description
- BusinessUnit\_Value

If a user defines a new financial dimension that is named **vendor**, when the Entity store is updated, a new table is added that is named LedgerActivityMeasure\_Vendor.

The LedgerActivityMeasure\_DimensionCombination table will also contain the following new set of fields:

- Vendor\_FK
- Vendor\_Description
- Vendor\_Value

## How a Power BI report author can create reports that use financial dimensions

A business user can create a new report that uses financial dimensions by using Power BI desktop. Existing reports that use financial dimensions can be updated so that they include additional fields.

In this example, we will use Power BI desktop to create a report that uses the Ledger Activity measure group.

1. In a development environment, connect to the Entity store database by using Power BI desktop.
2. Select the following tables:
  - LedgerActivityMeasure\_LedgerActivityMeasureGroup
  - LedgerActivityMeasure\_FiscalPeriodDateAggregateDimension
  - LedgerActivityMeasure\_DimensionCombination
3. Use the **Manage relationships** option in Power BI desktop to define the following relationships between table fields:
  - Define a join between **LedgerActivityMeasure\_LedgerActivityMeasureGroup.LEDGERDIMENSION** and **LedgerActivityMeasure\_DimensionCombination.DIMENSIONCOMBINATIONRECID**.
  - Define a join between **LedgerActivityMeasure\_LedgerActivityMeasureGroup.LEDGERGREGORIANDATEID** and **LedgerActivityMeasure\_FiscalPeriodDateAggregateDimension.LEDGERPERIODGREGORIANDATEID**.

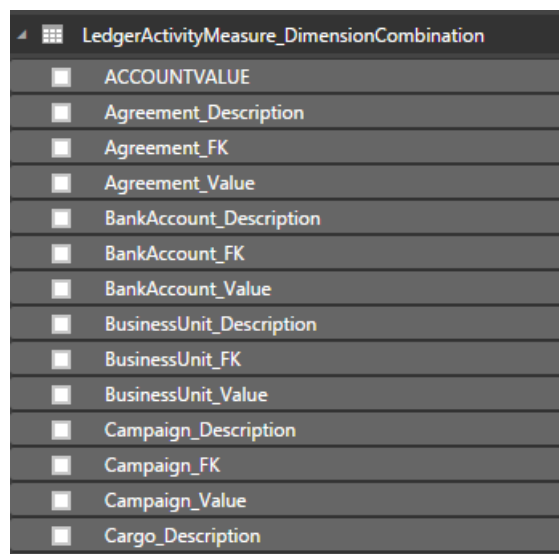


4. Create a matrix report that uses the **Sales** and **YearName** fields. The report should resemble the following example.

YEARNAME	SALES
2015	355,958,240.86
2016	478,064,122.48
2017	10,712,838.30
<b>Total</b>	<b>844,735,201.64</b>

Next, we will add financial dimension values.

5. In the list of fields in Power BI desktop, expand the **LedgerActivityMeasure\_DimensionCombination** table. You will see that the list of dimension fields is expanded into table fields, as shown here.

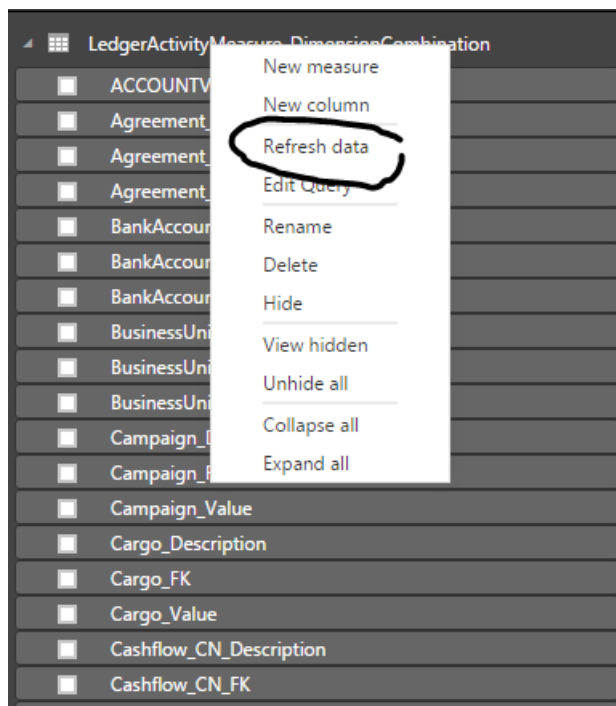


6. Include the **BusinessUnit\_Description** field in the report. Your report should now show sales by business unit, as shown in the following example.

YEARNAME	Electronics	Home	IT Consulting Practice	Management Consulting Practice	Sporting	Total
2015	7,362,985.20	152,524,507.23	100,045,909.46	699,928.66	7,937,635.30	<b>355,958,240.86</b>
2016	6,374,315.40	167,483,885.95	115,009,475.39	769,787.36	7,622,420.20	<b>478,064,122.48</b>
2017	100,000.00	10,390,021.77	0.00	26,571.26		<b>10,712,838.30</b>
<b>Total</b>	<b>13,837,300.60</b>	<b>330,398,414.95</b>	<b>215,055,384.85</b>	<b>1,496,287.28</b>	<b>15,560,055.50</b>	<b>844,735,201.64</b>

Notice that you have the **BusinessUnit\_description** field and the **BusinessUnit\_Value** field. The value field lets you to get a numerical value that can be used to sort columns.

7. Define a new financial dimension, and update the Entity store.
8. When the update is completed, right-click the **LedgerActivityMeasure\_DimensionCombination** table, and then select **Refresh data**. Notice that the new financial dimension that you defined is reflected in the list of fields that are available for reporting.



9. You can include the new financial dimension fields in the report.

### Creating reports that use expanded Financial dimension tables

As we discussed earlier, new tables were created in the Entity store. Each financial dimension field is also added to a new table. The new tables contain individual fields for the name, value, and description.

Notice that the new tables contain a key that can be used to join them with the LedgerActivityMeasure\_DimensionCombination table that is used to create the report. If you want to use fields from these additional tables, just include them in the report and relate them by using the keys.

1. Open the report that you created earlier.

We will now add a financial dimension table to the report.

2. In Power BI desktop, click **Recent Sources** on the menu, and then select the **AXDW** data connection.

Table navigator is shown.

3. Select the **LedgerActivityMeasure\_LegalEntity** table for the report.

#### NOTE

If you're using a development or demo environment, you see this table because **LegalEntity** is a financial dimension that is defined in demo data. If you're working with your own data, the tables that you see will correspond to financial dimensions that you've defined.

We will now relate the selected table to existing tables in the report.

4. In the report, open the **Manage relationships** dialog box.

5. Join **LedgerActivityMeasure\_DimensionCombination.LegalEntity\_FK** to **LedgerActivityMeasure\_LegalEntity.KEY\_**.

If you select a different dimension table in step 3, you must relate the corresponding **FieldName\_FK** field from the combination table to the **KEY\_** field in the dimension table.

## How a developer can enable financial dimensions

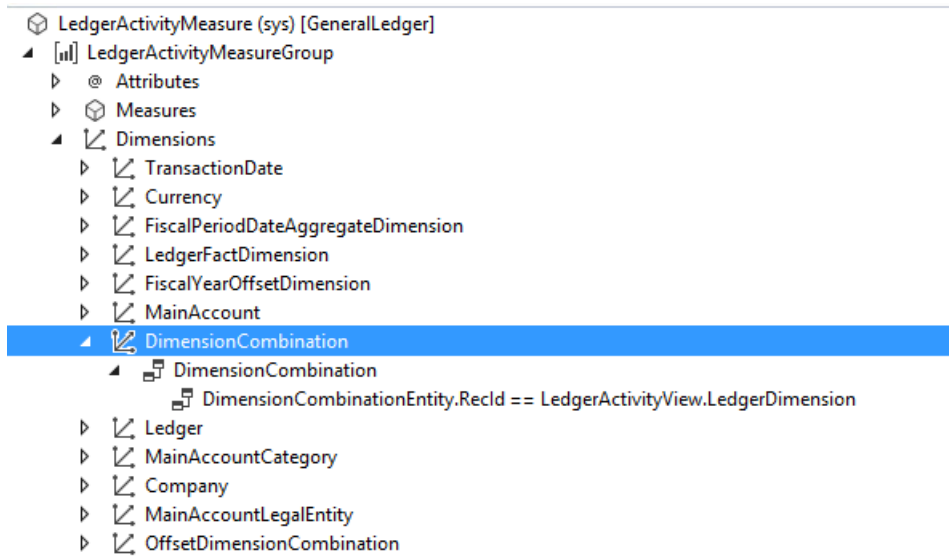
Before users can see expanded financial dimensions in the Entity store, a developer must enable the use of financial dimensions in aggregate measurements. As a best practice, financial dimensions should be included when you model any aggregate measurement that involves financial data. To include financial dimensions, create an aggregate dimension that is based on "financial dimension tables." As we saw earlier, an aggregate dimension that you create by using financial dimension tables will be expanded at runtime.

### What are financial dimension tables?

Financial dimension tables are base tables that contain financial dimension data. Examples include `DimensionAttributeValueCombination` and `DimensionAttributeValueSet`.

If you use a table, a view, or an entity that is based on those two tables, the fields of the aggregate dimension will be expanded at runtime.

Consider the following example from the `LedgerActivityMeasure` aggregate measurement.



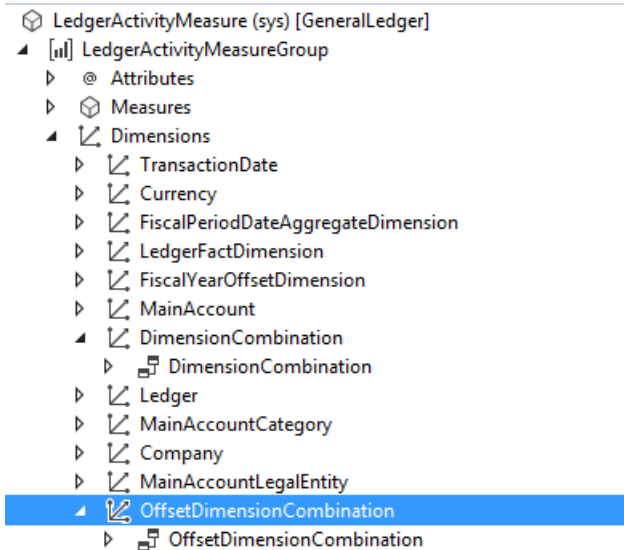
`DimensionCombination` is an aggregate dimension that is modeled by using the `DimensionAttributeValueCombination` base table. In this case, the developer has referenced the aggregate dimension by using the `LedgerActivityMeasureGroup` measure group.

At runtime, new dimension fields are added (that is, the `DimensionCombination` table is expanded) as new financial dimensions are defined in the system.

### Creating role-playing financial dimensions

When you report by using ledger data, you might require reporting on primary accounts and offset accounts. For an example, if a ledger transaction involves the transfer of an amount from one account to another account, the primary account is the "from" account, whereas the offset account is the "to" account. This pattern is known as the *role-playing dimensions* pattern.

Both primary accounts and offset accounts must be associated with transaction data. Therefore, you must expand financial dimension fields of both primary accounts and offset accounts. We will now see how you can meet this requirement. Consider the following example.



We have modeled two dimension references for LedgerActivityMeaureGroup. The first reference, DimensionCombination, is joined by using the **LedgerDimension** field. We saw this pattern earlier in this topic.

The second reference, OffsetDimensionCombination, is another reference to the same dimension. We have reused the DimensionCombination aggregate dimension and given it a new name. In the second case, we can join by using the **OffsetLedgerDimension** field.

At runtime, the system will expand both these dimensions with additional fields. Therefore, you can report on primary and offset dimension fields.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create analytical reports by using Power BI Desktop

2/18/2021 • 4 minutes to read • [Edit Online](#)

If you're a power user or a business analyst, you probably create many reports for your organization. You might create these reports in Microsoft Excel by formatting and relating data before you share it with other people. People in your organization might even come to you when they require modifications to the report. This solution offers an easy way to create rich, interactive reports. As a report writer, you can use Microsoft Power BI Desktop as the reporting tool. The reports that you create can then be published to PowerBI.com. For more information about Power BI Desktop, see [Create stunning reports and visualizations with Power BI Desktop](#).

## Accessing the local Entity Store by using DirectQuery

You can create Microsoft Power BI reports by using Open Data Protocol (OData) endpoints that are exposed via data entities. Despite the limitations of this approach, the Entity Store still supports it for legacy solutions. However, DirectQuery is now the preferred method for sourcing data for analytical solutions. For more information about the benefits and limitations of DirectQuery, see [Use DirectQuery in Power BI Desktop](#).

When you use Power BI Desktop, you can create a report in your development or test environment by connecting directly to the local Entity Store database. When you're satisfied with the report, your administrator can help you migrate it to your production environment. The rest of this section walks you through this process.

### NOTE

To develop or extend analytical workspaces and reports in the application suite, customers must use a development environment running in their own subscription or on local machines. You won't be able to develop or extend embedded analytical reports in Microsoft-provided Tier-1 environments. You need administrator rights to install Power BI Desktop.

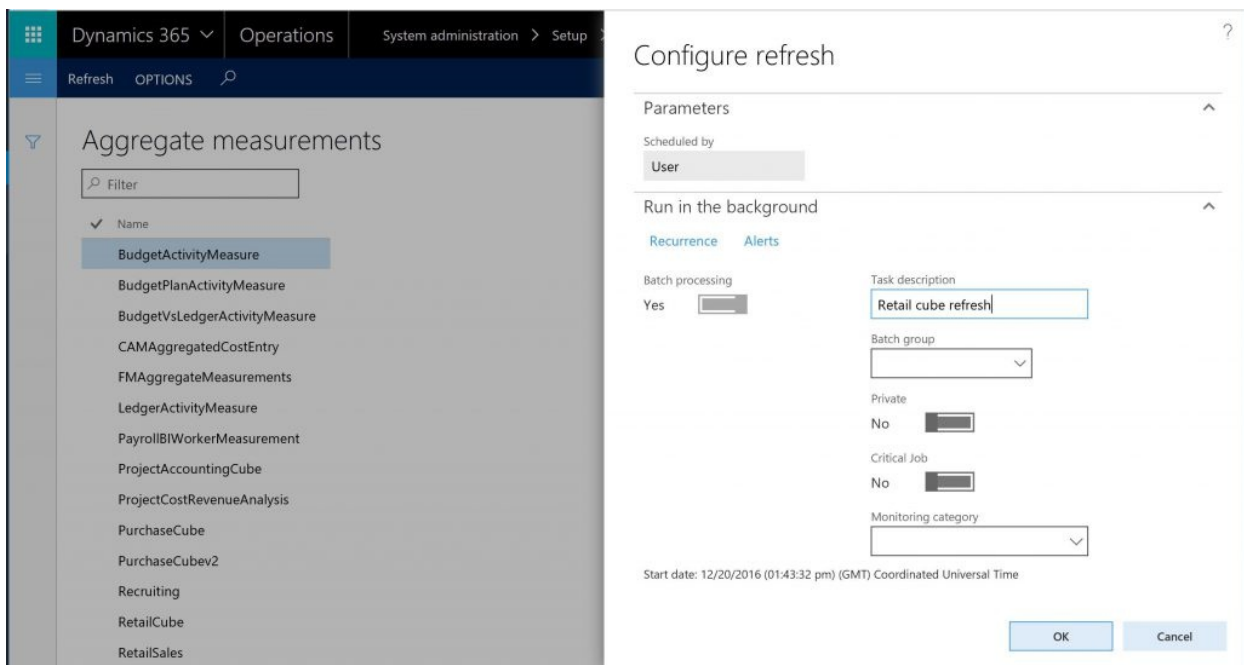
Tier-1 environments now include a service compatible version of Power BI Desktop. To develop or extend analytical workspaces and reports in the application suite, customers can use the Power BI Desktop application pre-installed on the development environment. Alternatively, you can use the latest compatible release of Power BI Desktop with Preview features turned off to author analytical reports for Finance and Operations apps. Download the August 2020 Update of Power BI Desktop at [Previous monthly updates to Power BI Desktop](#).

### Step 1: Populate the local Entity Store database

For this example, we will stage the aggregate models that the Commerce analytical solution consumes in the local Entity Store. The models that the application uses are defined in the RetailCube aggregate measurement.

1. In the client, open the **Entity Store** page. (Select **System administration** > **Setup** > **Entity Store**.)
2. Select the **RetailCube** aggregate measurement, and then select **Refresh**.
3. Enter a name for the job that will be run in the background, and then select **OK**.

The following illustration shows the administrator dialog box that is used to configure the frequency of updates for the aggregate model.

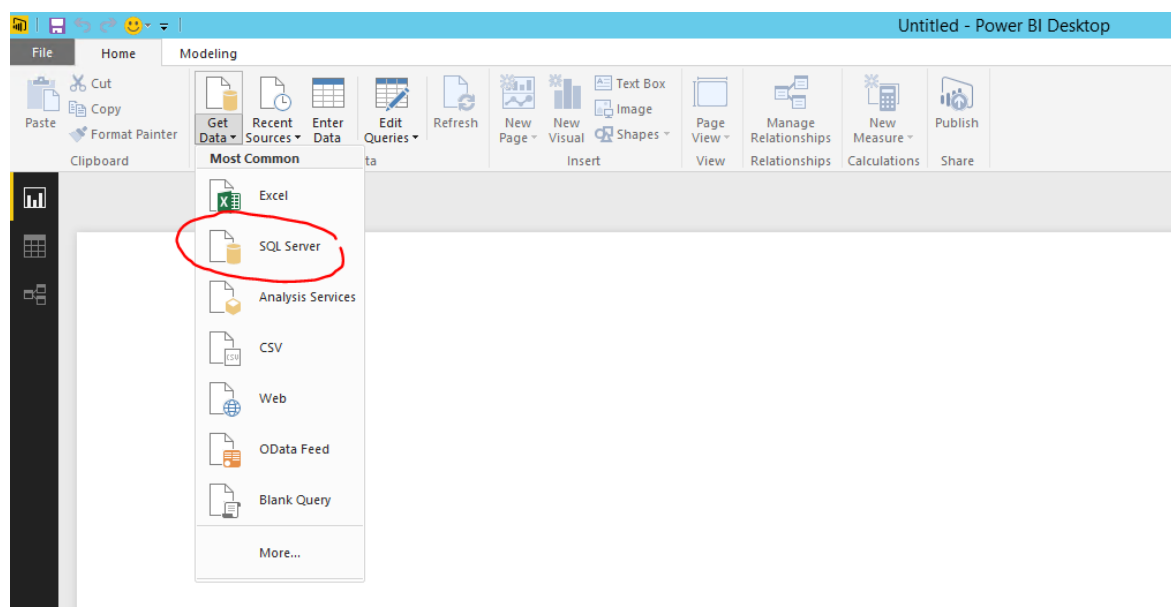


To monitor the progress of the job that stages the data, you can use the batch job monitoring page. (Select **System administration** > **Database** > **Batch jobs**.) If you're using demo data, the job should take about a minute. After the data is in the Entity Store, you can write reports.

## Step 2: Connect to the local Entity Store database

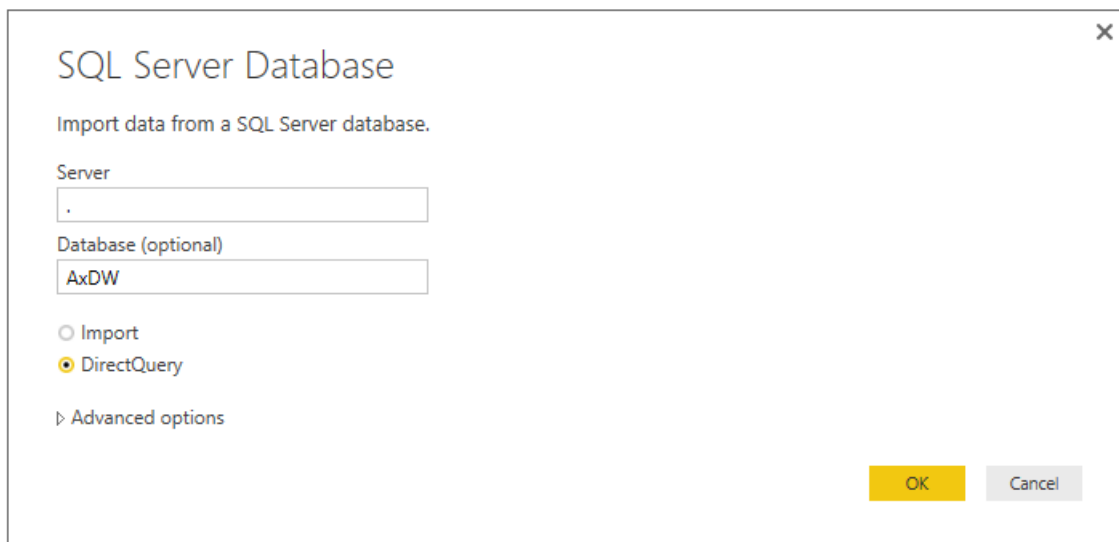
1. Start Power BI Desktop. If any updates are available for Power BI Desktop, you might have to download and apply them.
2. On the Power BI **Welcome** page, select **Get data**.

Alternatively, when Power BI Desktop starts, you can select **Get Data** > **SQL Server**.



3. In the **SQL Server Database** dialog box, enter . as the server name and **AxDW** as the database name. Then select the **DirectQuery** option.

The following illustration shows the settings that enable Power BI Desktop to access the local Entity Store database.



**NOTE**

The **Import** option isn't currently supported.

4. Select **OK**.

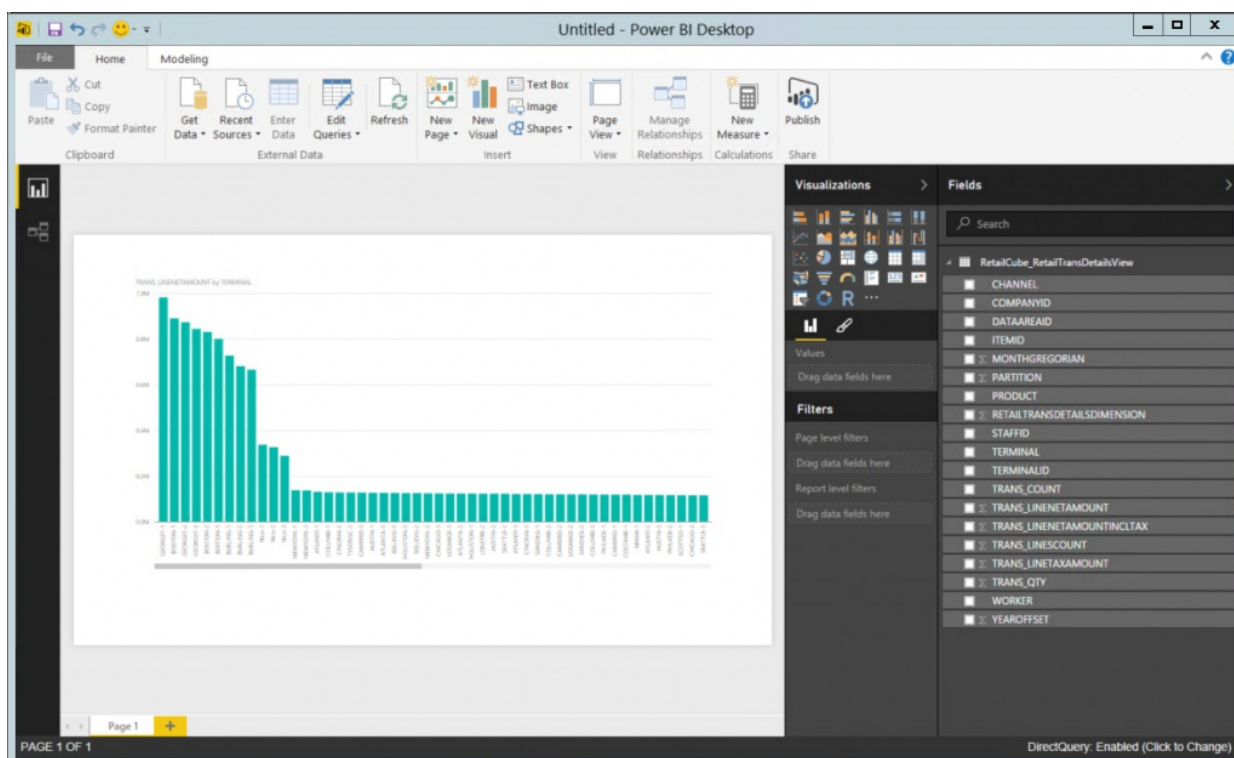
The **Navigator** dialog box appears. You use this dialog box to select which tables and views from the Entity Store you want to report on.

5. In the search box, enter **Retail** to filter for entities that are related to the RetailCube aggregate measurement.

6. Select the **RetailCube\_RetailTransDetailsView** table that is shown in the navigator, and then select **Load**.

You can now create a report. You can drag measures and fields to the canvas, and can explore data and trends interactively.

The following illustration shows a basic report that uses the local Entity Store database as its source.



Power BI Desktop also supports the creation of calculations and lets you combine data from multiple aggregate measurements. Within minutes, you can create analytical reports by using data in the local development environment. When you're satisfied with the report, you can migrate it to the production environment, so that users can use the report to interact with production data.

## Validating reports in a demo environment

The report shows the demo or test data in your developer environment. If you want to integrate the report into a demo environment, you can continue to publish this report to your PowerBI.com account and pin it to the client.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Add analytics to workspaces by using Power BI Embedded

2/18/2021 • 5 minutes to read • [Edit Online](#)

## NOTE

This feature is supported in Finance and Operations (version 7.2 and later).

## Introduction

This topic shows how to embed a Microsoft Power BI report on the **Analytics** tab of a workspace. For the example that is given here, we will extend the **Reservation management** workspace in the Fleet Management application to embed an analytical workspace on an **Analytics** tab.

## Prerequisites

- Access to a developer environment that runs Platform update 8 or later.
- An analytical report (.pbix file) that was created by using Microsoft Power BI Desktop, and that has a data model that is sourced from the Entity store database.

## Overview

Whether you extend an existing application workspace or introduce a new workspace of your own, you can use embedded analytical views to deliver insightful and interactive views of your business data. The process for adding an analytical workspace tab has four steps.

1. Add a .pbix file as a Dynamics 365 resource.
2. Define an analytical workspace tab.
3. Embed the .pbix resource on the workspace tab.
4. Optional: Add extensions to customize the view.

## NOTE

For more information about how to create analytical reports, see [Getting started with Power BI Desktop](#). This page is a great source for insights that can help you create compelling analytical reporting solutions.

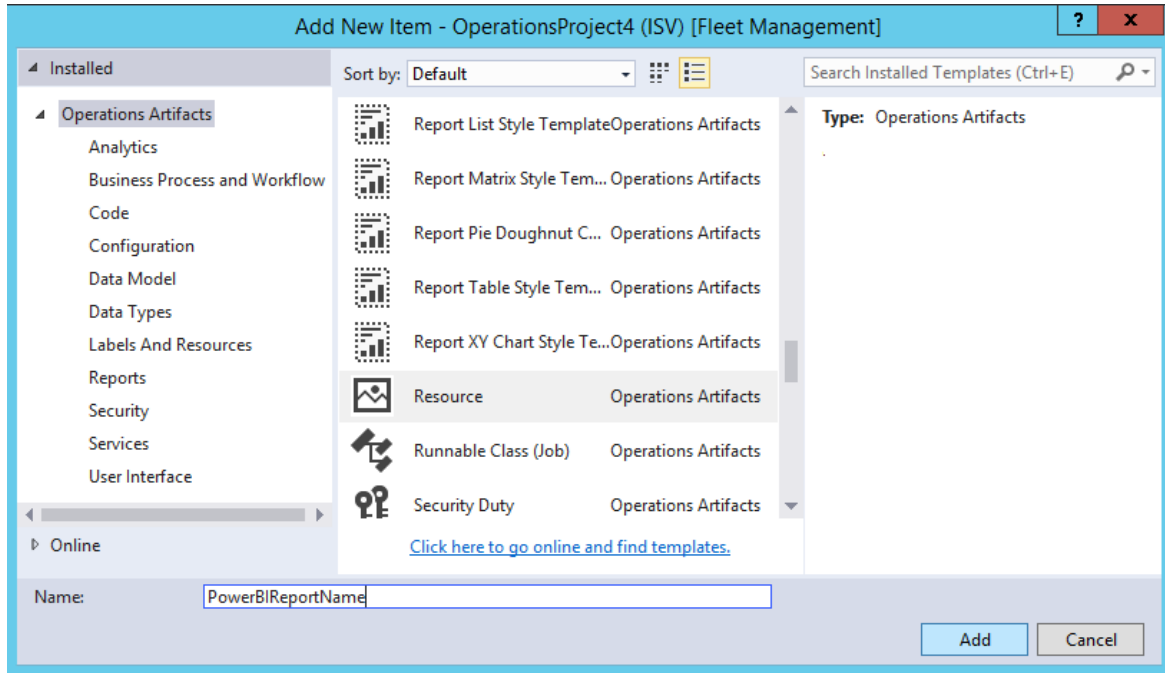
## Add a .pbix file as a resource

Before you begin, you must create or obtain the Power BI report that you will embed in the workspace. For more information about how to create analytical reports, see [Getting started with Power BI Desktop](#).

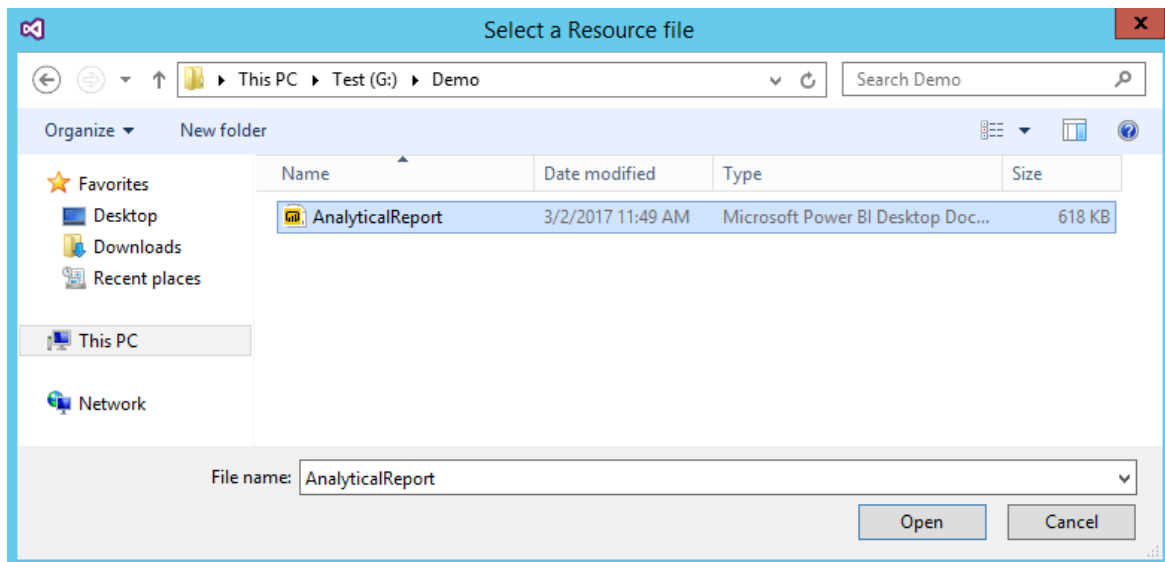
Follow these steps to add a .pbix file as a Visual Studio project artifact.

1. Create a new project in the appropriate model.
2. In Solution Explorer, select the project, right-click, and then select **Add > New Item**.
3. In the **Add New Item** dialog box, under **Operations Artifacts**, select the **Resource** template.

4. Enter a name that will be used to reference the report in X++ metadata, and then click **Add**.



5. Find the .pbix file that contains the definition of the analytical report, and then click **Open**.

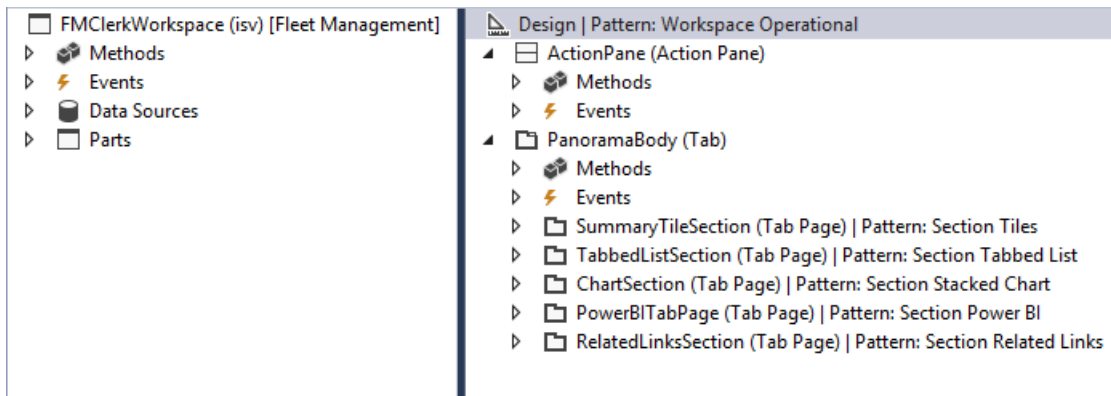


Now that you've added the .pbix file as a Dynamics 365 resource, you can embed the reports in workspaces and add direct links by using menu items.

## Add a tab control to an application workspace

In this example, we will extend the **Reservation management** workspace in the Fleet Management model by adding the **Analytics** tab to the definition of the **FM ClerkWorkspace** form.

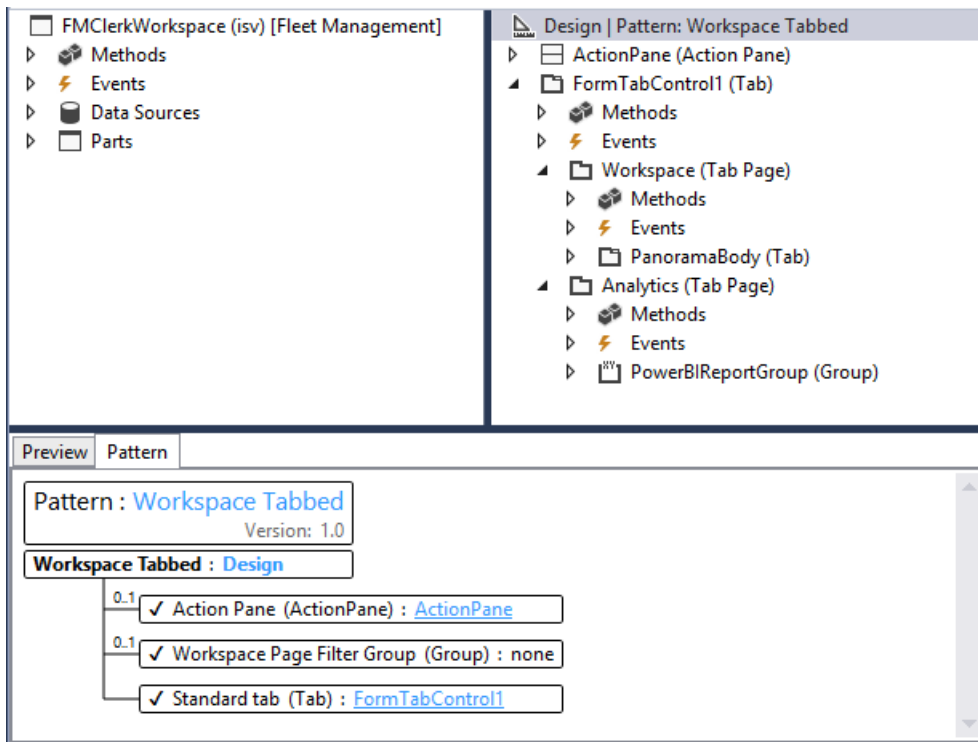
The following illustration shows what the **FM ClerkWorkspace** form looks like in the designer in Microsoft Visual Studio.



Follow these steps to extend the form definition for the **Reservation management** workspace.

1. Open the form designer to extend the design definition.
2. In the design definition, select the top element that is labeled **Design | Pattern: Workspace Operational**.
3. Right-click, and then select **New > Tab** to add a new control that is named **FormTabControl1**.
4. In the form designer, select **FormTabControl1**.
5. Right-click, and then select **New Tab Page** to add a new tab page.
6. Rename the tab page to something meaningful, such as **Workspace**.
7. In the form designer, select **FormTabControl1**.
8. Right-click, and then select **New Tab Page**.
9. Rename the tab page to something meaningful, such as **Analytics**.
10. In the form designer, select **Analytics (Tab Page)**.
11. Set the **Caption** property to **Analytics**, and set the **Auto Declaration** property to **Yes**.
12. Right-click the control, and then select **New > Group** to add a new form group control.
13. Rename the form group to something meaningful, such as **powerBIReportGroup**.
14. In the form designer, select **PanoramaBody (Tab)**, and then drag the control onto the **Workspace** tab.
15. In the design definition, select the top element that is labeled **Design | Pattern: Workspace Operational**.
16. Right-click, and then select **Remove pattern**.
17. Right-click again, and then select **Add pattern > Workspace Tabbed**.
18. Perform a build to verify your changes.

The following illustration shows what the design looks like after these changes are applied.



Now that you've added the form controls that will be used to embed the workspace report, you must define the size of the parent control so that it accommodates the layout. By default, both the **Filters Pane** page and the **Tab** page will be visible on the report. However, you can change the visibility of these controls as appropriate for the target consumer of the report.

#### NOTE

For embedded workspaces, we recommend that you use extensions to hide both the **Filters Pane** and **Tab** pages, for consistency.

You've now completed the task of extending the application form definition. For more information about how to use extensions to do customizations, see [Customize through extension and overlayering](#).

## Add X++ business logic to embed a viewer control

Follow these steps to add business logic that initializes the report viewer control that is embedded in the **Reservation management** workspace.

1. Open the **FM ClerkWorkspace** form designer to extend the design definition.
2. Press F7 to access the code behind the code definition.
3. Add the following X++ code.

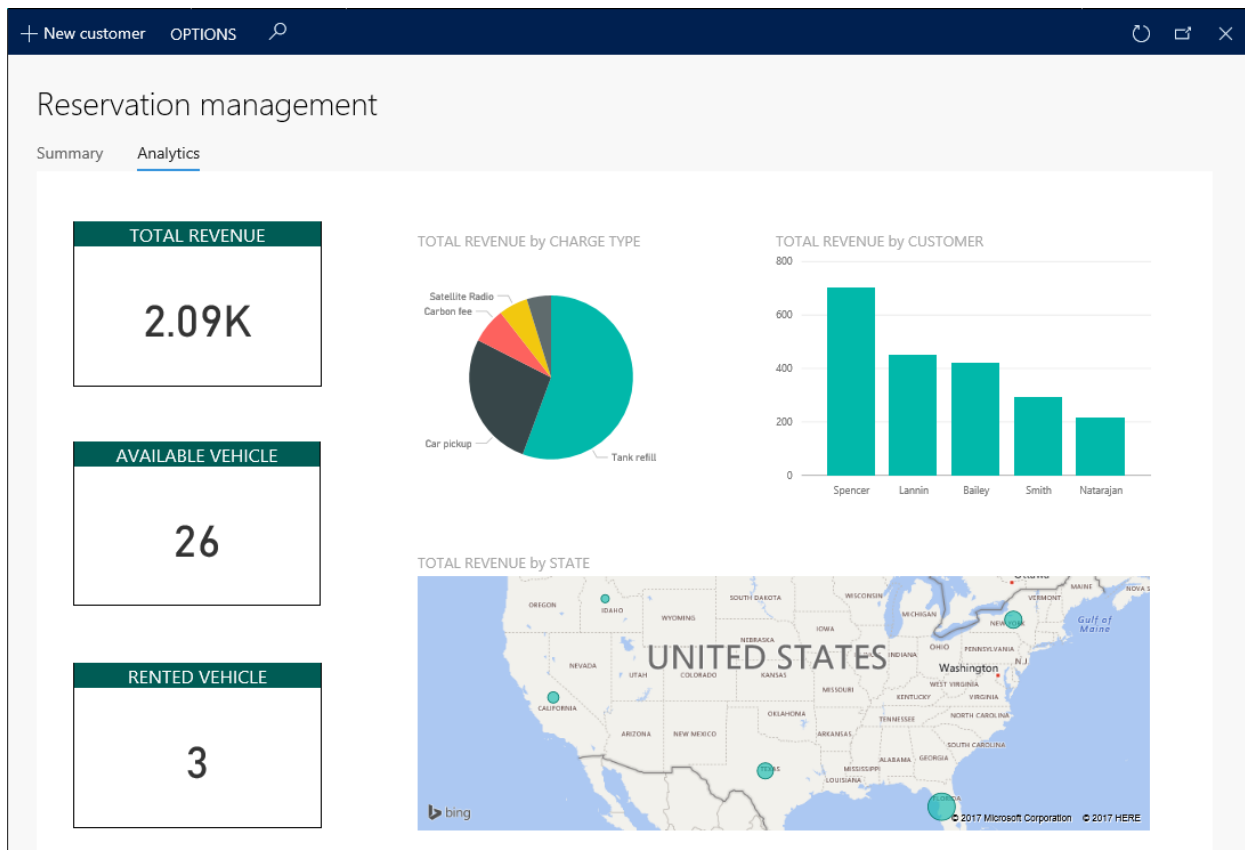
```

[Form]
public class FMCLerkWorkspace extends FormRun
{
    private boolean initReportControl = true;
    protected void initAnalyticalReport()
    {
        if (!initReportControl)
        {
            return;
        }
        // Note: secure entry point into the Workspace's Analytics report
        if (Global::hasMenuItemAccess(menuItemDisplayStr(FMCLerkWorkspace), MenuItemType::Display))
        {
            // initialize the PBI report control using shared helper
            PBIReportHelper::initializeReportControl('FMPBIWorkspaces', powerBIReportGroup);
        }
        initReportControl = false;
    }
    /// <summary>
    /// Initializes the form.
    /// </summary>
    public void init()
    {
        super();
        this.initAnalyticalReport();
    }
}

```

4. Perform a build to verify your changes.

You've now completed the task of adding business logic to initialize the embedded report viewer control. The following illustration shows what the workspace looks like after these changes are applied.



## NOTE

You can access the existing operational view by using the workspace tabs below the page title.

# Reference

## PBIReportHelper.initializeReportControl method

This section provides information about the helper class that is used to embed a Power BI report (.pbix resource) in a form group control.

### Syntax

```
public static void initializeReportControl(  
    str                _resourceName,  
    FormGroupControl  _formGroupControl,  
    str                _defaultPageName = '',  
    boolean            _showFilterPane = false,  
    boolean            _showNavPane = false,  
    List               _defaultFilters = new List(Types::Class))
```

### Parameters

NAME	DESCRIPTION
resourceName	The name of the .pbix resource.
formGroupControl	The form group control to apply the Power BI report control to.
defaultPageName	The default page name.
showFilterPane	A Boolean value that indicates whether the filter pane should be shown ( <b>true</b> ) or hidden ( <b>false</b> ).
showNavPane	A Boolean value that indicates whether the navigation pane should be shown ( <b>true</b> ) or hidden ( <b>false</b> ).
defaultFilters	The default filters for the Power BI report.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Help secure analytical workspaces and reports by using Power BI Embedded

2/18/2021 • 5 minutes to read • [Edit Online](#)

## NOTE

This feature is supported in Microsoft Dynamics 365 for Finance and Operations, Enterprise edition (July 2017) (version 7.2) and later releases.

## Introduction

This topic provides a walk-through for application developers who want to help secure analytical workspaces and reports that are delivered by using Microsoft Power BI Embedded. It describes the recommended strategies for securing access to both the reports and the data set, based on viewer access rights. By using the techniques that are described in this topic, you can hide reports from users and filter reports to show the data set that is appropriate for a specific user, based on the active company context.

## Prerequisites

- Access to a developer environment that runs Platform update 8 or later
- An analytical report (.pbix file) that was created by using Microsoft Power BI Desktop, and that has a data model that is sourced from the Entity store database

## Overview

Whether you're extending an existing application workspace or adding your own workspace, you can use embedded analytical views to deliver insightful and interactive views of your business data. Before you add new analytical workspaces and reports, it's important that you establish a strategy to help secure the content.

There are several considerations that you should be aware of when you develop analytical solutions by using Power BI Embedded. Analytical reports are secured by using menu items. After they have access to a report, all viewers can access the underlying data model that is defined in the report. Although service options are available that automatically hide the fields that back a report data set, all viewers of the report have effective access to the fields in the data model. Additionally, X++ extensions are available that influence the way that the report is presented in the client. Both the **Filter** pane and the **Report** tabs can be hidden. However, Microsoft Power BI filters can be modified by using client-side script injections.

### Recommendation

Create scenario-specific .pbix files to deliver analytical views:

- Area overviews that are delivered by using workspaces
- Subject matter-specific analytical reports

## NOTE

These analytical reports are often used to deliver reports that contain medium-business-impact and high-business-impact data.

For more information about how to create analytical reports, see [Getting started with Power BI Desktop](#). This page is a great source for insights that can help you create compelling analytical reporting solutions.

## Help secure analytical views that are provided through embedded Power BI reports

Power BI report filters and the **Filter** pane serve as a mechanism for passing session context into the report that is embedded on the **Analytics** tab. The capability to turn the visibility of the **Filter** pane on and off isn't a security feature. Power BI report filters and the capability to hide and show the **Filter** pane are user experience (UX) decisions that the application designer makes.

Row-level security that is defined isn't inherited by Power BI reports. Instead, application developers can help secure the workspace or the report.

### Help secure analytical workspaces on the Analytics tab

Analytical workspaces are embedded Power BI reports that are shown in a form control. Unless you complete the following procedure, anyone who has access to the workspace can see the **Analytics** tab and access the Power BI reports.

1. Add a menu item for the analytical workspace.
2. Verify that the form initialization uses the **hasMenuItemAccess** application programming interface (API) to verify that the user has access to the menu item.

```
// Note: secure entry point into the Workspace's Analytics report
if (Global::hasMenuItemAccess(menuItemDisplayStr(FMClerkWorkspace), MenuItemType::Display))
{
    FMPBIWorkspaceController controller = new FMPBIWorkspaceController();
    PBIReportHelper::initializeReportControl('FMPBIWorkspaces', powerBIReportGroup);
}
```

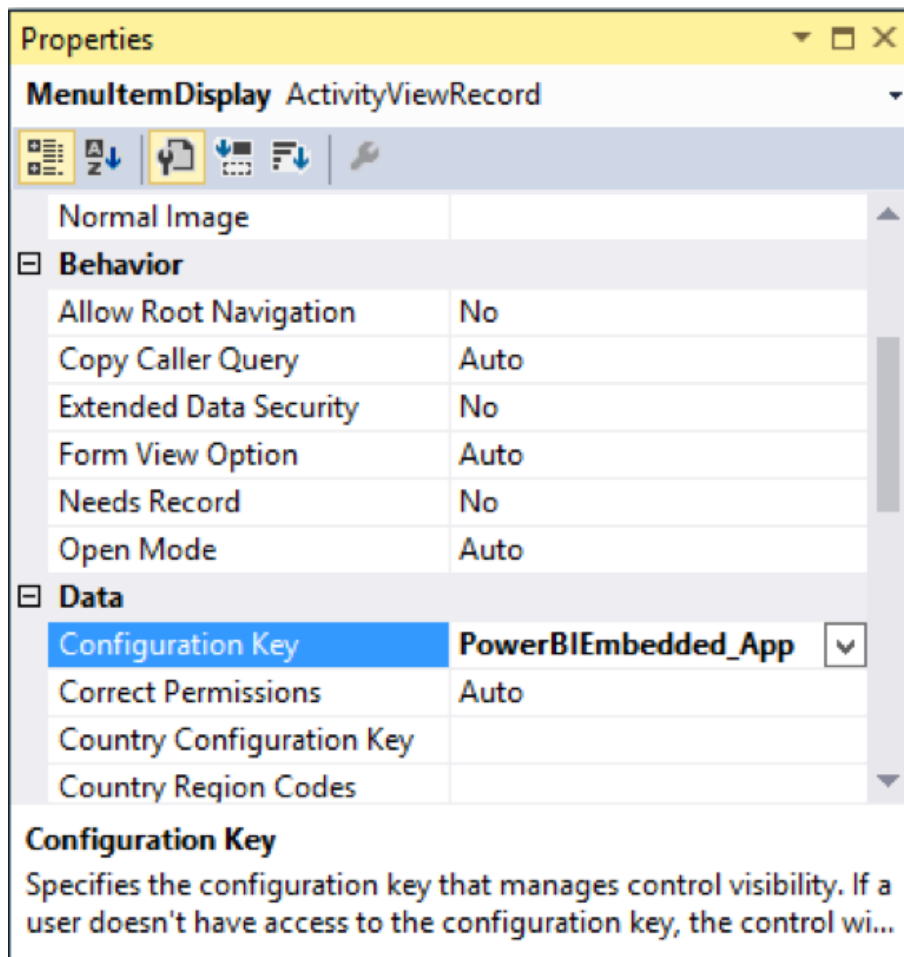
The preceding logic will prevent the Power BI Viewer control from being initialized. Therefore, an empty tab will appear on the page. By default, the framework automatically hides empty tabs. Therefore, the **Analytics** tab is hidden and can't be access if the user doesn't have access to the menu item that is associated with the analytical workspace.

### Help secure analytical reports

Embedded Power BI reports in the application are secured by using menu items. Users who try to access a Power BI report directly, by using a menu item in application, will receive an error. Follow these steps to help secure the analytical reports.

1. Add a menu item for the report or the appropriate tab. By default, the first tab of the report will be shown if no other tab is selected.
2. Link the menu item to the **PowerBIEmbedded\_App** configuration key.





The menu item is now associated with the availability of the Power BI Embedded service. If the service is unavailable, the links for the menu items will be removed from the application.

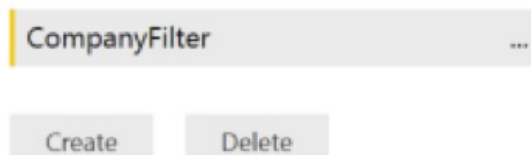
## Help secure analytical workspaces and reports by company

Power BI workspaces and reports can be secured by company (for example, **DataAreaID** value). Application solutions must apply the following steps for company-level security in analytical workspaces and reports.

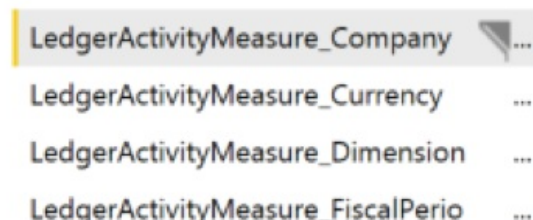
In this scenario, the workspaces and reports that the sales manager from Contoso USA sees are limited to data that is related to Contoso USA. The report viewer must not have access to data that associated with any other company, unless the company context is changed.

1. Open the analytical report in Power BI Desktop by double-clicking the resource in a Microsoft Visual Studio project.
2. On the **Modeling** tab, click **Manage Roles**.
3. Create a new role against a column in the data model that contains the **Company** field. Name the new role **CompanyFilter**. A **COMPANY** field must be present in the data model to restrict access by company.

### Roles



### Tables



4. In the **Table filter DAX expression** field, enter `[COMPANY]=username()`.
5. To make sure that the rules work, on the **Modeling** tab, click **View as Roles**. In the dialog box, set the following fields:
  - a. Clear the **None** check box.
  - b. Select the **Other user** check box, and then enter **USMF** in the text box.
  - c. Select the **CompanyFilter** check box.

The reports will now show data as if you're running the USMF company.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create reporting solutions

2/18/2021 • 10 minutes to read • [Edit Online](#)

This tutorial shows how to export data and create a report, expand predefined views and add navigation to charts, use the free-form report designer, and customize the parameter experience.

## Prerequisites

For this tutorial, you must access the application environment, and you must be provisioned as an administrator on the instance.

## Key concepts

- Describe the various ways reports can be created and consumed
- Add interactivity to embedded aggregate reports in forms and workspaces
- Use framework extensions to customize the parameter experience for SSRS based business documents
- Export List Page data to create reports with external tools including Microsoft Excel
- Author modern report designs using enhanced developer tooling in Visual Studio

## What's new in Reporting?

- Embedded report drill-through navigations to AX forms and reports
- Navigate between reports using embedded report drill-through links
- Several Document Routing Service enhancements including support for custom print settings
- Introduced Document Brand Management administrative tools
- Additional visualizations available through the Embedded Charting Control
- Dates and Amounts in the report body are formatted based on the "Date, time, and number format" user setting
- Network Print monitoring form
- Table extensions need to be supported through the VS Query picker

## What is a report in Dynamics 365 Finance and Operations apps?

Reports can be defined simply as any visualization of a structured data set. This may include transactional data presented in a tabular layout and advanced graphical views of aggregate information. To account for this broad definition, the application offers several tools to produce reports to satisfy complex business requirements. Some common applications of reports in an ERP include:

- Creating and archiving transactional documents as part of a posting process
- Producing packing slips for tracking orders from Manufacturing to Warehousing to Sales
- Monitoring key performance metrics and surfacing trends in data
- Navigating the client through filtered searches
- Distributing heavily branded documents to customers and employees
- Extracting data in such a way that it articulates the health of a business

The most difficult task for developers is selecting the *right* Business Intelligence visualization tool for the job, given a customer's requirements. To accomplish this, it's important to understand the capabilities offered through the tools that are available for creating reports. We offer tooling to support the following basic

reporting requirements:

- **Excel Integration** – Allows data management and analysis using Microsoft Excel
- **Embedded Analytics** – Add aggregate data to a Workspaces using native controls like charts and grids
- **Reporting Services** – Create business documents that require precision using SSRS-based solutions
- **Power BI Integration** – Author and share reports that can be accessed anywhere
- **Management Reporter** – Designed to help users create financial reports

The following table can be used to compare the basic characteristics of these reporting tools:

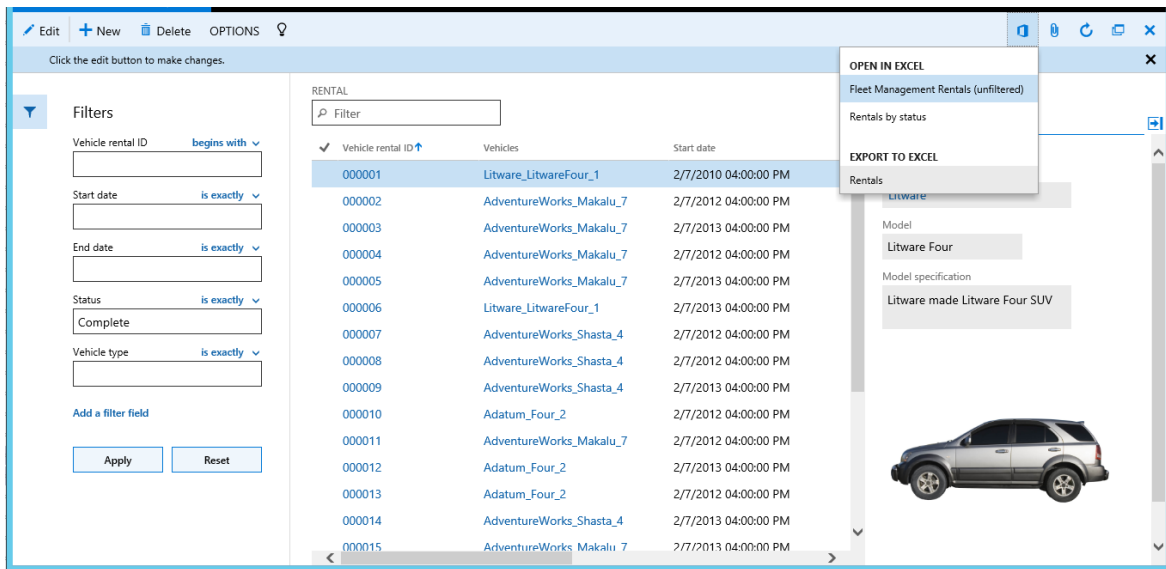
TOOL	AUTHOR	TARGET	DATA	CONSUMPTION	DESIGNER
Excel	Power User	Power User	Transactional	External	Free form
Embedded BI	Developer	All users	Aggregates	Internal	Modeled
SSRS Report	Developer	All users	Transactional and Aggregates	Internal/External	Free form
Power BI	Power User	All users	Aggregates	Internal/External	Free form
Management Reporter	Power User	Power User	Transactional	External	Modeled

## Create a report using List Pages

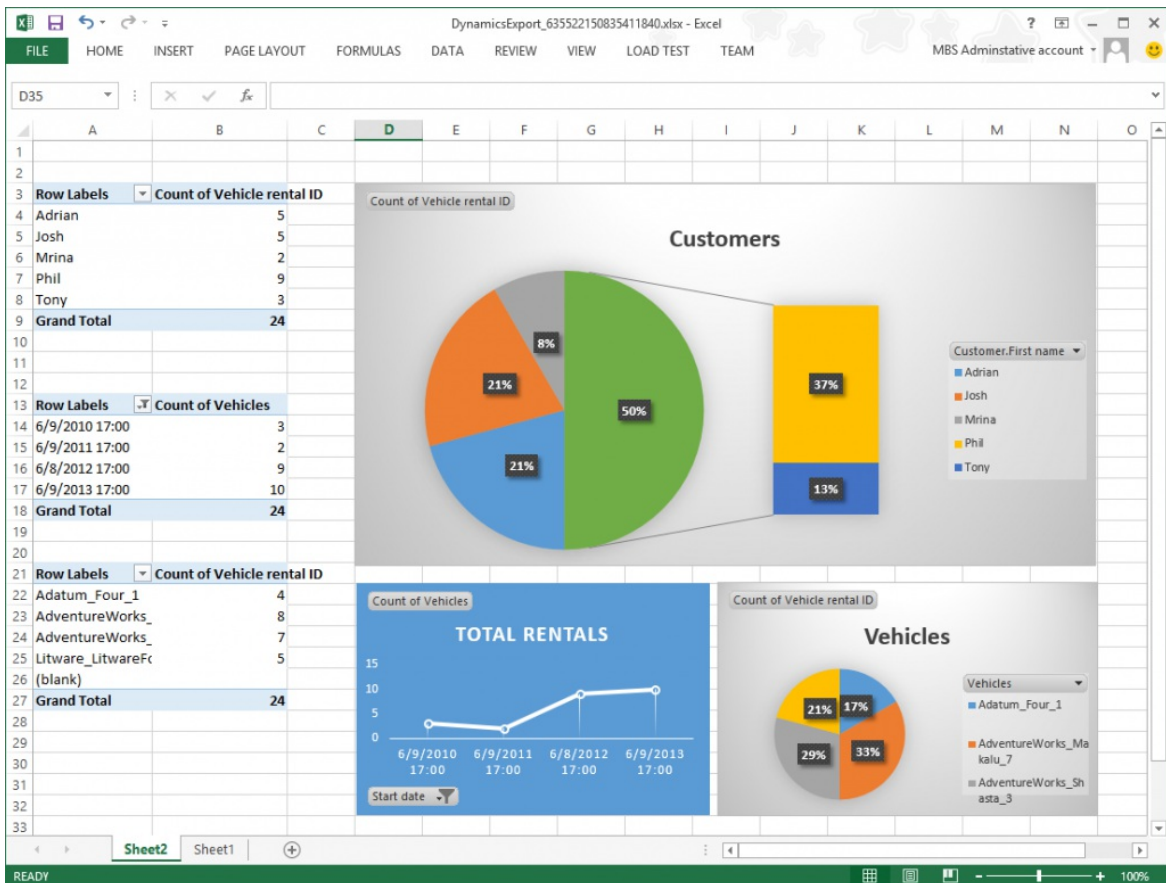
In this section, we'll walk you through the process of exporting data displayed in an entity details form. Here we'll demonstrate how every form in can be viewed as a source of data for management and analysis using the power of Excel.

### Create an analysis report based on all rentals in Fleet Management that are marked as complete

1. Open Internet Explorer, and navigate to your instance base URL and sign in.
  - a. On the cloud environment, the base URL is obtained from LCS
  - b. On a local VM, the base URL is `https://usnconeboxax1aos.cloud.onebox.dynamics.com`.
2. On the Dashboard, scroll to the far right, and click the **Reservation management** tile.
3. Under the **Summary** section, click on the **All rentals** tile.
4. Expand the **Filters** pane by clicking on the Filters icon next to the main grid.
5. Click **Add a filter field**, and then select **Status** in the drop-down list.
6. Enter **Complete** in the Status filter field, and then click **Apply**.
7. Expand the page action bar, click **Open in Office**, and then select **Rentals**.



After the data has been exported, you can use the power and flexibility of the Excel tools to create reports for presentation or additional analysis. The following is an example of a self-service report authored with Excel.



8. Close the browser session and the exported Excel file.

### Advantages of List Pages

- Flexible source for accessing transactional business data using external reporting and analytical tools like Excel
- Allows end-users to model the data sets they need without requiring a developer
- Direct access to real-time business information

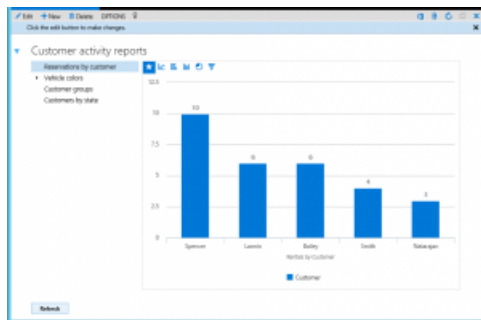
### Expand predefined views and add navigation to charts

Business Intelligence can be useful at every level of an organization. Use embedded controls to elevate the most

relevant information based on the target persona. Native controls offer users an intuitive and convenient way of interacting with aggregate data allowing for informed decision making.

### Create development project

1. On the Desktop, right-click the **Visual Studio** shortcut, and select **Run as administrator** to open the development environment.
2. On the toolbar, click **View**, and then select **Application Explorer**
3. Use the **Application Explorer** to search for the **FMReservationsReport** form in the **Fleet Management** module.
4. In the **Application Explorer's** search results, right-click **FMReservationsReport** form, and then select **Add to new project**.
5. Select the **Finance and Operations Project** template, and then click **OK** to create the project.
6. In **Solution Explorer**, right-click **FMReservationsReport** menu item, and then select **Set as startup object**.
7. Press **Ctrl+Shift+B** to save and build the project.
8. Press **Ctrl+F5** to load the form containing the report.

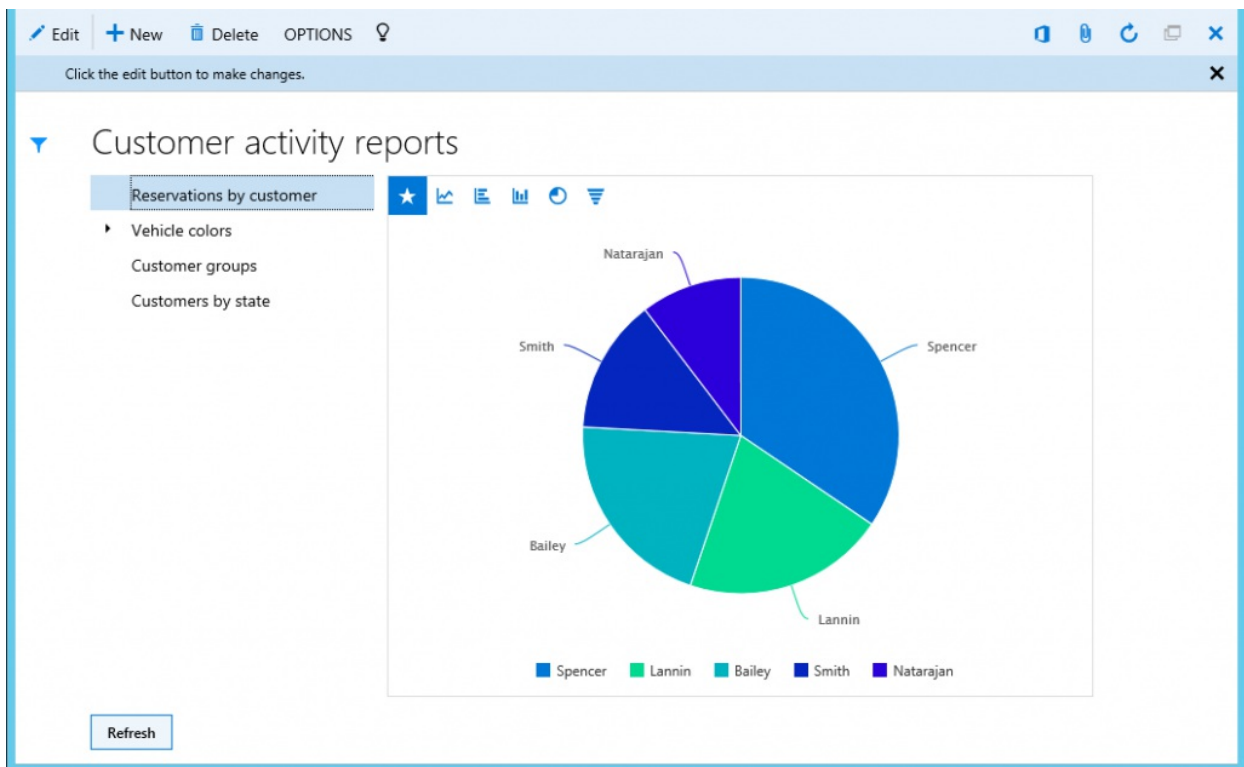


At this point, you have a collection of pre-defined chart visualizations of Aggregate data. The controls offer basic interactive features like hover text, series filtering, and touch expansion. However, it is often appropriate that a greater degree of user interactivity is required.

### Change the default chart type for the visualization

1. Close the browser session and return to the Visual Studio project.
2. Open the **Application Explorer**, locate the **FMReservationsByCustPart** form, right-click, and then select **Add to project**.
3. Open the **FMReservationsByCustPart** form designer.
4. Access the **Series** collection in the **ReservationsByCust** chart definition.
5. Select the **SysBuildChatMeasure1** series definition
6. Locate the **Appearance** section in the **Properties** window.
7. Update the **Chart type** value and select **Pie**
8. Press **Ctrl+Shift+B** to save and rebuild the project.
9. Press **Ctrl+F5** to run the report.

The **Reservations by customer** view now visualizes the aggregate data set using a Pie chart to simplify the task of comparing rental activity across customers.



Additional functions include:

- Define contextual drill-through navigations using modeled properties
- Manage drill-through links using X++ form logic

#### Advantages of embedded aggregate visualizations

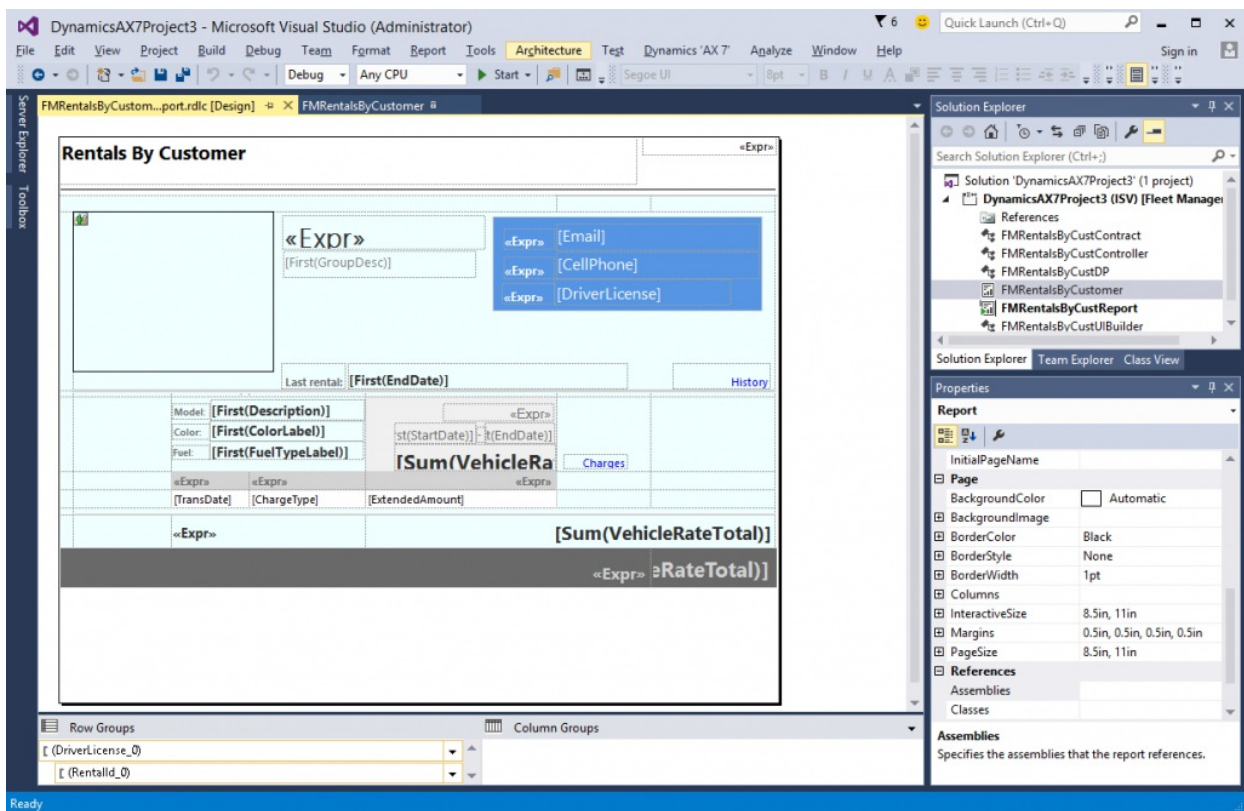
- Designed to be simple and intuitive to promote informed decisions at every level of the organization
- Pre-defined views tailored to meet the needs of the consumer
- Augments the user's ability to recognize key performance trends in the data

## Using the freeform report designer

SSRS continues to be the platform for producing advanced Business Document solutions for your ERP application. The hosted service is designed for high volume and heavily transactional reports with an integrated parameter experience. Backed by the Document Printing & Distribution Services, these solutions are ideal for producing precision documents which are intended for email, printing, archive, and bulk distribution.

#### Open the Precision Designer

1. Close the browser session and return to the Visual Studio project.
2. Click **File > Close Solution**, to close the active project.
3. Use the **Application Explorer** to search for objects with **FMRentalsByCust** in the name
4. In the **Application Explorer's** search results, select all **Classes**, the **Output Menu Item**, and the **FMRentalsByCustomer** report
5. Right-click and then select **Add to new project**.
6. Select the **Dynamics 365** template, and then click **OK** to create the project.
7. In **Solution Explorer**, double-click on the **FMRentalsByCustomer** report to open the designer.
8. Expand the **Designs** collection in the designer to view the list of design definitions
9. Double-click on the **Report** design to view the report definition using the Precision Designer



The above is a screen-shot of the FMRentalsByCustomer report design definition as viewed using the Visual Studio Precision Designer. The Precision Designer offers a free-form design surface with built-in tools that allow you to customize the content and layout of the report. You can also take advantage of embedded VB code to create run-time design manipulations and support user interactions. As an integrated tool, developers are able to reference AX labels and public APIs to format data in the report body based on AX EDTs. MSDN offers a rich collection of developer documentation related to SSRS formatting capabilities. See the article [Reporting Services Reports \(SSRS\)](#) on for a good primer on designing effective SSRS reports.

## Customizing the parameter experience

The Reporting Framework offers flexibility through service extensions to facilitate advanced solutions with requirements that cannot be addressed using a modeled solution. Use the VS designer to add basic parameter formatting, grouping, and input validation. X++ based data contract validation is available for more advanced scenarios. Consider adding User Interface (UI) Builder Classes to customize the parameter pane used to prompt for session inputs before running a report. These custom extensions are effective for addressing the following functions:

- Overriding dialog field events
- Validating report parameters inline
- Adding customized lookups to dialog fields
- Changing the layout of the dialog controls
- Adding custom controls including images

### Defining parameters defaulting using code

1. In **Solution Explorer**, double-click on the **FMRentalsByCustUIBuilder** class to open the designer.
2. Locate the class **build** method and update the initialization code as follows



```

public void build()
{
    Dialog dialogLocal = this.dialog();
    contract = this.dataContractObject() as FMRentalsByCustContract;// associate dialog field with
data contract method
    this.addDialogField(methodStr(FMRentalsByCustContract, parmCustGroupId),
contract);dialogLocal.addGroup("@SYS41297");
    fromDateField = this.addDialogField(methodStr(FMRentalsByCustContract, parmFromDate), contract);
    toDateField = this.addDialogField(methodStr(FMRentalsByCustContract, parmToDate), contract);//
set the default date range values
    fromDateField.value(today() - 365);
    toDateField.value(today());
}

```

3. Press Ctrl+Shift+B to save and rebuild the project.

4. Press Ctrl+F5 to run the report.

The parameter initialization code above sets the default values of the report execution relative to today's date. Use the classes `UIBuilder` to override the framework's default handling of report parameters. Additional extension scenarios supported:

- Automatically set query ranges based on session context using `Controller` classes
- Select report designs at runtime using a shared menu item
- Modify report data contract values using business logic

#### NOTE

This example demonstrates a `UIBuilder` extension with an RDP-based based report. For a Query based example that includes a `UIBuilder` extension, view the `FMCustomerList` report.

## Using VB code to manage running balances

The Reporting Framework offers flexibility through service extensions to facilitate advanced solutions with requirements that cannot be addressed using a modeled solution. Use the VS designer to add basic parameter formatting, grouping, and input validation. X++ based data contract validation is available for more advanced

scenarios. Consider adding User Interface (UI) Builder Classes to customize the parameter pane used to prompt for session inputs before running a report. These custom extensions are effective for addressing the following functions:

- Overriding dialog field events
- Validating report parameters inline
- Adding customized lookups to dialog fields
- Changing the layout of the dialog controls
- Adding custom controls including images

### Import the section resources

1. Close the browser session and return to the Visual Studio project.
2. On the toolbar, click **DYNAMICS AX**, and then select **Import Project...**
3. In the File name window, browse to C:\FmLab\Lab10-3, select **FMrentalDetailsReport.axpp**, and then click **Open**.
4. In the Project file location text box, enter C:\FmLab\Lab10-3.
5. Select the **Current solution** radio button.
6. Click **OK** to close. Wait for the project to be imported and opened.
7. In the **Solution Explorer**, select the new project, right-click, and then select **Set as startup project**.
8. In the **Solution Explorer**, right-click **FMrentalDetails** menu item, and then select **Set as startup object**.
9. Select the Project in Solution Explorer, right-click, and then select **Deploy Reports**
10. Press **Ctrl+F5** to view the report.

**Rental Details**  
Contoso Entertainment System USA

Labels!@SYS182566  
11/17/2015  
2:17 PM

Previous: \$1,575.00

**Tony Smith**

Counter	RentalID	Type	Description	Amount	Running balance
1	000005	Car pickup	2014 Adventure Works Makalu	\$35.00	\$1,610.00
2	000005	Child Seat	2014 Adventure Works Makalu	\$10.00	\$1,620.00
3	000005	Tank refill	2014 Adventure Works Makalu	\$40.00	\$1,660.00
4	000005	Carbon fee	2014 Adventure Works Makalu	\$5.00	\$1,665.00
5	000012	Tank refill	2014 Adatum Four	\$40.00	\$1,705.00
6	000012	Carbon fee	2014 Adatum Four	\$5.00	\$1,710.00
7	000026	Carbon fee	2014 Litware Four	\$5.00	\$1,715.00
8	000026	Tank refill	2014 Litware Four	\$40.00	\$1,755.00
9	000028	Car pickup	2014 Litware Four	\$35.00	\$1,790.00
10	000026	Car pickup	2014 Litware Four	\$35.00	\$1,825.00
11	000028	Carbon fee	2014 Litware Four	\$5.00	\$1,830.00
12	000028	Tank refill	2014 Litware Four	\$40.00	\$1,870.00

Total: \$295.00  
Previous: \$1,575.00  
**Grand total: \$1,870.00**

This report uses embedded VB script to keep track of running totals so that the balances from the previous page can be referenced on the active page. To inspect report's VB code, load the report design in the Precision Designer, access the **Report Properties**, and then select the **Code** section. Here you'll see several functions referenced by the report designs to surface running balances within the report headers and footers.

### Advantages of SSRS reports

- Built-in back office document management capabilities including email support, scheduled executions via Batch, and Print Archive
- Parameterized views with drill-through navigations to forms and other reports
- Used to produce precision documents for compliance with local regulatory business practices

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customize App Suite reports by using extensions

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic discusses a series of scenarios for customizing App Suite reports.

Finance and Operations offers an expanded set of tools to support custom solutions. Customizations to reporting solutions in the standard application are fully supported using a pure extension model. This topic contains guidance about how to add the most common customizations to standard application reports without overlayering Application Suite artifacts. Here are some of the key benefits of using an extension-based approach when customizing the application:

- It reduces the footprint of your application solutions by minimizing code duplication.
- Custom reports benefit from enhancements made to standard solutions including updates to business logic in Report Data Provider (RDP), data contracts, and UI Builder classes.
- Standard application solutions are unaffected and continue to be available in concert with custom reports.

Report extensions do not break or prevent access to standard application reports. Instead, the platform supports run-time selection of the target report allowing you to choose the appropriate report design based on the context of the user session. For more information about customizations using extensions, see [Customize through extension and overlayering](#)

## Scenarios

There are four key scenarios which demonstrate the flexibility available. The first two scenarios involve extending existing RDP classes for custom reporting solutions.

- [Expand Application Suite report data sets](#) – Use table extensions and integrate custom business logic to add custom columns to an existing dataset.
- [Composing custom datasets](#) – Add more data to application reports by extending an existing RDP class to return a custom dataset.

The other two scenarios offer insights on how to use extensions to redirect application navigations to your custom solutions.

- [Extend report menu items to redirect user navigation](#) – Customize application menu items to redirect references to a custom report design.
- [Create custom designs for business documents](#) – Delegate handlers allow you to add custom report designs to an existing Print Management document instance.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

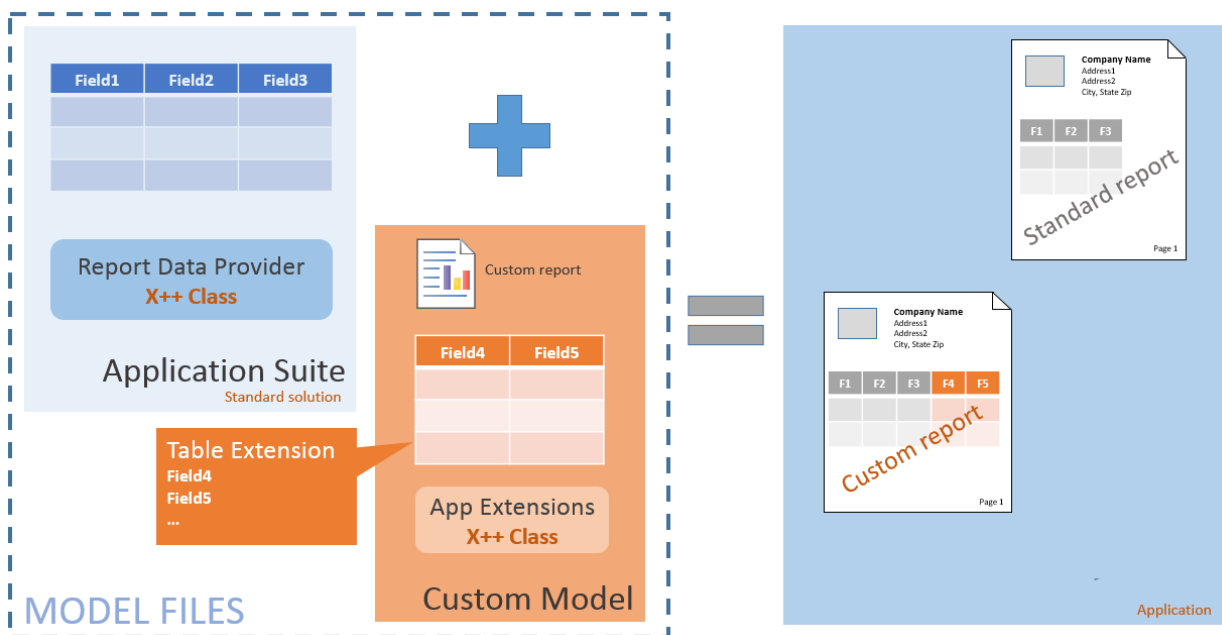
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Expand Application Suite report data sets

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic shows how to expand an existing report data set that is produced by using X++ business logic in a report data provider (RDP) class.

This topic focuses on the expansion of an existing report data set that is produced by using X++ business logic in a report data provider (RDP) class. You use custom delegate handlers and table extensions to include additional field data and/or calculations. You don't have to over-layer the Application Suite. You then create custom designs that replace the standard application solutions and present the data to users. The following illustration shows a typical application customization, as described in this topic.



## What's important to know?

There are a few basic assumptions that you should be aware of before you apply this solution.

- **You can't directly extend RDP classes.** However, the platform provides extension points that enable data set expansion without duplicating business logic in the standard application.
- **There are two methods that can be used to expand report data sets.** Use the strategy that is appropriate for your solution:
  - **Data processing post-handler** – This method is called only one time, after the **ProcessReport** method is completed and before the data set is returned to the report server. Register for this post-handler to perform bulk updates on the temporary data set that is produced by the standard application solution.
  - **Temp table inserting event** – This method is called for each row that is added to the temporary table. It's more suitable for calculations and inline evaluations. Try to avoid expensive queries that have many joins and look-up operations.
- **Use event handlers to redirect menu items to your new report design.** You can customize all aspects of an application reporting solution by using event handlers. Add a **PostHandler** event for the controller class to reroute user navigations to a custom report design.

# Expand a report data set

The following walkthrough shows the process of expanding an existing application data set by using a "pure" extension-based solution. The solution includes a custom **Rentals list** report for the Fleet Management application. The new report includes additional rental charge data in the rental details. The application customizations are defined in an extension model. The following illustrations show the standard design and the custom solution.

## Before (standard design)

The screenshot shows the 'Rentals By Customer' report for Phil Spencer. The report displays two rental records. The first record is for a 2014 Litware Four (Red, Standard fuel) with a total charge of \$480.00. The second record is for a 2012 Adventure Works Shasta (Gray) with a total charge of \$240.00. The charges table for the first rental includes Carbon fee (\$5.00), Tank refill (\$40.00), and Car pickup (\$35.00).

Date	Rental charge type	Extended amount
12/24/2010	Carbon fee	\$5.00
12/24/2010	Tank refill	\$40.00
	Car pickup	\$35.00

## After (custom solution)

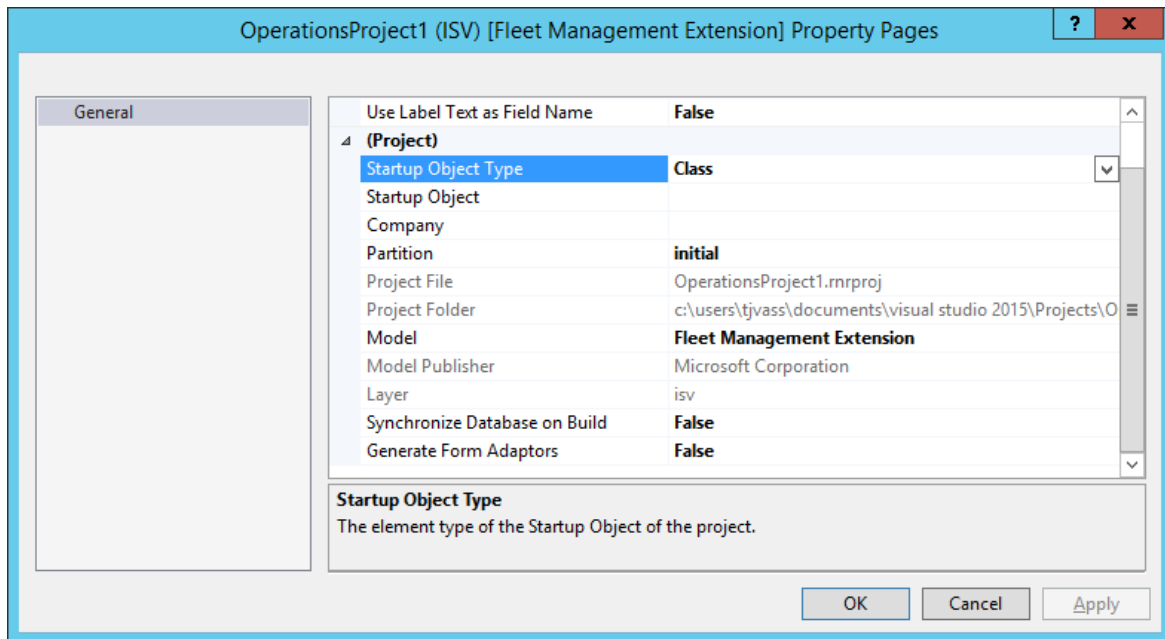
The screenshot shows the custom 'Rentals By Customer (CUSTOM)' report for Phil Spencer. The report displays the same two rental records as the standard design. The charges table for the first rental includes Carbon fee (\$5.00), Tank refill (\$40.00), and Car pickup (\$35.00). The custom solution adds additional details to the charges table, such as 'Help offset carbon emissions.' and 'We refill the gas tank.'.

Date	Rental charge type	Extended amount
12/24/2010	Carbon fee	\$5.00
	Help offset carbon emissions.	
12/24/2010	Tank refill	\$40.00
	We refill the gas tank.	
	Car pickup	\$35.00
	We arrange pickup from another location.	

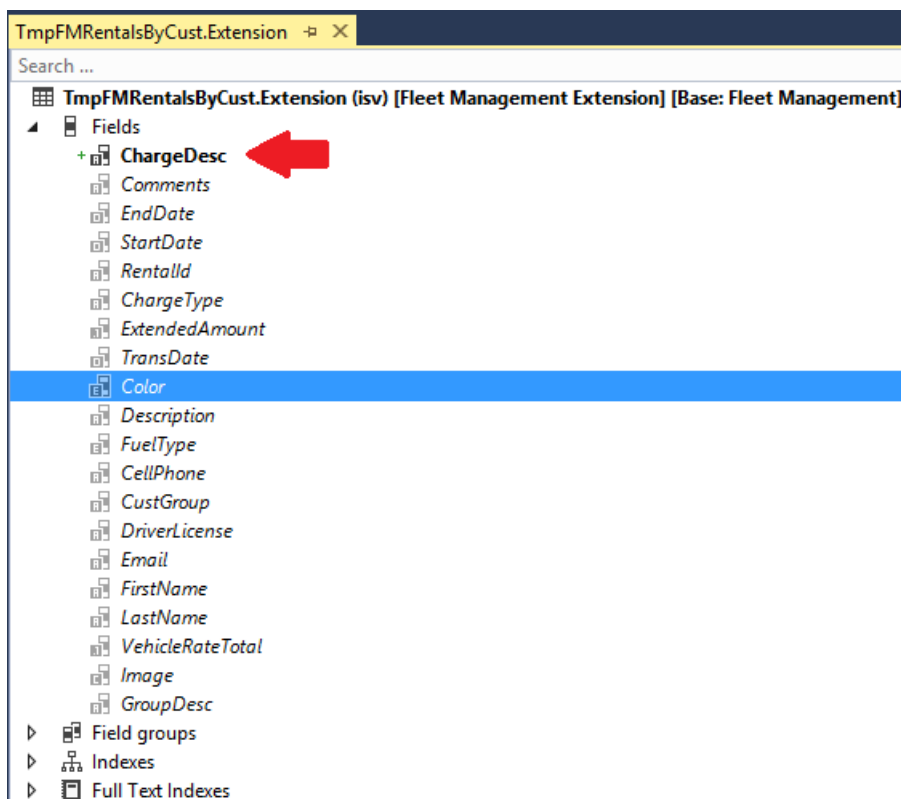
1. Create a new model for your application customizations. For more information about extension models, see [Customize through extension and overlaying](#). For this example, add a custom report to the

## Fleet Management Extensions model.

2. Create a new project in Microsoft Visual Studio. Make sure that the project is associated with your extension model. The following illustration shows the project settings.



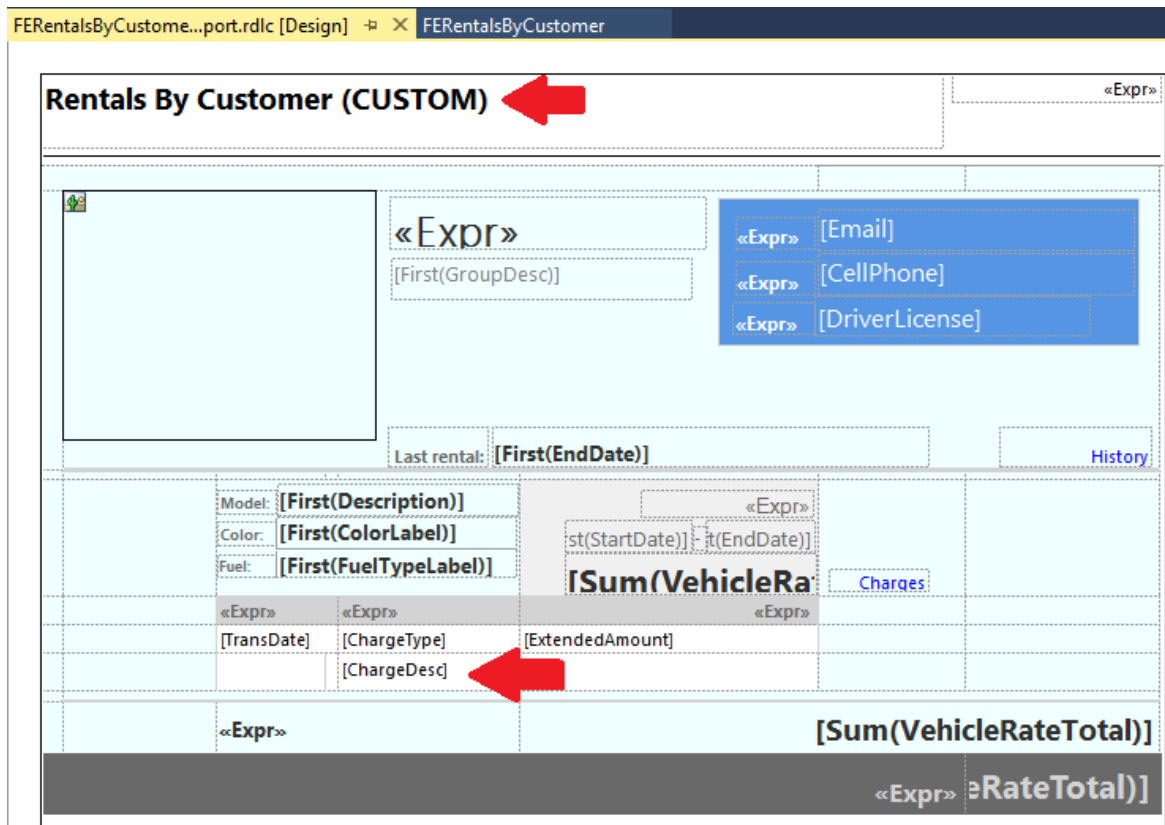
3. Add a table extension to store the custom report data. Find the temporary cache for the TmpFMRentalsByCust data set that is populated by the RDP class, and create an extension in your model. Define the fields that will be used to store the data for the report server, and then click **Save** to save your changes. The following illustration shows the table extension that is required for this example.



4. Add your custom report to the project. The custom design closely resembles the standard solution. Therefore, you can just duplicate the existing application report in the **Fleet Management Extension** model, and then update the report design so that it includes the custom title and additional text box in the Rental Charges container.
5. Rename the report so that it has a meaningful name. For this example, rename the custom report

FERentalsByCustomer to distinguish it from the standard solution.

6. **Restore the report data set references.** Open the report designer, expand the **Datasets** collection, right-click the data set that is named **FMRentalsByCustDS**, and then click **Restore**. The data set is expanded so that it includes the newly introduced columns. Therefore, these columns are now available in the report designer.
7. **Customize the report design.** The designer offers a free-form design surface that you can use to create the custom solution. The following illustration shows the custom design that is used for this example.



8. **Add a new report handler (X++) class to the project.** Give the class a name that appropriately describes that it's a handler for an existing application report. For this example, rename the class **FERentalsByCustomerHandler** to distinguish it from other report handlers.
9. **Add a PostHandler method to begin to use your custom report.** In this example, extend the controller class in the standard solution, **FMRentalsByCustController**, by using the following code.

```
class FERentalsByCustomerHandler
{
    [PostHandlerFor(classStr(FMRentalsByCustController), staticMethodStr(FMRentalsByCustController,
construct))]
    public static void ReportNamePostHandler(XppPrePostArgs arguments)
    {
        FMRentalsByCustController controller = arguments.getReturnValue();
        controller.parmReportName(ssrsreportstr(FERentalsByCustomer, Report));
    }
}
```

User navigations in the application will now be rerouted to the custom reporting solution. Take some time to deploy the custom report to the report server and verify that the application is using it. At this point, you just have to add the business logic that is used to populate the custom fields that you introduced in step 3. In the next step, you must select the method of data set expansion that is appropriate for your solution.



10. **Add X++ business logic to populate the custom field data.** Select the data processing technique that makes sense for the type of transformation that you require for the solution.

- **Option 1: Add a data processing post-handler.** Apply this technique for bulk insert operations that use a single pass over the result set of the standard solution. Here is the code that expands the data set by using a table lookup.

```
class FERentalsByCustomerHandler
{
    [PostHandlerFor(classStr(FMRentalsByCustDP), methodstr(FMRentalsByCustDP, processReport))]
    public static void TmpTablePostHandler(XppPrePostArgs arguments)
    {
        FMRentalsByCustDP dpInstance = arguments.getThis() as FMRentalsByCustDP;
        TmpFMRentalsByCust tmpTable = dpInstance.getTmpFMRentalsByCust();
        FMRentalCharge chargeTable;
        ttsbegin;
        while select forUpdate tmpTable
        {
            select * from chargeTable where chargeTable.RentalId == tmpTable.RentalId;
            tmpTable.ChargeDesc = chargeTable.Description;
            tmpTable.update();
        }
        ttscommit;
    }
}
```

- **Option 2: Add a temp table Inserting event.** Apply this technique for row-by-row calculations. Here is the code that expands the data set by using a table lookup.

```
class FERentalsByCustomerHandler
{
    [DataEventHandlerAttribute(tableStr(TmpFMRentalsByCust), DataEventType::Inserting)]
    public static void TmpFMRentalsByCustInsertEvent(Common c, DataEventArgs e)
    {
        TmpFMRentalsByCust tmpTable = c;
        FMRentalCharge chargeTable;
        // update the value of the 'ChargeDesc' column during 'insert' operation
        select * from chargeTable where chargeTable.RentalId == tmpTable.RentalId
        && chargeTable.ChargeType == tmpTable.ChargeType;
        tmpTable.ChargeDesc = chargeTable.Description;
    }
}
```

You've now finished expanding the report data set. After the application is compiled, it will begin to reroute user navigations to the new report design by using the custom X++ business logic that you defined in the report class handler that is defined in the extension model.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

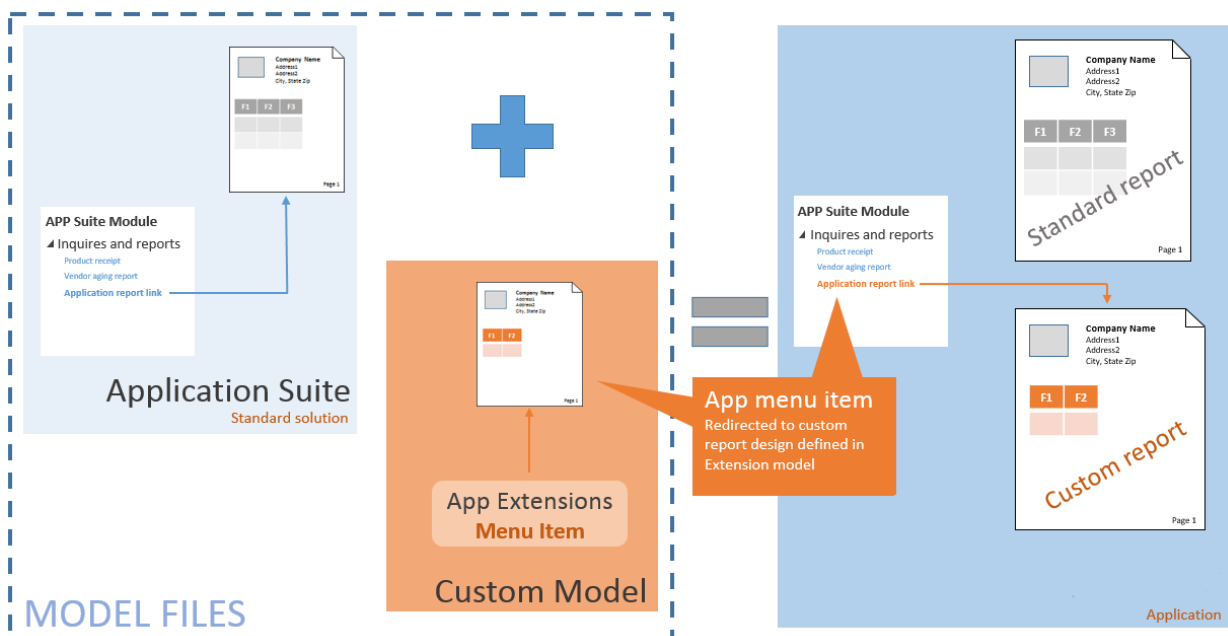
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Extend report menu items to redirect user navigation

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic shows how to extend existing application menu items so that, after only minimal code changes, navigations are redirected to a custom reporting solution.

This topic focuses on the process of extending existing application menu items so that, after only minimal code changes, navigations are redirected to a custom reporting solution. By using this technique, you will avoid the inconvenience of tracking down and replacing all references to an existing application report. Just extend an existing application menu item to redirect application navigations to reports that are defined in an extension model. The following illustration shows a typical application customization.



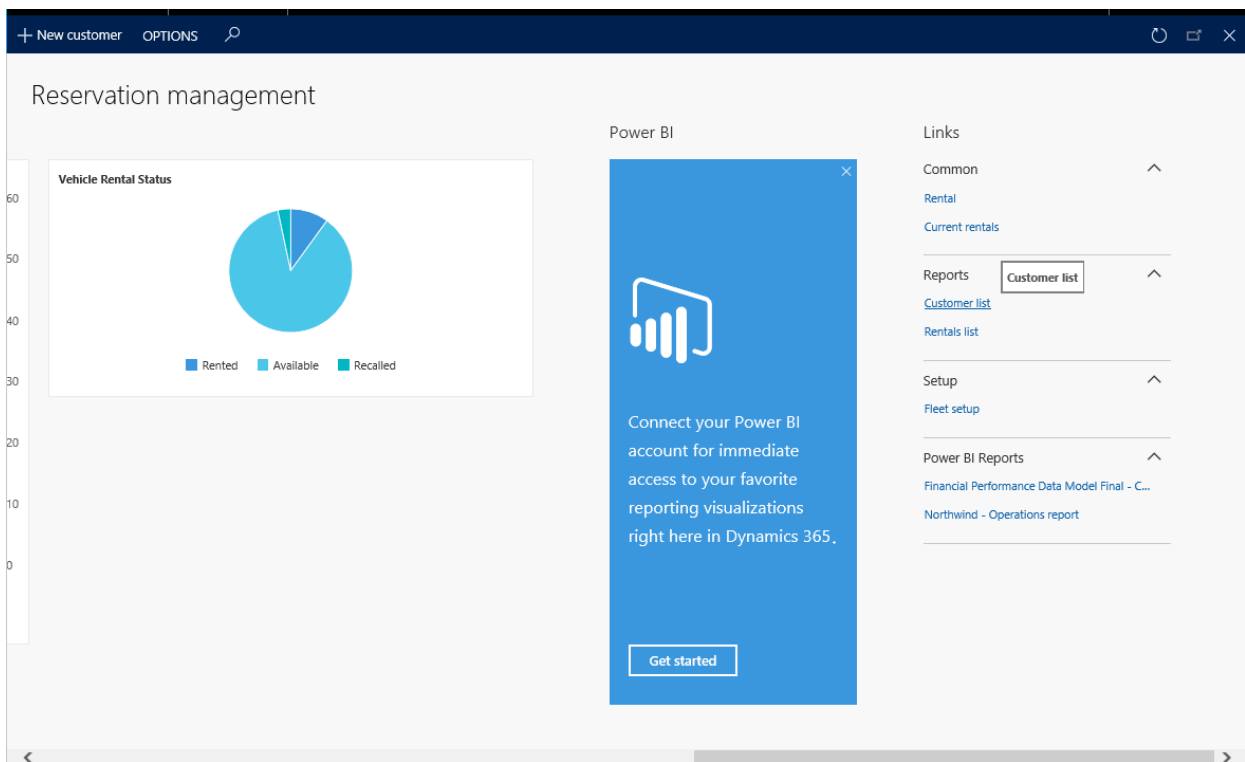
## What's important to know?

There are a few basic assumptions that you should be aware of before you apply this solution.

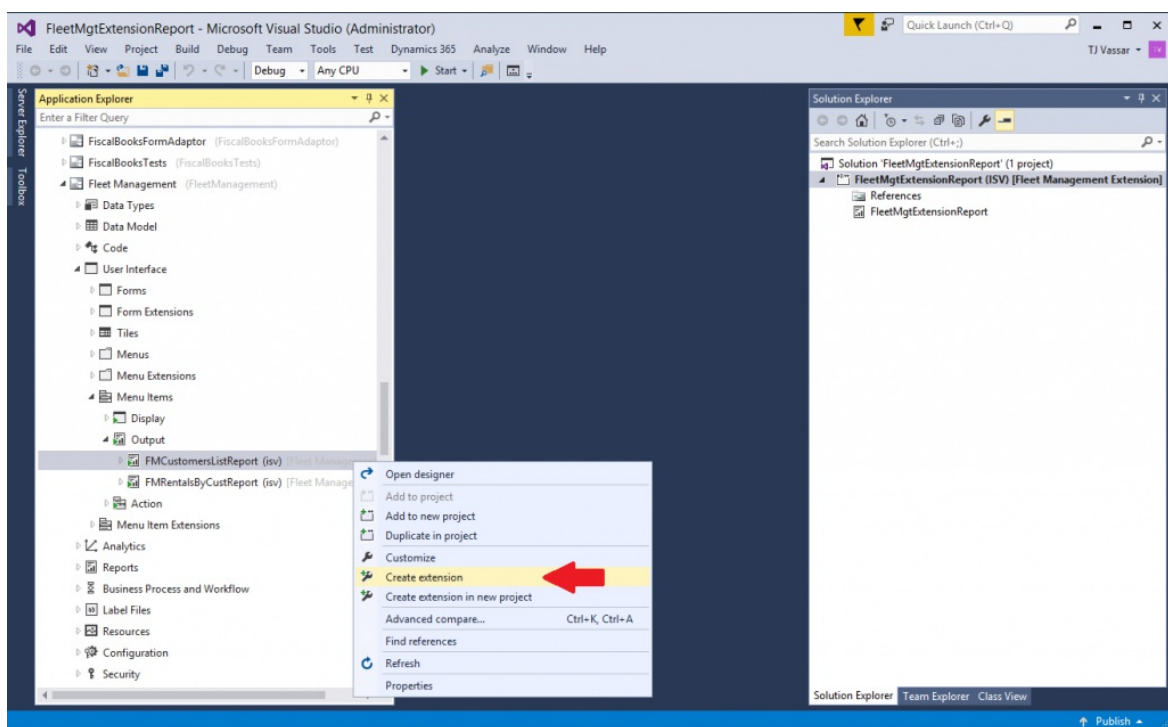
- Extended menu items let you override both the display string and the target.
- This technique can be used for all types of reports, from simple query-based reports to complex report data provider (RDP)-based reports.
- Extended menu items are available for direct references to reports and solutions that orchestrate the reporting session by using a controller class.

## Extend report menu items

The following walkthrough shows how to use menu item extensions to redirect user navigations in the application to a custom solution. The solution includes a custom **Customer list** report for the Fleet Management application and defines all the application customizations in a pure extension model. The following illustration shows the menu item that you use to access the custom **Customer list** report.



1. Create a new model for your application customizations. For more information about extension models, see [Customize through extension and overlaying](#).
2. Create a new project in Microsoft Visual Studio, and add your custom report. Additionally, add all the solution artifacts. These artifacts include the RDP class or source query, the controller class, and UI builders, if they are present.
3. Create an extension of the menu item that is used to access the report. In this example, the output menu item is named `FMCustomerListReport`. Use the menu item structure to find the menu item name that is exposed in the application. The following illustration shows the action in Application Explorer.



4. Modify the properties of the menu item extension. Update the report design or controller reference in the menu item to direct navigations to your custom solution.

**NOTE**

The property changes that you can make on the object depend on the original application solution. If the application report manages the solution by using a controller, a controller class is required for the report.

**5. Rebuild the solution, and deploy the custom report.**

You've now finished extending the report menu item. Navigations to the standard menu item will now be redirected to your custom reporting solution.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

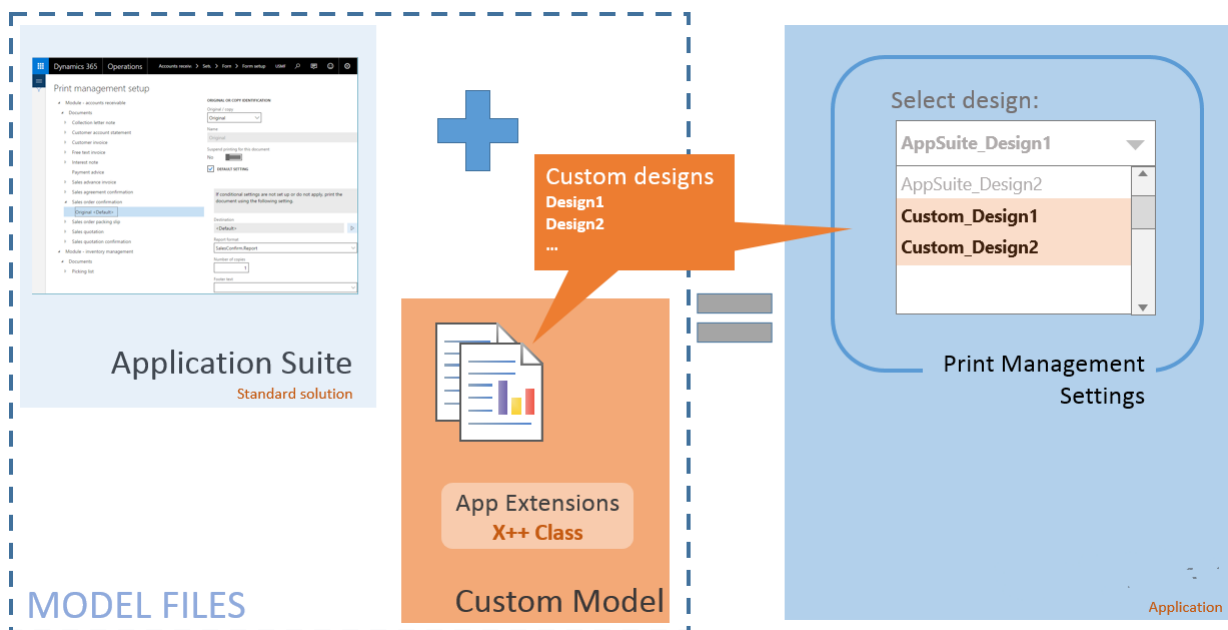
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create custom designs for business documents

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic shows how to create a custom report design for an existing application business document by using a pure extension model.

Microsoft Dynamics 365 Finance includes an expanded set of tools to support custom solutions. This topic focuses on the steps for creating a custom report design for an existing application business document by using a pure extension model. Follow the steps later in this topic to associate a custom report design with an instance of an application document. When you've finished, users can configure Print management settings to select the custom design whenever it's appropriate, based on transaction details. The following illustration shows a typical application customization.



## What's important to know?

Here are some important points that you should be aware of before you apply this solution:

- Print management settings are scoped to the active legal entity. Custom designs can be associated with one or more Print management settings.
- Standard report designs continue to be available alongside custom solutions. Use Print management settings to select the appropriate design, based on transaction details.
- If you introduce a business document for a custom business process, more work is required. For more information about how to create a custom business document solution, see the [Print Management Integration Guide](#).

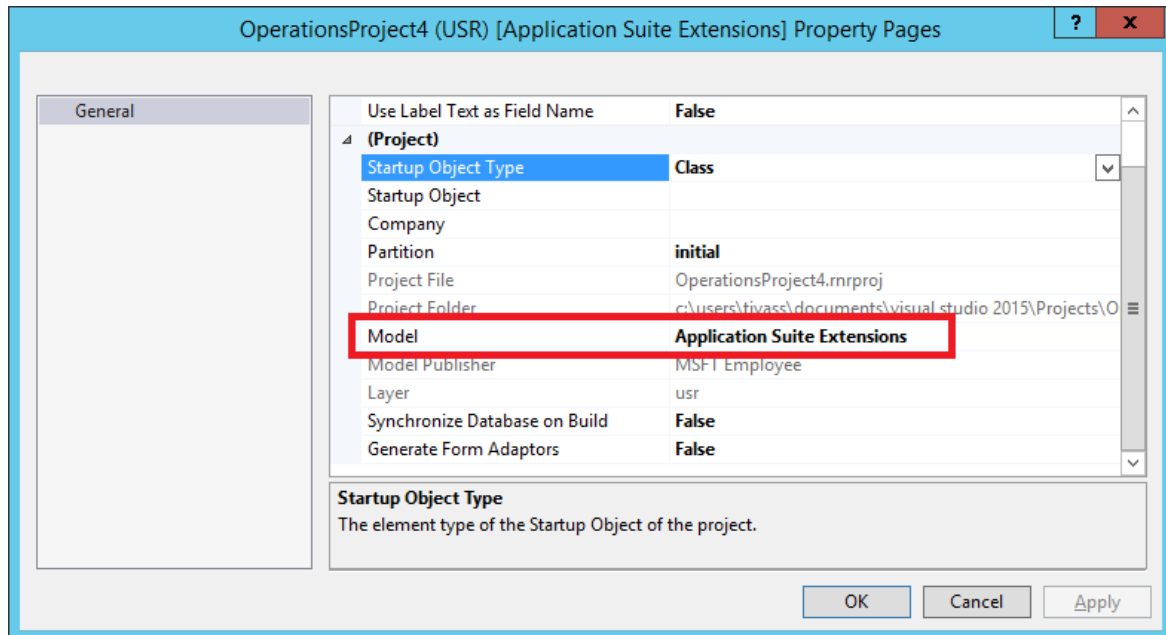
## Customize a business document

The following walkthrough shows the process of introducing a custom report design for an existing application business document and then using Print management to select the new design. The solution includes a custom design definition for the **Sales confirmation** report that is provided in the standard application as part of the Application Suite model. The application customizations will be defined in an extension model.

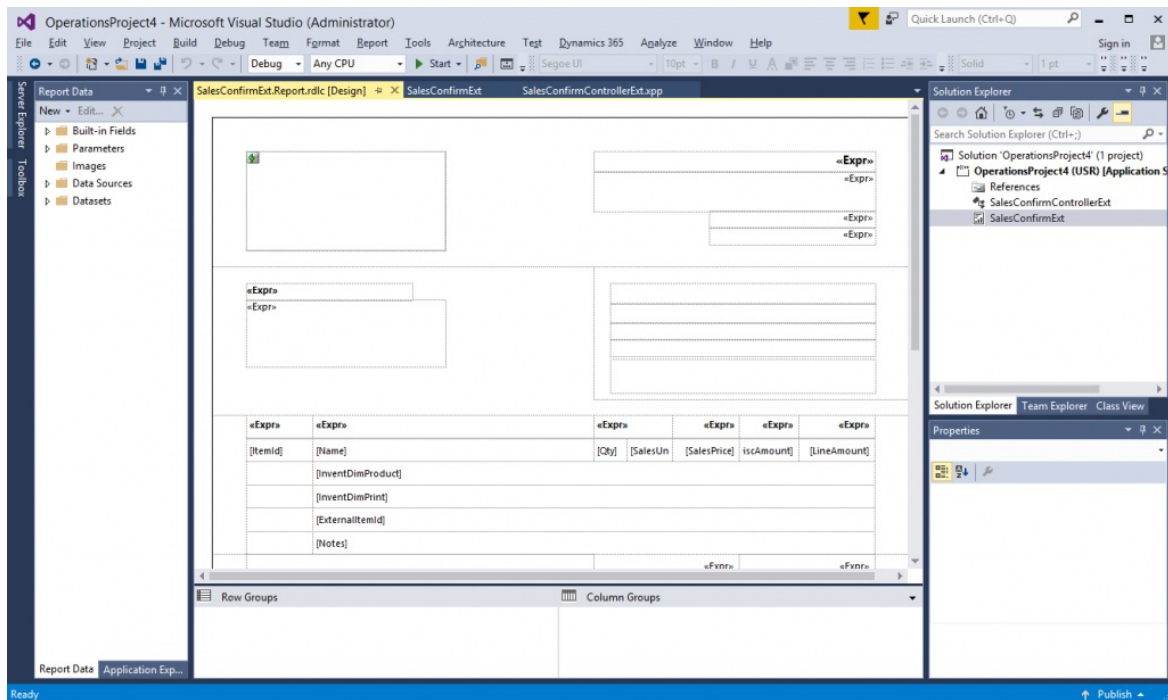
1. **Create a new model for your application customizations.** For more information about extension

models, see [Customize through extension and overlaying](#). For this example, you add a model that is named **Application Suite Extensions**, and that references the Application Suite, Application Platform, and Application Foundation packages.

2. **Create a new project in Microsoft Visual Studio.** Make sure that the project is associated with your extension model. The following illustration shows the project settings.



3. **Create a custom report design for the business document.** You must make sure that your custom solution consumes the correct report data contract. Find the existing Application Suite report in Application Explorer. This report is named **SalesConfirm**. Right-click it, and then click **Duplicate in project** to create the custom solution.
4. **Rename the report so that it has a meaningful name.** For this example, name the custom report **SalesConfirmExt** to distinguish it from the standard solution. Compile the project, and deploy the report to verify that the changes have no errors.
5. **Use the free-form designer to customize the report design.** Select the report design that is named **Report**, right-click it, and open the precision designer. Customize the design to satisfy the organization's business requirements. The following illustration shows a custom design definition for the **Sales confirmation** report.



6. Add a new X++ class that extends the standard report controller. Give the class a name that appropriately describes that it's a handler for an existing application report. For this example, rename the class `SalesConfirmControllerExt` to distinguish it from other report controllers.
7. Use the extended class to load the custom design. Add a `main` method that refers to the custom report design. (You can just copy the `main` method from the standard solution and add references to the new `Controller` class.) Here is the code that extends the standard solution.

```

class SalesConfirmControllerExt extends SalesConfirmController
{
    public static SalesConfirmControllerExt construct()
    {
        return new SalesConfirmControllerExt();
    }
    public static void main(Args _args)
    {
        SrsReportRunController formLetterController = SalesConfirmControllerExt::construct();
        SalesConfirmControllerExt controller = formLetterController; controller.initArgs(_args,
        srsReportStr(SalesConfirmExt, Report));
        if (classIdGet(_args.caller()) == classNum(SalesConfirmJournalPrint))
        {
            formLetterController.renderingCompleted +=
            eventhandler(SalesConfirmJournalPrint::renderingCompleted);
        }
        formLetterController.startOperation();
    }
}

```

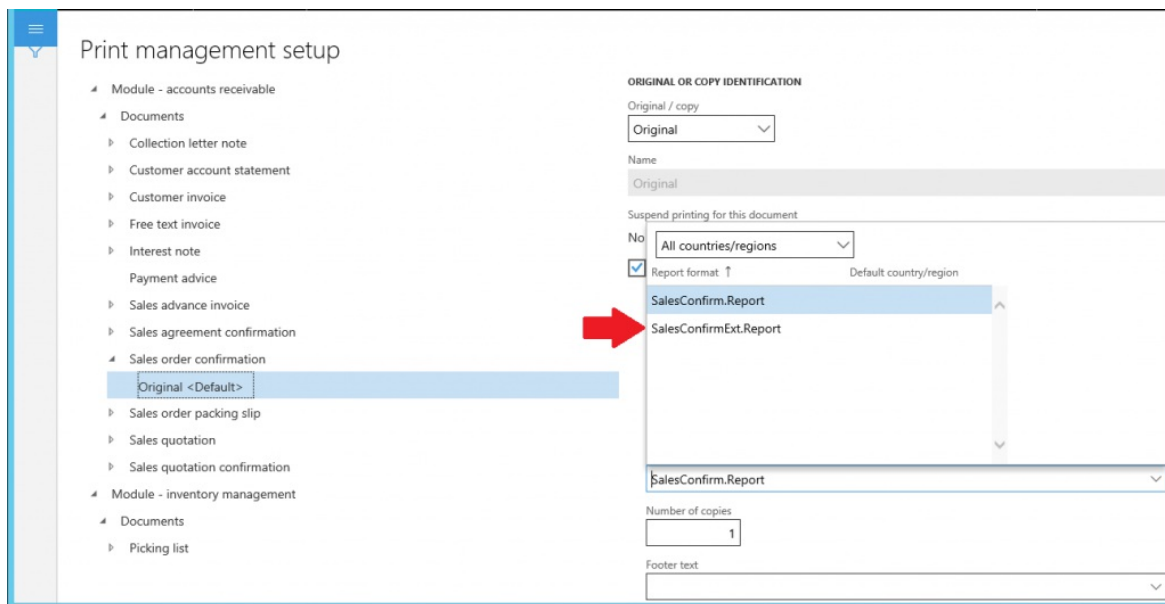
8. Add a new report handler (X++) class to the project. Give the class a name that appropriately describes that it's a handler for Print management–based documents. For this example, rename the class `PrintMgtDocTypeHandlerExt` to distinguish it from other object handlers.
9. Add a delegate handler method to start to use your custom report. In this example, extend the `getDefaultReportFormatDelegate` method in the `PrintMgtDocTypeHandlerExt` class by using the following code.

```

class PrintMgtDocTypeHandlersExt
{
    [SubscribesTo(classstr(PrintMgmtDocType), delegatestr(PrintMgmtDocType,
getDefaultReportFormatDelegate))]
    public static void getDefaultReportFormatDelegate(PrintMgmtDocumentType _docType,
EventHandlerResult _result)
    {
        switch (_docType)
        {
            case PrintMgmtDocumentType::SalesOrderConfirmation:
                _result.result(ssrsReportStr(SalesConfirmExt, Report));
                break;
        }
    }
}

```

10. **Extend the menu item for the application report.** Find the existing Application Suite menu item in Application Explorer. This menu item is named **SalesConfirmation**. Right-click it, and then click **Create extension**. Open the new extension object in the designer, and set the value of the **Object** property to **SalesConfirmControllerExt** to redirect user navigations to the extended solution.
11. **Update the Print management settings to use the custom business document.** For this example, go to **Accounts receivable > Setup > Forms > Form setup**. Click **Print Management**, find the document configuration settings, and then select the custom design. The following illustration shows the Print management settings after the changes have been compiled.



You've now finished customizing the business document. Users will now be presented with the custom report design for the business document when they process transactions in the application.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Help prevent long-running reports from timing out

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides tips that can help you prevent reports that run for a long time from timing out.

Paginated reports and documents are generated by using Microsoft SQL Server Reporting Services. Reporting Services retrieves report data from Application Object Server (AOS) by using a custom extension that uses Windows Communication Foundation (WCF) to communicate with AOS. The size of the data set and the complexity of the report that is generated can affect the time that is required to display the report. Additionally, if various time-outs and other thresholds are reached, report generation might fail. Service time-outs in deployments are fixed and limit interactive connections to 10 minutes. Any data set generation process that exceeds this service time-out limit won't be completed. This article describes the extensions that support long-running reports. These extensions help guarantee that long-running reports can be generated even if the process exceeds the 10-minute service time-out limit.

## Preprocess the data source

If the report uses the Report Data Provider (RDP) to retrieve data, the report should be modified to use a pre-processed RDP class as the data source. In this way, processing logic is invoked before a call is made to Reporting Services. For more information about RDP classes, see [Using Report Data Provider Classes to Access Report Data](#) and [Report Programming Guide](#).

### When is this necessary?

#### SYMPTOMS

Reports are failing due to timeouts

"A connection attempt failed because the connected party did not properly respond after a period of time...."

#### CAUSE

Default timeout only **10 minutes**

#### RESOLUTION

Pre-process the report data sets

## How do I migrate a regular RDP to a pre-process data access solution by using TempDB?

1. Change the RDP base class from `SRSReportDataProviderBase` to `SRSReportDataProviderPreProcessTempDB`.
2. Update the table type from `InMemory` to `TempDB`.
3. Rebuild the report's RDP class.
4. Restore the data source that is linked to the RDP class in the report designer.

5. Redeploy the report.
6. Introduce a **Controller** class to run the report.
7. Update the Output Menu Item so that it points to the **Controller** class instead of the report.

## Use batch processing

To improve performance when you print statements or reports that include large amounts of data, use batch processing. When you use batch processing, you can run specific tasks as batch jobs and then schedule those batch jobs to run on a different computer (a batch server). By moving the processing of these tasks to a batch server, you can improve the report performance on the client computer. You can also apply range restrictions to limit the size of each batch.

To further improve performance, don't submit one large batch. Instead submit multiple smaller batches for processing at the same time on different servers. Many tasks can be run as part of batch jobs. For more information, see [Batch processing overview](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Electronic reporting (ER) overview

2/18/2021 • 20 minutes to read • [Edit Online](#)

This topic provides an overview of the Electronic reporting (ER) tool. It includes information about key concepts, the scenarios that ER supports, and a list of formats that have been designed and released as part of the solution.

ER is a tool that you can use to configure formats for both incoming and outgoing electronic documents in accordance with the legal requirements of various countries/regions. ER lets you manage these formats during their lifecycle. For example, you can adopt new regulatory requirements, and generate business documents in the required format to electronically exchange information with government bodies, banks, and other parties.

The ER engine is targeted at business users instead of developers. Because you configure formats instead of code, the processes for creating and adjusting formats for electronic documents are faster and easier.

ER currently supports the TEXT, XML, Microsoft Word document, and OPENXML worksheet formats. However, an extension interface provides support for additional formats.

## Capabilities

The ER engine has the following capabilities:

- It represents a single shared tool for electronic reporting in different domains and replaces more than 20 different engines that do some type of electronic reporting for Finance and Operations.
- It makes a report's format insulated from the current implementation. In other words, the format is applicable for different versions.
- It supports the creation of a custom format that is based on an original format. It also includes capabilities for automatically upgrading the customized format when the original format is changed because of localization/customization requirements.
- It becomes the primary standard tool to support localization requirements in electronic reporting, both for Microsoft and for Microsoft partners.
- It supports the ability to distribute formats to partners and customers through Microsoft Dynamics Lifecycle Services (LCS).

## Key concepts

### Components

ER supports two types of components: **Data model** and **Format**.

#### **Data model and model mapping components**

A data model component is an abstract representation of a data structure. It's used to describe a specific business domain area with enough detail to satisfy the reporting requirements for that domain. A data model component consists of the following parts:

- A data model, as a set of domain-specific business entities and a hierarchically structured definition of relations between those entities.
- A model mapping that links selected application data sources to individual elements of a data model that specifies, at run time, the data flow and rules of business data population to a data model component.

A business entity of a data model is represented as a container (record). Business entity properties are represented as data items (fields). Each data item has a unique name, label, description, and value. The value of

each data item can be designed so that it's recognized as a string, integer, real, date, enumeration, Boolean, and so on. Additionally, it can be another record or records list.

A single data model component can contain several hierarchies of domain-specific business entities. It can also contain model mappings that support a report-specific data flow at run time. The hierarchies are differentiated by a single record that has been selected as a root for model mapping. For example, the data model of the payment domain area might support the following mappings:

- Company > Vendor > Payment transactions of the AP domain
- Customer > Company > Payment transactions of the AR domain

Note that business entities such as company and payment transactions are designed one time. Different mappings then reuse them.

A model mapping that supports outgoing electronic documents has the following capabilities:

- It can use different data types as data sources for a data model. For example, it can use tables, data entities, methods, or enums.
- It supports user input parameters that can be defined as data sources for a data model when some data must be specified at run time.
- It supports the transformation of data into required groups. It also lets you filter, sort, and sum data, and append logical calculated fields that are designed through formulas that resemble Microsoft Excel formulas. For more information, see [Formula designer in Electronic reporting \(ER\)](#).

A model mapping that supports incoming electronic documents has the following capabilities:

- It can use different updatable data elements as targets. These data elements include tables, data entities, and views. The data can be updated by using the data from incoming electronic documents. Multiple targets can be used in a single model mapping.
- It supports user input parameters that can be defined as data sources for a data model when some data must be specified at run time.

A data model component is designed for each business domain that should be used as a unified data source for reporting that isolates reporting from the physical implementation of data sources. It represents domain-specific business concepts and functionalities in a form that makes a reporting format's initial design and further maintenance more efficient.

#### **Format components for outgoing electronic documents**

A format component is the scheme of the reporting output that will be generated at run time. A scheme consists of the following elements:

- A format that defines the structure and content of the outgoing electronic document that is generated at run time.
- Data sources, as a set of user input parameters and a domain-specific data model that uses a selected model mapping.
- A format mapping, as a set of bindings of format data sources that have individual elements of a format that specify, at run time, the data flow and rules for format output generation.
- A format validation, as a set of configurable rules that control report generation at run time, depending on the running context. For example, there might be a rule that stops output generation of a vendor's payments and throws an exception when specific attributes of the selected vendor are missing, such as the bank account number.

A format component supports the following functions:

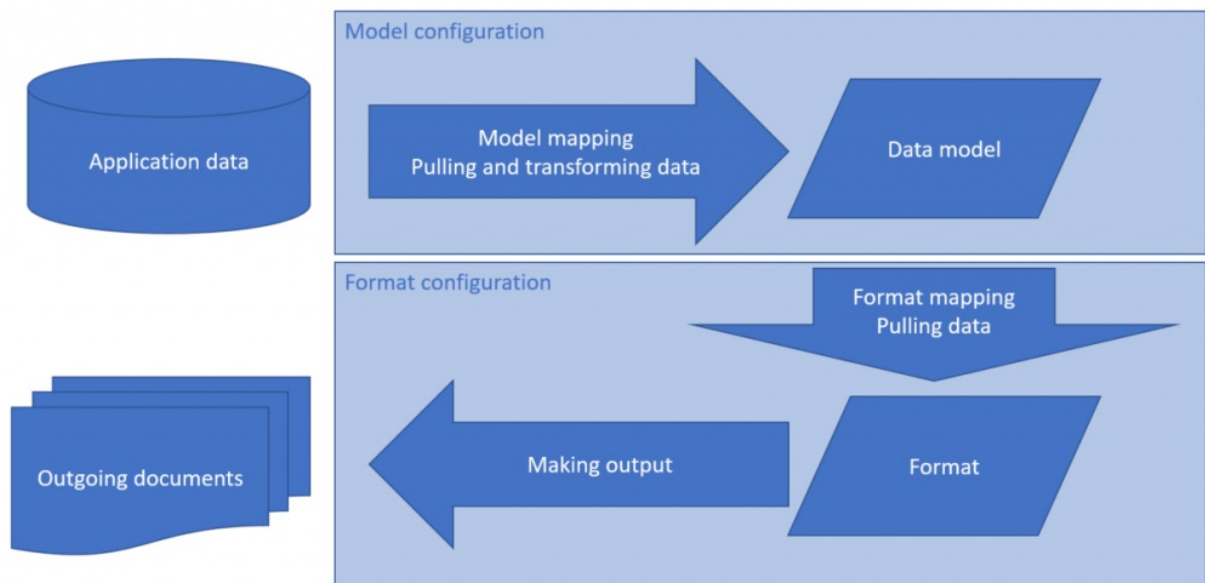
- Creation of reporting output as individual files in various formats, such as text, XML, Microsoft Word document, or worksheet.

- Creation of multiple files separately and encapsulation of those files into zip files.

A format component lets you attach specific files that can be used in the reporting output:

- Excel workbooks that contain a worksheet that can be used as a template for output in the OPENXML worksheet format
- Word files that contain a document that can be used as a template for output in the Microsoft Word document format
- Other files that can be incorporated into the format's output as predefined files

The following illustration shows how the data flows for these formats.



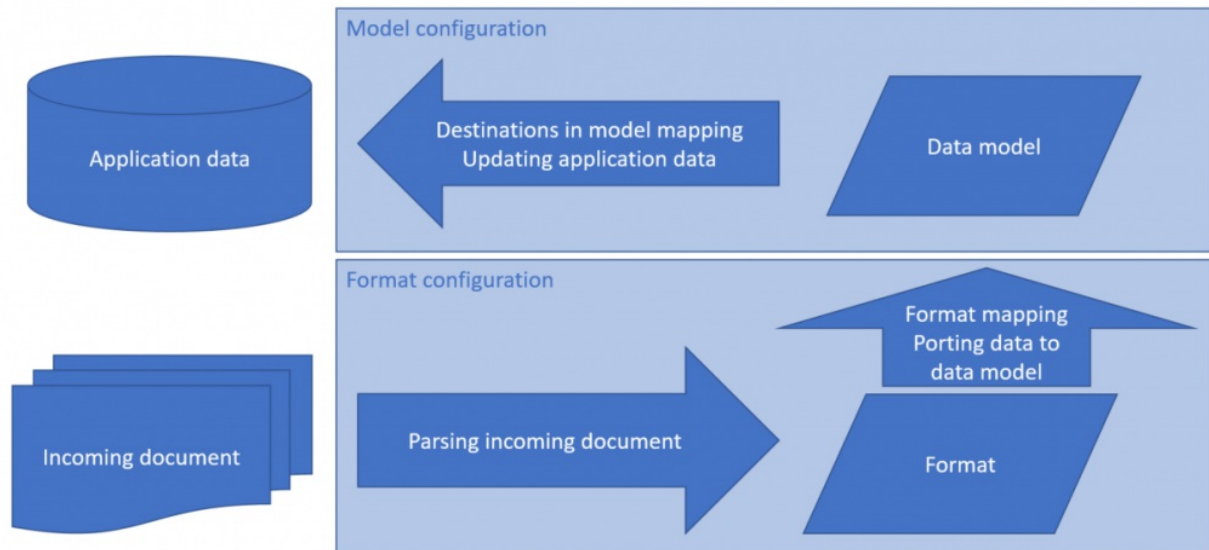
To run a single ER format configuration and generate an outgoing electronic document, you must identify the mapping of the format configuration.

#### **Format components for incoming electronic documents**

A format component is the scheme of the incoming document that is imported at run time. A scheme consists of the following elements:

- A format that defines the structure and content of the incoming electronic document that contains data that is imported at run time. A format component is used to parse an incoming document in various formats, such as text and XML.
- A format mapping that binds individual format elements to elements of a domain-specific data model. At run time, the elements in the data model specify the data flow and the rules for importing data from an incoming document, and then store the data in a data model.
- A format validation, as a set of configurable rules that control data import at run time, depending on the running context. For example, there might be a rule that stops data import of a bank statement that has a vendor's payments and throws an exception when a specific vendor's attributes are missing, such as the vendor identification code.

The following illustration shows how the data flows for these formats.



To run a single ER format configuration to import data from an incoming electronic document, you must identify the desired mapping of a format configuration, and also the integration point of a model mapping. You can use the same model mapping and destinations together with different formats for different type of incoming documents.

#### Component versioning

Versioning is supported for ER components. The following workflow is provided to manage changes in ER components:

1. The version that is originally created is marked as a **Draft** version. This version can be edited and is available for test runs.
2. The **Draft** version can be converted to a **Completed** version. This version can be used in local reporting processes.
3. The **Completed** version can be converted to a **Shared** version. This version is published on LCS and can be used in global reporting processes.
4. The **Shared** version can be converted to a **Discontinued** version. This version can then be deleted.

Versions that have either **Completed** or **Shared** status are available for other data interchange. The following actions can be performed on a component that has these statuses:

- The component can be serialized in XML format and exported as a file in XML format.
- The component can be reserialized from an XML file and imported into the application as a new version of an ER component.

#### Component date effectivity

ER component versions are date-effective. You can set the **Effective from** date for an ER component to specify the date that the component becomes effective for reporting processes. The application session date is used to define whether a component is valid for execution. If more than one version is valid for a particular date, the latest version is used for reporting processes.

#### Component access

Access to ER format components depends on the setting for the ISO country/region code. When this setting is blank for a selected version of a format configuration, a format component can be accessed from any company at run time. When this setting contains ISO country/region codes, a format component is available only from companies that have a primary address that is defined for one of a format component's ISO country/region codes.

Different versions of a data format component can have different settings for ISO country/region codes.

## Configuration

An ER configuration is the wrapper of a particular ER component. That component can be either a data model component or a format component. A configuration can include different versions of an ER component. Each configuration is marked as owned by a specific configuration provider. The **Draft** version of a component of a configuration can be edited when the owner of the configuration has been selected as an active provider in the ER settings in the application.

Each model configuration contains a data model component. A new format configuration can be derived from a specific data model configuration. In the configuration tree, the format configuration that is created appears as a child of the original data model configuration.

The format configuration that is created contains a format component. The data model component of the original model configuration is automatically inserted into the format component of the child format configuration as a default data source.

An ER configuration is shared for application companies.

## Provider

The ER provider is the party identifier that is used to indicate the author (owner) of each ER configuration. ER lets you manage the list of configuration providers. Format configurations that are released for electronic documents as part of the Finance and Operations solution are marked as owned by the **Microsoft** configuration provider.

To learn how to register a new ER provider, play the task guide, **ER Create a configuration provider and mark it as active** (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

## Repository

An ER repository stores ER configurations. The following types of ER repositories are currently supported:

- LCS shared library
- LCS project
- File system
- RCS
- Operations resources
- Global repository

An **LCS shared library** repository provides access to the list of configurations within the Shared asset library in Lifecycle Services (LCS). This type of ER repository can only be registered for the Microsoft provider. From the LCS Shared asset library you can import the latest versions of ER configurations into the current instance.

An **LCS project** repository provides access to the list of configurations of a specific LCS project (LCS project assets library) that was selected when the repository was registered. ER lets you upload shared configurations from the current instance to a specific **LCS project** repository. You can also import configurations from an **LCS project** repository into the current instance of your Finance and Operations apps.

A **File system** repository provides access to the list of configurations that are located as xml files in the specific folder of the local file system of the machine where the AOS service is hosted. Required folder is selected at the repository registration stage. You can import configurations from a **File system** repository into the current instance.

Note that this repository type is accessible in the following environments:

- Cloud-hosted environments deployed for development purposes (containing test models of enclosed suites)
- Locally deployed environments (on-premises)

For more information, see [Import Electronic reporting \(ER\) configurations](#).

An RCS repository provides access to the list of configurations of a specific instance of [Configuration service \(RCS\)](#) that was selected at the repository registration stage. ER lets you import completed or shared configurations from the selected RCS instance into the current instance so you can use them for electronic reporting.

For more information, see [Import Electronic reporting \(ER\) configurations from RCS](#).

A **Global repository** repository provides access to the list of configurations within the global repository in the [Configuration service](#). This type of ER repository can only be registered for the Microsoft provider. From the global repository, you can import the latest versions of ER configurations into the current instance.

For more information, see [Import Electronic reporting \(ER\) configurations from Global repository of Configuration service](#).

An **Operations resources** repository provides access to the list of configurations that Microsoft, as an ER configuration provider, initially releases as part of the application solution. These configurations can be imported into the current instance and used for electronic reporting or playing sample task guides. They can also be used for additional localizations and customizations. Note that the latest versions provided by Microsoft ER configurations must be imported from the LCS Shared asset library by using corresponding the ER repository.

Required **LCS project**, **File system**, and **Regulatory Configuration Services (RCS)** repositories can be registered individually for each configuration provider of the current instance. Each repository can be dedicated to a specific configuration provider.

## Supported scenarios

### **Building a data model**

ER provides a model designer that you can use to build a data model for a particular business domain. All domain-specific business entities, and the relations between them, can be presented in a data model as a hierarchical structure.

To become familiar with the details of this scenario, play the **ER Design domain specific data model** task guide (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

### **Translating data model content**

Data model content (labels and descriptions) can be translated into other languages that the applications support. You might want to translate data model content for the following reasons:

- At design time, to make the content more intelligible for format designers who speak other languages, and who will use the data model for data mapping of format components.
- At run time, to make the content more user-friendly by presenting prompts and help for run-time parameters, and configured validation messages (errors and warnings), in the language that the currently signed-in user prefers.

### **Configuring data model mappings for outgoing documents**

ER provides a model mapping designer that lets users map data models that they have designed to specific application data sources. Based on the mapping, the data will be imported at run time from selected data sources into the data model. The data model is then used as an abstract data source of ER formats that generate outgoing electronic documents.

To become familiar with the details of this scenario, play the **ER Define model mapping and select data sources** and **ER Map data model to selected data sources** task guides (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

### **Configuring data model mappings for incoming documents**



ER provides a model mapping designer that lets users map data models that they have designed to specific destinations. For example, data models can be mapped to the updatable data components (tables, data entities, and views). Based on the mapping, the data will be updated at run time by using the data from the data model. As abstract storage of the ER format, the data model is filled with data that is imported from an incoming electronic document.

### **Storing a designed model component as a model configuration**

ER can store a designed data model, together with associated data mappings, as a model configuration of the current instance. The following illustration shows an example of this type of data model configuration (the payment model configuration).

To become familiar with the details of this scenario, play the **ER Map data model to selected data sources** task guide (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

### **Building a format that uses a data model as a base**

ER supports a format designer that you can use to build the format of an electronic document for a selected business domain by selecting the model component as a base. The same ER format designer lets you map a format that you create to a selected domain's data model mapping as a data source.

To become familiar with the details of this scenario, play the **ER Design domain specific format** task guide (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

### **Building a configuration to generate electronic documents in OPENXML worksheet format**

The ER format designer can be used to build an electronic document in OPENXML worksheet format.

To become familiar with the details of this scenario, play the **ER Create a configuration for reports in OPENXML format** task guide (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process). As part of the task guide step for importing a template, use the [Template of Payment Report \(SampleVendPaymWsReport.xlsx\)](#) Excel file as a template.

### **Building a configuration to generate electronic documents in a Word document format**

The ER format designer can be used to build an electronic document in a Word document format. The following illustration shows an example of this type of format. Note that this format reuses the existing ER configuration that was originally designed to generate the report output in OPENXML format.

To become familiar with the details of this scenario, play the **ER Design a configuration for generating reports in Microsoft WORD format** task guide (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process). As part of the task guide step for importing a template, use the following Word files as templates for the ER format:

- [Template of Payment Report \(SampleVendPaymDocReport.docx\)](#)
- [Bounded template of Payment Report \(SampleVendPaymDocReportBounded.docx\)](#)

### **Building a configuration to import data from incoming electronic documents**

The ER format designer can be used to describe an electronic document that is planned for data import in either XML or text format. The designed format is used to parse an incoming document. The ER format mapping designer can be used to define the binding of the elements of the designed format to the data model.

To become familiar with the details of this scenario, play the **Create required ER configurations to import data from an external file** task guide (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process). Use the following files to play this guide:

- [ER data model configuration \(1099model.xml\)](#)
- [ER format configuration \(1099format.xml\)](#)
- [Sample of the incoming document in XML format \(1099entries.xml\)](#)

- [Sample of the workbook to manage data of incoming document \(1099entries.xlsx\)](#)

### **Storing a designed format component in a format configuration**

ER can store a designed format together with the configured data mappings as a format configuration of the current instance. The preceding illustration shows an example of this type of format configuration (**BACS (UK)**, which is a child of the **Payment model** configuration). To become familiar with the details of this scenario, play the **ER Design domain specific format task guide** (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

### **Configuring Finance to start to use a created format internally**

The application can be configured to start to use a created format to generate electronic reports. The reference to the created format configuration should be defined in the settings of a specific domain. For example, to start to use an ER format configuration for electronic vendor payments in BACS format, the format configuration should be referenced in specific methods of payment.

To become familiar with the details of this scenario, play the **ER Use format to generate electronic document for payments task guide** (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

## Handling ER components

### **Publishing an ER component in LCS to offer it externally (localization)**

The owner of a component (model or format) that has been created can use ER to publish the completed version of the component to LCS. A repository of the **LCS project** type for the current ER configuration provider is required. When the status of the completed version of a component is changed from **COMPLETED** to **SHARED**, that version is published in LCS. When a component has been published to LCS, the owner of that component becomes a provider of the service to support the component. For example, if the format component is designed to generate an electronic document that is legally required (for example, in accordance with a localization scenario), it's assumed that the format will be kept compliant with legislative changes, and that the provider will issue new versions of the component whenever new legislative requirements arise. To become familiar with the details of this scenario, play the **ER Upload a configuration into Lifecycle Services task guide** (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

### **Importing an ER component from LCS to use it internally**

ER lets you import ER components from LCS to the current instance. A repository of the **LCS project** type is required. When an ER component has been imported from LCS to the current instance, the owner of the instance becomes a consumer of the service that is provided by the owner (author) of the imported component. For example, if a format component is designed to generate a specific electronic document from the application in a country/region-specific format (localization scenario), it's assumed that the service consumer will be able to obtain any updates that are made to that format, to keep it compliant with legislative requirements. To become familiar with the details of this scenario, play the **ER Import a configuration from Lifecycle Services task guide** (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

### **Building a format selecting another format as a base (customization)**

ER lets you create (derive) a new component from the current version of a component (base) that was imported from LCS. For example, a user wants to derive a new format to implement some special requirements for an electronic document (such as an additional field or an extensive description) to support a customization scenario. To become familiar with the details of this scenario, play the **ER Upgrade format by adoption of new base version of it task guide** (part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process).

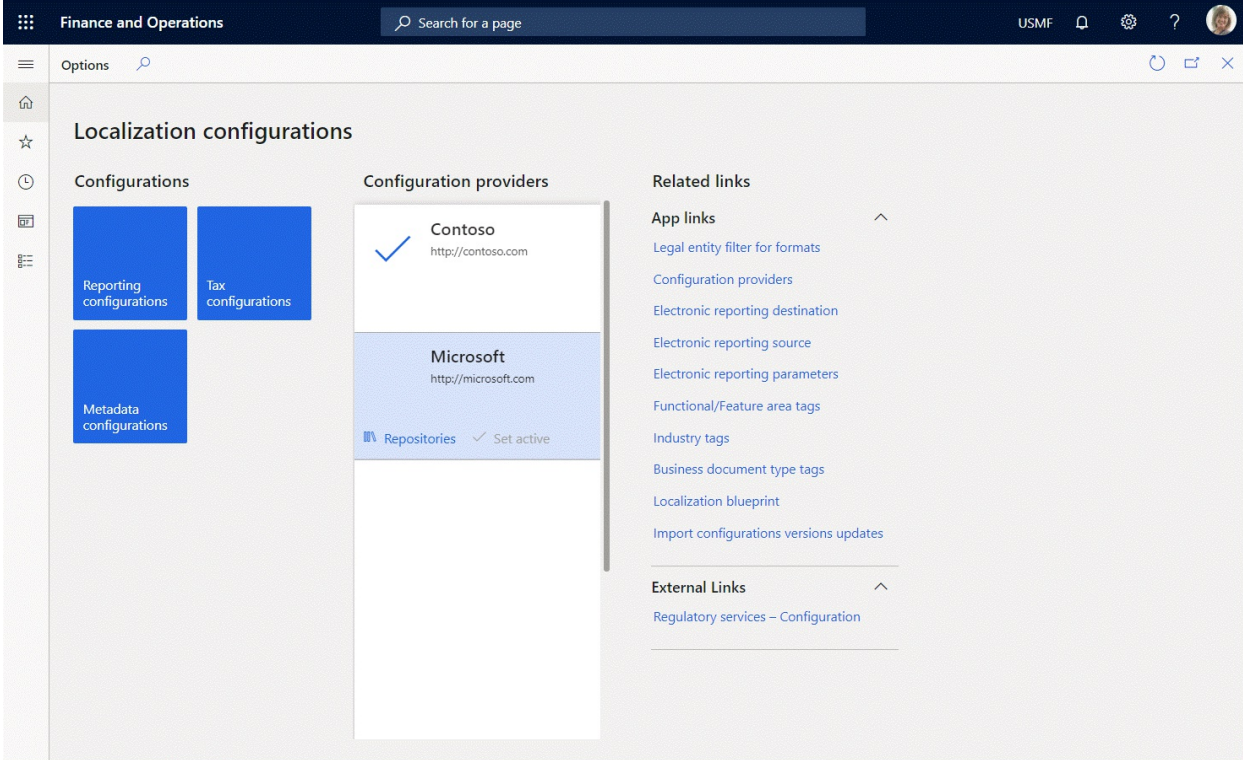
### **Upgrading a format selecting a new version of base format (rebase)**

ER lets you automatically adopt changes of the latest version of the base component in the current draft version

of the derived component. This process is known as *rebasing*. For example, a new regulatory change that has been introduced in the latest version of the format that was imported from LCS can be automatically merged into the customized version of this format of the electronic document. Any changes that can't be merged automatically are considered conflicts. These conflicts are presented for manual resolution in the designer tool for the appropriate component. To become familiar with the details of this scenario, play the **ER Upgrade format by adoption of new base version of that format** task guide (part of the 7.5.5.3 Acquire/Develop changed IT service/solution component (10683) business process).

## List of ER configurations that have been released in Finance

The list of ER configurations for Finance is constantly updated. Open the [Global repository](#) to review the list of ER configurations that are currently supported. On the **Discontinuation details** FastTab, you can review the information about configurations that have been discontinued or that are no longer being maintained.



The screenshot displays the 'Localization configurations' page in the Dynamics 365 Finance and Operations application. The page is divided into three main sections:

- Configurations:** A grid of blue tiles for 'Reporting configurations', 'Tax configurations', and 'Metadata configurations'.
- Configuration providers:** A list of providers. 'Contoso' (http://contoso.com) is at the top with a checkmark. 'Microsoft' (http://microsoft.com) is highlighted in blue and has a 'Repositories' icon and a 'Set active' checkbox.
- Related links:** A list of links under 'App links' and 'External Links'. 'App links' includes: 'Legal entity filter for formats', 'Configuration providers', 'Electronic reporting destination', 'Electronic reporting source', 'Electronic reporting parameters', 'Functional/Feature area tags', 'Industry tags', 'Business document type tags', 'Localization blueprint', and 'Import configurations versions updates'. 'External Links' includes: 'Regulatory services - Configuration'.

## Additional resources

- [Create Electronic reporting \(ER\) configurations](#)
- [Manage the Electronic reporting \(ER\) configuration lifecycle](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create Electronic reporting (ER) configurations

2/18/2021 • 13 minutes to read • [Edit Online](#)

As part of the requirements for LCS solutions for localization and translation, localization ISV solution providers must implement country/region-specific or solution-specific features by using the Electronic reporting tool. This article provides background information that will help you start to use Electronic reporting to create configurations. This article isn't meant to replace any available and upcoming Electronic reporting documentation, but is intended as a supplemental view from the perspective of localization requirements.

## Electronic reporting

General electronic reporting (GER) is a new configurable tool that helps you create and maintain regulatory electronic reporting and payments, based on the following three concepts.

### **Configuration instead of coding**

- Configuration can be done by a business user and doesn't require a developer.
- The data model is defined in business terms.
- Visual editors are used to author all components of the GER configuration.
- A Microsoft Excel-like formula language is used for data transformation.

### **One configuration for multiple Dynamics 365 Finance releases**

- Manage one domain specific data model that is defined in business terms.
- Isolate application release specifics in release-dependent data model mappings.
- Maintain one format configuration for multiple releases of the current version, based on the data model.

### **Easy or automatic upgrade**

- Versioning of GER configurations is supported.
- The Microsoft Dynamics Lifecycle Services (LCS) Assets library can be used as a repository for GER configurations for version exchange.
- Localizations that are based on origin GER configurations can be introduced as child versions.
- A GER configuration tree is provided as a tool that helps control dependencies for versions.
- Only differences in localization (delta configuration) are recorded to enable automatic upgrade to a new version of the origin GER configuration.
- It's easy to manually resolve conflicts that are discovered during automatic upgrade of localization versions.

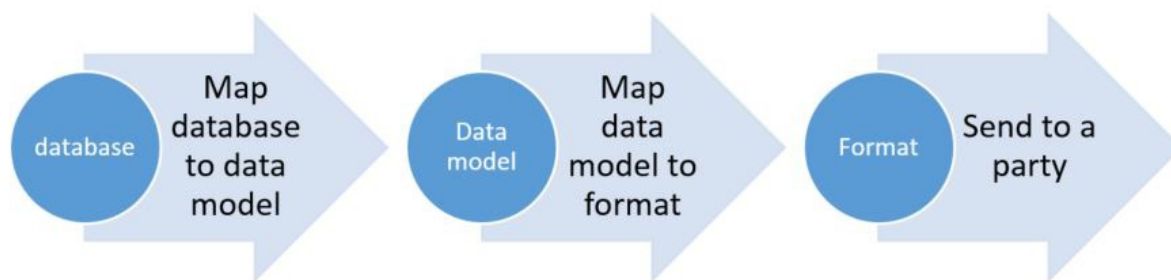
GER lets users define electronic format structures, and then describe how those structures should be filled by using data and algorithms. Users can use a formula language that is very similar to the Excel language for data transformation. To make the database-to-format mapping more manageable, reusable, and independent of format changes, an intermediate data model concept is introduced. This concept enables implementation details to be hidden from the format mapping and also enables a single data model to be reused for multiple format mappings.

## What's new here

WHAT CAN YOU DO?	MICROSOFT DYNAMICS AX 2012	CURRENT VERSION OF FINANCE	WHY IS THIS IMPORTANT?
Configure and generate electronic documents to meet the legal requirements in various countries/regions.	Electronic documents are hard-coded in X++ or as Extensible Stylesheet Language Transformations (XSLTs).	GER is a new tool for configuring and generating electronic documents that target a business user instead of a developer.	GER simplifies the creation, maintenance, and upgrade of electronic document formats to meet legal requirements in various countries/regions.
Configure regulatory updates to formats.	Any format adjustments require development effort.	A business user can configure the formats, based on domain-specific data models (for example, for payments, Intrastat reports, or tax reports).	GER makes the process of creating or changing electronic document formats faster and easier. These changes can be made by business users instead of developers.
Separation of data and formats makes updates easier.	Access to data and access to formatting aren't isolated.	GER lets you set up data models that are domain-specific and independent of the database as data sources for document formats. Formats can be configured based on these domain-specific data models by using simple visual tools that are similar to Excel. Data models and formats support versioning, and formats can be date-effective.	GER makes it faster and easier for partners and customers to upgrade their format and customizations to new versions of formats that are released by Microsoft or other partners.
Distribute data models and formats.	An adjusted format deployment requires a new hotfix package that overrides the existing format.	Each data model or format version is stored in a separate configuration, and is distributed to partners and customers through LCS. By using LCS, partners can share their data model and format configurations with other partners and customers, who can customize and share them further.	GER provides one common way (through LCS) for Microsoft and partners to distribute electronic document configurations to other partners and customers. GER also makes it easier for partners and customers to customize, upgrade, and distribute electronic document formats for their specific business requirements.
Customization and upgrade are easier.	Custom modification of each format must be manually ported to the source code of a new hotfix package.	Partners and customers can customize Microsoft data models and formats, or create their own. GER saves partner and customer configuration changes as deltas to Microsoft configurations to simplify upgrades to new versions of Microsoft configurations. Delta customization and easy upgrade are supported through the whole customization chain.	GER provides one common way (through LCS) for Microsoft and partners to distribute electronic document configurations to other partners and customers. GER also makes it easier for partners and customers to customize, upgrade, and distribute electronic document formats for their specific business requirements.

# Basic concepts

## Main data flow



## Data model configuration creation

It's a good idea to reuse and customize data models that are released by Microsoft whenever you can, or to create a business domain area-specific data model that will introduce the abstract model of required entities and their relations. In this way, you will be aligned with future updates that are released by Microsoft, or can at least reuse your model for the design and maintenance of multiple domain-specific electronic documents that have different formats that are required in different scenarios or countries/regions.

### Design the data model of the created model configuration

A data model is designed to recognize and describe the required business entities and the relations between them in the selected domain. A data model consists of descriptors that express entities by using data containers (records). Properties of entities are expressed by using data items. A record definition is an entity that contains fields (the data items). Each data item has a unique name, label, description, and value. The value of each data item can be designed so that it's recognized as string, integer, real, date, enumerate type, and so on. Additionally, the value can be another record or record list. A single record definition can be selected as a root of the data model. (A root is the starting point of the entire model for data source mapping.) In this case, the model is used as a data source that delivers data according to the single predefined data flow. If no record definition was selected as a root of the data model, the data model contains record definitions that can be assigned as a root at the format mapping stage. The data flow of such a model can be defined as a data source in multiple ways, depending on the nature of the format. For example, a single data model can be designed for the payments domain area. This data model can include data record definitions for the company as a legal entity, for vendors and customers, and also for payments. However, according to the nature of the format, the data must be presented in the following way: payer > payee > payments. Therefore, a single data model can offer data according to the following alternative paths:

- Company > vendor > payment for the Accounts payable domain when the company record definition is selected as a root.
- Customer > company > payment for the Accounts receivable domain when the customer record definition is selected as a root.

## Format configuration creation

You use the data model configuration that is created to hold abstract data for a new electronic format that you want to design. If you intend to consume a data model that was prepared earlier when you create a new format, make sure that you select the **Format based on data model** option for **Create configuration**. After you have a format configuration, you must define a format structure. The structure can be created manually or automatically by importing an example of an XML file or an Excel template. The data model of the parent configuration is automatically offered for format mapping, together with a proper root container. Nevertheless,

a format might require that data be represented in a specific way. Therefore, you can use formula designer to define expressions as virtual data items (calculated fields) for our data containers.

#### NOTE

Although GER allows for direct mapping of format components to database artifacts (tables or data entities), we don't recommend this approach, because it's likely that multiple formats will be maintained in some business domain areas that use the same data sources. Whenever the structure of such database artifacts is changed, the format mapping to the database artifacts must also be changed, and the cost of these changes will be multiplied by the number of maintained formats. Therefore, we recommend that you work through the data model as the abstract description of the domain-specific data structure, and that you use the direct binding of format elements to database components only for simplification and for coverage for specific customizations (for example, to refer to custom tables when these references are required in a limited number of maintained formats).

## Version control

One of the principles behind the design of Electronic reporting is that it should be easy to distribute a data model and formats together with an enhanced maintenance model for their customizations. All the configurations are versioned, and an existing customization can be "cloned" to derive a new configuration for localization or customization implementation. For example, we represent a company that is named Proseware, Inc. We have subscribed to the service of a company that is named Litware Inc., which provides us with the Intrastat returns configuration and supports all legal requirements in it. We have already received specific configurations, from Litware Inc., together with data model and formats, and have deployed them. Our company is working in a district where, in addition to the federal requirements, we must support the following regional requirements:

- As part of Intrastat transactions details, our XML file must show the statistical procedure code that isn't required anywhere else.
- We must limit the length of the company name that is presented in the Intrastat returns header block to 200 characters.

To support these requirements and comply with local district authorities, we must implement this localization as a localized configuration. However, we must keep the link with the origin configuration, so that we can adopt any future changes that are introduced at the federal level as new versions of the origin configuration. Therefore, we import the Litware Inc. origin configuration from LCS, derive it as a new localized configuration, introduce the required changes, complete this work by introducing a first version of the localized format, and start to use it internally. Whenever Litware Inc. offers us a new version of the origin configuration, we import it from LCS, rebase our localized configuration to this version, adopt changes to support new federal requirements, complete this work by introducing a next version of the localized format, and continue to use it internally.

#### NOTE

The draft version of any configuration must be "completed" before it can become available locally for further action, such as the following:

- Make it available so that it can be referenced as a data source from a new format.
- Enable configuration exchange between companies or instances via configuration import/export, and so on.

## Electronic reporting domain coverage

Several out-of-box configurations can be used to meet electronic reporting requirements for specific countries/regions. The following list show some examples of format configurations that are grouped into business domains. To get a complete, up-to-date list of available and supported configurations, open a

configuration repository setup to show the configurations that are available for import from either resources or an LCS Assets library.

- Audit file
  - FEC
  - GDPdU...
- Payments (ISO20022)
  - SEPA CT
  - SEPA DD
  - JBA
  - BACS...
- Statistical reports
  - EU Intrastat...
- Tax reports
  - CIS
  - BAS
  - ELSTER
  - EU Sales list...
- Customer e-Invoice
  - OIOUBL...

## Your solution uptake

You can choose how to move your electronic reporting functionality into GER. However, you should consider the following high-level steps when you plan that move.

1. Review the electronic reporting functionality that your solution currently provides.
2. Identify domain areas that your solution covers, such as Payments and E-Invoices.
3. Review the configurations that are provided by Microsoft. It's likely that you'll find a configuration that you can use as a base. For example, if your solution customizes the SEPA CT payment format, you should extend the SEPA CT configuration.
4. Create new configurations that are based on either an existing model or format, or a new model or format.
5. Define input parameters that users must select when they run the report, and validations for the content of the report.
6. Define mappings with the model by using arithmetic, string, data, or other available Excel-like functions.
7. Define labels and translation to different languages, where applicable.
8. Define templates that have named ranges, and set links from the configuration for the Excel report, if applicable.

## Terminology

TERM	DEFINITION
------	------------



TERM	DEFINITION
GER	Electronic reporting is an engine that simplifies the creation of electronic reports for information interchange with governments, banks, and other parties. Currently, Electronic reporting supports text, XML, and OpenXML spreadsheet formats, and provides an extension interface to support more formats.
Transformation	If you have a typical action that must be done on the source of data before it is sent as output to a format, you can introduce a transformation and attach it to format components. Transformation is a GER formula that takes one value as a parameter and returns another value. For example, you have many format fields that contain spaces, and the spaces should be replaced by spaces when the fields are exported. In this case, you can create a transformation that takes a string argument and uses the REPLACE function to do the job. You can then create string components and associate them with that transformation.
Data model	A data model provides a structure for data. This structure is used to abstractly describe certain business domain areas at sufficient detail to satisfy the reporting requirements in this domain.
Configuration	A container for either a data model or a format, together with its mappings to data sources, that can be maintained and executed, and that supports versioning. The configuration is the entity that will be imported or exported to organize electronic document format exchange between Finance and Operations instances.
Derive action	An operation that uses a configuration that already exists as a basis to create a new configuration.
Rebase action	An operation that updates a derived configuration with changes that were introduced in a new version of the base configuration. The version number is selected at the rebase initialization stage.
Update conflict	A conflict that is discovered during the rebase action, where the new base version contains adjustments of a format/mapping element (name, property, and so on) that has also been adjusted in the derived version.
Relocation conflict	A conflict that is discovered during the rebase action, where the new base version contains a new position (parent element) of a format element (name, property, and so on) that has also been relocated to a different position in the derived version.
Duplication conflict	A conflict that is discovered during the rebase action, where the new base version introduced a new format element that is the same as an element (in other word, it has the same name and child components) that has also been entered in the derived version.

## Additional resources

## Electronic reporting (ER) overview

### Manage the Electronic reporting (ER) configuration lifecycle

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Design domain specific data model

2/18/2021 • 5 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can create a new Electronic reporting (ER) configuration that contains a data model for electronic payment documents. This data model will later be used as a data source when you create the format of the payment documents.

In this example, you will create a configuration for sample company, Litware, Inc. These steps can be performed in any company as ER configurations are shared among companies. To complete these steps, you must first complete the steps in the "Create a configuration provider and mark it as active" procedure.

1. Go to Organization administration > Workspaces > Electronic reporting.

Select the configuration provider for sample company, 'Litware, Inc.' If you don't see this configuration provider, you must first complete the steps in the "Create a configuration provider and mark it as active" procedure.

2. Click Reporting configurations.

You will create a configuration that contains a data model for electronic payment documents. This data model will be used later as a data source when you create the format for the payment documents.

## Create a new data model configuration

1. Click Create configuration to open the drop dialog.
2. In the Name field, type 'Payments (simplified model)'.
3. In the Description field, type 'Payment model configuration'.

The active configuration provider is automatically entered here. This provider will be able to maintain this configuration. Other providers can use this configuration, but will not be able to maintain it.

4. Click 'Create configuration' button to complete the configuration creation task

## Create a data model

You're creating a new data model for the selected configuration. This configuration version will have a status of Draft.

1. Click Designer.

## Define the structure of a party participating in a payment process

1. Click New to open the drop dialog.
2. In the Name field, type 'Party'.
3. Click Add.
4. Click New to open the drop dialog.
5. In the Name field, type 'Name'.
6. In the Item type field, select 'String'.
7. Click Add.
8. In the Find field, type 'Party'.

9. Click Find previous.

## Define the bank structure for this model

1. Click New to open the drop dialog.

2. In the Name field, type 'Agent'.

3. In the Item type field, select 'Record'.

4. Click Add.

5. In the Description field, enter 'Financial institution (for instance, a bank) servicing an account for the party (debtor/creditor).'

Financial institution (for instance, a bank) servicing an account for the party (debtor/creditor).

6. Click New to open the drop dialog.

7. In the Name field, type 'Name'.

8. In the Item type field, select 'String'.

9. Click Add.

10. Click New to open the drop dialog.

11. In the Name field, type 'SWIFT'.

12. Click Add.

13. In the Description field, enter 'Bank identification code'.

14. Click New to open the drop dialog.

15. In the Name field, type 'RoutingNumber'.

16. Click Add.

17. In the Description field, enter 'Routing number'.

18. Click Find previous.

## Define the bank account structure for this model

1. Click New to open the drop dialog.

2. In the Name field, type 'Account'.

3. In the Item type field, select 'Record'.

4. Click Add.

5. In the Description field, enter 'Identification of an account of a party in a financial institution (for instance, a bank).'

Identification of an account of a party in a financial institution (for instance, a bank).

6. Click New to open the drop dialog.

7. In the Name field, type 'Currency'.

8. In the Item type field, select 'String'.

9. Click Add.
10. In the Description field, enter 'Currency code'.
11. Click New to open the drop dialog.
12. In the Name field, type 'Number'.
13. Click Add.
14. Click New to open the drop dialog.
15. In the Name field, type 'IBAN'.
16. Click Add.
17. In the Description field, enter 'International bank account number'.

## Define the payment message structure for credit transfer payment type

1. Click New to open the drop dialog.
2. In the New node as a field, enter 'Model root'.
3. In the Name field, type 'CustomerCreditTransferInitiation'.
4. Click Add.
5. In the Find field, type 'CustomerCreditTransferInitiation'.
6. Click Find previous.
7. Click New to open the drop dialog.
8. In the Name field, type 'MessageIdentification'.
9. Click Add.
10. In the Description field, enter 'The point-to-point reference assigned by the instructing party (and sent to the next party) to identify a message.'.

The point-to-point reference assigned by the instructing party (and sent to the next party) to identify a message.

11. Click New to open the drop dialog.
12. In the Name field, type 'ProcessingDateTime'.
13. In the Item type field, select 'DateTime'.
14. Click Add.
15. In the Description field, enter 'Date and time at which the payment message was created.'.
16. Click New to open the drop dialog.

Define the payment transaction structure for this model.

17. In the Name field, type 'Payments'.
18. In the Item type field, select 'Record list'.
19. Click Add.

20. In the Description field, enter 'Payment lines of the current message'.
21. Click New to open the drop dialog.
22. In the Name field, type 'Creditor'.
23. In the Item type field, select 'Record'.
24. Click Add.
25. In the Description field, enter 'Party to which an amount of money is due.'.
26. Click Switch item reference.
27. In the Find field, type 'Party'.
28. Click Find next.
29. Click OK.
30. In the Find field, type 'Payments'.
31. Click Find next.
32. Click New to open the drop dialog.
33. In the Name field, type 'Debtor'.
34. Click Add.
35. In the Description field, enter 'Party that owes an amount of money to the (ultimate) creditor.'.
36. Click Switch item reference.
37. In the Find field, type 'Party'.
38. Click Find next.
39. Click OK.
40. Click Find next.
41. Click New to open the drop dialog.
42. In the Name field, type 'Description'.
43. In the Item type field, select 'String'.
44. Click Add.
45. Click New to open the drop dialog.
46. In the Name field, type 'Currency'.
47. Click Add.
48. In the Description field, enter 'Currency code'.
49. Click New to open the drop dialog.
50. In the Name field, type 'TransactionDate'.
51. In the Item type field, select 'Date'.
52. Click Add.

53. In the Description field, enter 'Transaction date'.
54. Click New to open the drop dialog.
55. In the Name field, type 'InstructedAmount'.
56. In the Item type field, select 'Real'.
57. Click Add.
58. In the Description field, enter 'The amount of money to be moved between the debtor and creditor, before deduction of charges. The amount should be expressed in the currency as ordered by the initiating party.'  
  
The amount of money to be moved between the debtor and creditor, before deduction of charges. The amount should be expressed in the currency as ordered by the initiating party.
59. Click New to open the drop dialog.
60. In the Name field, type 'End2EndID'.
61. In the Item type field, select 'String'.
62. Click Add.
63. In the Description field, enter 'The unique identification assigned by the initiating party. This identification is passed on, unchanged, throughout the entire end-to-end chain.'
64. In the Name field, type 'PaymentModel'.  
  
The PaymentModel name aligns with predefined interfaces of payment forms.
65. Click Save.
66. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Prepare application metadata to be used in RCS

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can create a new Electronic reporting (ER) configuration that contains application metadata for designing ER model mapping configurations in Regulatory configuration service (RCS). This configuration will be used for designing a sample ER model mapping configuration to access foreign trade transactions. In this example, you will create a configuration for sample company, Litware, Inc. These steps can be performed in any company. To complete these steps, you must first complete the steps in the topic, [Create configuration providers and mark them as active](#).

## Prerequisites

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as **Active**. If you don't see this configuration provider, complete the steps in the procedure [Create configuration providers and mark them as active](#).
3. Click **Metadata configurations**.
4. Assume that RCS will be used to design an ER solution for a Finance and Operation application that will generate electronic documents that contain information from foreign trade business domain. To specify the mapping between ER data model and sources of required data, in RCS we need to have access to metadata of the Finance and Operation application. Therefore, as part of designing ER solution we configure a new ER metadata configuration containing all metadata that is currently required for generation ER reports for selected business domain.

## Add metadata configuration

1. Click **Create configuration** to open the drop dialog.
2. In the **Name** field, type 'Foreign trade metadata'.
3. Click **Create configuration**.
4. Click **Designer**.
5. Click **Add**.

### NOTE

You can select all metadata for the entire application or selected models or selected modules. Be aware that in this case the following metadata will be automatically added: tables of records, enumerations, and extended data types. When additional types of metadata are needed, they must be added manually.

We have some foreign trade transactions related metadata by selecting metadata items manually.

6. Click **Add data source**.
7. Click **Table records**.
8. Use the Quick Filter to filter on the **Name** field with a value of 'Intrastat'.
9. Select the **Intrastat** table record.
10. Click **OK**.

We added metadata information about the Intrastat table of records.



11. In the tree, expand **Table records Intrastat>Relations**.
12. In the tree, select **Table records Intrastat>Relations\IntrastatCommodity (Table records EcoResCategory)**.
13. Click **Add metadata**.

**NOTE**

Metadata about required relations for selected table of records must be added manually.

16. Click **Add data source**.
17. Click **Enumeration**.
18. Use the Quick Filter to filter on the **Name** field with a value of 'IntrastatDirection'.
19. Select the **IntrastatDirection enumeration** record.
20. Click **OK**.
21. Click **Save**.
22. Close the page.

## Complete the draft version of metadata configuration

1. Click **Change status**.
2. Click **Complete**.
3. Click **OK**.
4. Select the completed version 1.

## Export the completed version of metadata configuration from application as XML file

1. Click **Exchange**.
2. Click **Export as XML file**.
3. Click **OK**.

The created ER metadata configuration has been saved as XML file that can be imported to RCS and used as the source of information about metadata for the foreign trade business domain. Based on this information, we can specify the mapping between application metadata and ER data model.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Access application metadata by using ER configuration

2/18/2021 • 3 minutes to read • [Edit Online](#)

The following steps explain how a Regulatory configuration service (RCS) user in the System Administrator or Electronic Reporting Developer role can design a new Electronic reporting (ER) model mapping by using the application metadata. Application metadata will be accessed by using an ER metadata configuration that contains a sample set of metadata to access foreign trade transactions. To complete these steps, in RCS you must first complete the steps in the topic, [Create configuration providers and mark them as active](#) procedure. Then complete the steps in the topic, [Prepare application metadata to be used in RCS](#).

## Prerequisites

1. Go to **All workspaces > Electronic reporting**.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as **Active**. If you don't see this configuration provider, complete the steps in the procedure [Create configuration providers and mark them as active](#).

## Import metadata configuration

1. Click **Metadata configurations**.
2. Import the ER metadata configuration that contains metadata that has been configured to generate electronic documents for foreign trade business. This ER metadata configuration has been exported as XML file while the steps in the [Prepare application metadata to be used in RCS](#) procedure have been completed.
3. Click **Exchange**.
4. Click **Load from XML file**.
5. Click **Browse** and select the 'Foreign trade metadata.xml' file.
6. Click **OK**.
7. Close the page.

## Create data model configuration

1. Click **Reporting configurations**.
2. Click **Create configuration** to open the drop dialog.
3. In the **Name** field, type 'Foreign trade model'.
4. Click **Create configuration**.
5. Click **Designer**.
6. Click **New** to open the drop dialog.
7. In the **Name** field, type 'Root'.
8. Click **Add**.
9. Click **New** to open the drop dialog.
10. In the **Name** field, type 'Transaction'.
11. In the **Item type** field, select **Record list**.
12. Click **Add**.
13. Click **New** to open the drop dialog.
14. In the **Name** field, type 'Commodity code'.

15. In the **Item type** field, select **String**.
16. Click **Add**.
17. Click **New** to open the drop dialog.
18. In the **Name** field, type 'Invoiced amount'.
19. In the **Item type** field, select **Real**.
20. Click **Add**.
21. Click **New** to open the drop dialog.
22. In the **Name** field, type 'Date'.
23. In the **Item type** field, select **Date**.
24. Click **Add**.
25. Click **Root reference**.
26. Click **OK**.
27. Click **Save**.
28. Close the page.
29. Click **Change status**.
30. Click **Complete**.
31. Click **OK**.

## Create model mapping configuration

1. Click **Create configuration** to open the drop dialog.
2. In the **New** field, enter 'Model Mapping based on data model Foreign trade model'.
3. In the **Name** field, type 'Foreign trade mapping'.
4. Click **Create configuration**.
5. Expand the **Prerequisites** section.
6. Click **Edit**.
7. Click **New**.
8. In the list, mark the selected row.
9. In the **Prerequisite component type** field, select **Configuration**.
10. Select **Foreign trade metadata** configuration.
11. Click **Save**.
12. We added the reference to the version 1 of the 'Foreign trade metadata' configuration. Application metadata from this configuration will be offered while this model mapping will be designed.
13. Close the page.
14. Click **Designer**.
15. Click **Designer**.
16. In the tree, select **Dynamics 365 for Operations\Table records**.
17. Click **Add root**.
18. In the **Name** field, type 'Intrastat'.
19. Select **Intrastat** table records.
20. Click **OK**.

### NOTE

The only 2 tables were offered as the only 2 tables were added into the set of metadata which is currently in use.

21. Click **Bind**.
22. In the tree, expand **Intrastat**.

23. In the tree, select **Intrastat\AmountMST**.
24. In the tree, expand **Transaction = Intrastat**.
25. In the tree, select **Transaction = Intrastat\Invoiced amount**.
26. Click **Bind**.
27. In the tree, select **Intrastat\TransDate**.
28. In the tree, select **Transaction = Intrastat\Date**.
29. Click **Bind**.
30. In the tree, expand **Intrastat> Relations**.
31. In the tree, expand **Intrastat> Relations\IntrastatCommodity**.
32. In the tree, select **Intrastat> Relations\IntrastatCommodity\Code**.
33. In the tree, select **Transaction = Intrastat\Commodity code**.
34. Click **Bind**.
35. Click **Validate**.

#### **NOTE**

We have successfully bound elements of data model with items of data sources that are described by using details of application metadata from the referred ER metadata configuration. 36. Click **Save**. 37. Close the page. 38. Close the page. 39. When needed, you can extend the existing set of metadata and then export the new completed version of ER metadata configuration. You can then import it to RCS, and update the prerequisites of the configured model mapping configuration referring to a new version of imported metadata configuration.

#### **NOTE**

This way of getting information about application metadata is the only one available for locally deployed applications (when local business data (LBD), or on-premises, deployment model is used).

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Access application metadata by using connected applications

2/18/2021 • 3 minutes to read • [Edit Online](#)

The following steps explain how a Regulatory configuration service (RCS) user in the System Administrator or Electronic Reporting Developer role can design a new Electronic reporting (ER) model mapping by using metadata in Finance and Operations. Application metadata will be accessed online by using the RCS connected application. Sample ER model mapping will be configured to access foreign trade transactions. To complete these steps, in RCS you must first complete the steps in the topic, [Create configuration providers and mark them as active](#). If you have not completed the steps in the topic, [Access application metadata by using ER configuration](#), go to the [Electronic reporting examples page](#) to download and save the following ER configurations: Foreign trade metadata.xml; Foreign trade model.xml; Foreign trade mapping.xml, and then complete the steps in the procedure.

## Prerequisites

1. Go to **All workspaces > Electronic reporting**.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as **Active**. If you don't see this configuration provider, complete the steps in the procedure [Create configuration providers and mark them as active](#).

## Get required ER configurations

1. Click **Reporting configurations**.
2. If you already completed the steps in the [Access application metadata by using ER configuration](#) procedure, you already have all necessary ER configurations (foreign trade metadata, model and mapping configurations) in the current RCS instance. You can skip all the remaining steps of this sub-task.
3. Click **Exchange**.
4. Click **Load from XML file**.
5. Click **Browse** and select the **Foreign trade metadata.xml** file.
6. Click **OK**.
7. Click **Exchange**.
8. Click **Load from XML file**.
9. Click **Browse** and select the **Foreign trade model.xml** file.
10. Click **OK**.
11. Click **Exchange**.
12. Click **Load from XML file**.
13. Click **Browse** and select the **Foreign trade mapping.xml** file.
14. Click **OK**.

## Register a connected application

1. Close the page.
2. Close the page.
3. Go to **All workspaces > Electronic reporting**.
4. Click **Connected applications**.

5. Make sure that the configured application is Azure based and accessible for the current RCS user. It is also required that the current RCS user has access to the selected application and has been registered as a user of this application playing a role giving them privileges to access application's metadata.
6. Click **New**.
7. In the **Name** field, type 'MyConnectedApp'.
8. In the **Application** field, type 'https:// mycompany.operations.dynamics.com'.
9. In the **Tenant** field, type 'mycompany.onmicrosoft.com'.
10. Click **Save**.
11. When you check connection to configured application, on the **Connect to remote application** page click **Click here to connect to selected remote application** link.
12. Click **Check connection**.
13. Click **Close**.
14. When the connection validation succeeded, version and tenant details will be updated for the configured application in the current grid.

## Review existing model mapping configuration

1. Close the page.
2. Click **Reporting configurations**.
3. In the tree, expand **Foreign trade model**.
4. In the tree, select **Foreign trade model\Foreign trade mapping**.
5. Expand the **Prerequisites** section.

### NOTE

Currently, this mapping refers to the metadata configuration. Application metadata from this configuration will be offered while this model mapping will be designed.

6. Click **Designer**.
7. Click **Designer**.
8. In the tree, select **Dynamics 365 for Operations\Table records**.
9. Click **Add root**.
10. In the **Table** field, enter or select a value.

### NOTE

Currently, this mapping refers to the metadata configuration. Application metadata from this configuration will be offered while this model mapping will be designed.

11. Click **Cancel**.
12. Close the page.
13. Close the page.

## Assign connected application to model mapping

1. Click **Edit**.
2. Select **MyConnectedApp** application.

**NOTE**

Currently, this mapping refers to the metadata of the selected connected application. When the same mapping refers to metadata configuration and connected application at the same time, metadata of the connected application will be used.

3. Click **Designer**.
4. Click **Designer**.
5. In the tree, select **Dynamics 365 for Operations\Table records**.
6. Click **Add root**.
7. In the **Table** field, enter or select a value.

**NOTE**

More than two application tables were offered now as this mapping uses all the metadata of the connected application that has been assigned for it.

8. Click **Cancel**.
9. Click **Validate**.

**NOTE**

We successfully bound elements of data model with items of data sources that are described by using details of metadata of the connected application that has been assigned for this mapping.

10. Close the page.
11. Close the page.

When you need to evaluate this model mapping by using metadata of a different version application, register another connected application, assign it to this model mapping and validate it against new metadata.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Define ER model mappings and select data sources for them

2/18/2021 • 4 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can select data sources for an Electronic reporting (ER) data model. The data sources will be bound to individual components of the selected data model at design time and populate business data to that data model at run-time. In this example, you will select data sources for an existing data model that has been created for sample company, Litware, Inc. To complete these steps, you must first complete the steps in the "Create a new data model" procedure.

## Open the Electronic Reporting configurations tree

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Reporting configurations.

## Insert a new model mapping

1. In the tree, select 'Payments (simplified model)'.
2. Click Designer.
3. Click Map model to datasource.
4. Click New.
  - This will create a new record that will map the data model to data sources. In this example, you will map the data model to data sources for the desired payment type: credit transfer. It is possible to design more than one mapping for a particular data model. For example, you could create a mapping for the different types of payments, such as for direct debit or for credit transfers. In this example, you will create a mapping for credit transfers.
5. In the Name field, type 'CT mapping'.
  - CT mapping
6. In the Description field, type 'Payment model mapping CT'.
  - Payment model mapping CT
7. In the Definition field, type 'CustomerCreditTransferInitiation'.
  - CustomerCreditTransferInitiation
8. ResolveChanges the Definition.
9. Click Save.

## Define required data sources for the current model mapping

1. Click Designer.
2. In the tree, select 'Dynamics 365 for Operations\Table records'.
3. Click Add root.
  - Enter this data source to access payment transactions.
4. In the Name field, type 'Transactions'.
  - Transactions
5. In the Label field, enter 'Transactions'.
  - Transactions



6. In the Help field, enter 'Ledger journal lines'.
  - Ledger journal lines
7. Select Yes in the Ask for query field.
  - Select Yes.
8. In the Table field, type 'LedgerJournalTrans'.
  - LedgerJournalTrans
9. Click OK.
  - Select the LedgerJournalTrans table as a data source for the current data model.
10. In the tree, select 'Functions\Calculated field'.
11. Click Add.
  - Click Add to add a new calculated field.
12. In the Name field, type '\$EndToEndID'.
  - \$EndToEndID
13. Click Edit formula.
14. In the tree, select 'String\CONCATENATE'.
15. Click Add function.
16. In the tree, expand 'Transactions'.
17. In the tree, select 'Transactions\Voucher'.
18. Click Add data source.
19. In the Formula field, enter 'CONCATENATE(Transactions.Voucher, "-", ' '.
  - Type [ , "-", ] at the end of the formula.
20. In the tree, select 'String\TEXT'.
21. Click Add function.
22. In the tree, select 'Transactions\Record-ID(ReclId)'.
  - Record-ID(ReclId)
23. Click Add data source.
24. In the Formula field, enter 'CONCATENATE(Transactions.Voucher, "-", TEXT(Transactions.ReclId))'.
  - Type [)])] at the end of the formula.
25. Click Save.
  - Make sure that no errors have been discovered for the created formula. See the ERRORS tab below the formula editor control.
26. Close the page.
27. Click OK.
  - Add the calculated field to this data source.
28. Click Add.
  - Click Add to add a new calculated field.
29. In the Name field, type '\$Amount'.
  - \$Amount
30. Click Edit formula.
31. In the tree, expand 'Transactions'.
32. In the tree, select 'Transactions\Debit(AmountCurDebit)'.
  - Debit(AmountCurDebit)
33. Click Add data source.
34. In the Formula field, enter 'Transactions.AmountCurDebit - '.
  - Type [ - ] at the end of the formula.
35. In the tree, select 'Transactions\Credit(AmountCurCredit)'.
  - Credit(AmountCurCredit)
36. Click Add data source.
37. Click Save.
38. Close the page.

39. Click OK.
  - This will add the \$Amount calculated field to the selected data source for the current data model.
40. In the tree, select 'Transactions\$Amount'.
41. In the tree, expand 'Transactions'.
42. In the tree, expand or collapse 'Transactions\$Amount'.
43. In the tree, expand or collapse 'Transactions'.
44. In the tree, select 'Dynamics 365 for Operations\Table records'.
45. Click Add root.
  - Enter this data source to access the company's bank account details.
46. In the Name field, type 'BankAccount'.
  - BankAccount
47. In the Label field, enter 'Bank Account'.
  - Bank Account
48. In the Help field, enter 'Bank Account'.
  - Bank Account
49. Select Yes in the Ask for query field.
  - Select Yes.
50. In the Table field, type 'BankAccountTable'.
  - BankAccountTable
51. Click OK.
  - Select the BankAccountTable table as a data source for the current data model.
52. Click Add root.
  - Enter this data source to access the company's requisites.
53. In the Name field, type 'Company'.
  - Company
54. In the Label field, type a value.
  - Company information
55. In the Help field, enter 'Company information'.
  - Company information
56. Select Yes in the Ask for query field.
  - Select Yes.
57. In the Table field, type 'CompanyInfo'.
  - CompanyInfo
58. Click OK.
  - Select the CompanyInfo table as a data source for the current data model.
59. In the tree, select 'Functions\Calculated field'.
60. Click Add root.
  - Insert a calculated field as a new data source.
61. In the Name field, type 'ProcessingDateTime'.
  - ProcessingDateTime
62. In the Label field, enter 'Processing date & time'.
  - Processing date & time
63. Click Edit formula.
64. In the tree, select 'Date/time\SESSIONNOW'.
65. Click Add function.
66. Click Save.

67. Close the page.

68. Click OK.

- Add the ProcessingDateTime calculated field as a data source for the current data model.

69. Click Save.

70. Close the page.

71. Close the page.

72. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Map data model to selected data sources

2/18/2021 • 3 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can map an Electronic reporting (ER) data model to selected data sources. This model mapping will later be used as a data source in a format configuration that will be used to manage electronic payment documents. In this example, you map a data model for sample company, Litware, Inc. to data sources. To complete these steps, you must first complete the steps in the "Select data sources for model mapping" procedure.

## Open ER configurations tree

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Configurations.

## Select created model mapping

1. In the tree, select 'Payments (simplified model)'.
  - Make sure that the model configuration "Payments (simplified model)" has been created in advance. Otherwise, stop now and return after completion of the task guide 'Create a new configuration with data model of the selected domain'.
2. Click Model designer.
3. Click Map model to datasource.
4. Select the 'CT mapping' record.
  - CT mapping

## Bind created data sources to data model elements

1. Click Designer.
2. In the tree, select 'Processing date & time(ProcessingDateTime)'.
3. In the tree, select 'Processing date(ProcessingDateTime)'.
4. Click Bind.
5. In the tree, select 'Payment message identification(MessageIdentification)'.
6. In the tree, expand 'Transactions'.
7. In the tree, select 'Transactions\Journal batch number(JournalNum)'.
8. Click Bind.
9. In the tree, select 'Payments'.
10. In the tree, select 'Transactions'.
11. Click Bind.
12. In the tree, expand 'Payments= Transactions'.
13. In the tree, expand 'Payments= Transactions\Creditor'.
14. In the tree, expand 'Payments= Transactions\Creditor\Account'.
15. In the tree, select 'Payments= Transactions\Creditor\Account\Currency code(Currency)'.
16. In the tree, expand 'Transactions\vendBankAccountInTransactionCompany()'.
17. In the tree, select 'Transactions\vendBankAccountInTransactionCompany()\Currency(CurrencyCode)'.
18. Click Bind.
19. In the tree, select 'Payments= Transactions\Creditor\Account\IBAN code(IBAN)'.

20. In the tree, select 'Transactions\vendBankAccountInTransactionCompany()\IBAN(BankIBAN)'.
21. Click Bind.
22. In the tree, select 'Payments= Transactions\Creditor\Account\Number'.
23. In the tree, select 'Transactions\vendBankAccountInTransactionCompany()\Bank account number(AccountNum)'.
24. Click Bind.
25. In the tree, expand 'Payments= Transactions\Creditor\Agent'.
26. In the tree, select 'Payments= Transactions\Creditor\Agent\Name'.
27. In the tree, select 'Transactions\vendBankAccountInTransactionCompany()\Name'.
28. Click Bind.
29. In the tree, select 'Payments= Transactions\Creditor\Agent\Routing number(RoutingNumber)'.
30. In the tree, select 'Transactions\vendBankAccountInTransactionCompany()\Routing number(RegistrationNum)'.
31. Click Bind.
32. In the tree, select 'Payments= Transactions\Creditor\Agent\SWIFT code(SWIFT)'.
33. In the tree, select 'Transactions\vendBankAccountInTransactionCompany()\SWIFT code(SWIFTNo)'.
34. Click Bind.
35. In the tree, select 'Payments= Transactions\Creditor\Name'.
36. In the tree, expand 'Transactions\findVendTable()'.
37. In the tree, select 'Transactions\findVendTable()\name()'.
38. Click Bind.
39. In the tree, select 'Payments= Transactions\Currency code(Currency)'.
40. In the tree, expand 'Transactions>Relations'.
41. In the tree, expand 'Transactions>Relations\Currency table(Currency)'.
42. In the tree, select 'Transactions>Relations\Currency table(Currency)\Currency code(CurrencyCodeISO)'.
43. Click Bind.
44. In the tree, expand 'Payments= Transactions\Debtor'.
45. In the tree, expand 'Payments= Transactions\Debtor\Account'.
46. In the tree, select 'Payments= Transactions\Debtor\Account\Currency code(Currency)'.
47. In the tree, select 'Bank Account(BankAccount)'.
48. In the tree, expand 'Bank Account(BankAccount)'.
49. In the tree, select 'Bank Account(BankAccount)\Currency(CurrencyCode)'.
50. Click Bind.
51. In the tree, select 'Bank Account(BankAccount)\IBAN'.
52. In the tree, select 'Payments= Transactions\Debtor\Account\IBAN code(IBAN)'.
53. Click Bind.
54. In the tree, select 'Payments= Transactions\Debtor\Account\Number'.
55. In the tree, select 'Bank Account(BankAccount)\Bank account number(AccountNum)'.
56. Click Bind.
57. In the tree, expand 'Payments= Transactions\Debtor\Agent'.
58. In the tree, select 'Payments= Transactions\Debtor\Agent\Name'.
59. In the tree, select 'Bank Account(BankAccount)\Name'.
60. Click Bind.
61. In the tree, select 'Payments= Transactions\Debtor\Agent\Routing number(RoutingNumber)'.
62. In the tree, select 'Bank Account(BankAccount)\Routing number(RegistrationNum)'.
63. Click Bind.
64. In the tree, select 'Payments= Transactions\Debtor\Agent\SWIFT code(SWIFT)'.

65. In the tree, select 'Bank Account(BankAccount)\SWIFT code(SWIFTNo)'.
66. Click Bind.
67. In the tree, select 'Payments= Transactions\Debtor\Name'.
68. In the tree, select 'Company information(Company)'.
69. In the tree, expand 'Company information(Company)'.
70. In the tree, select 'Company information(Company)\Name'.
71. Click Bind.
72. In the tree, select 'Payments= Transactions\Description'.
73. In the tree, select 'Transactions\Description(Txt)'.
74. Click Bind.
75. In the tree, select 'Payments= Transactions\End to end identification code(End2EndID)'.
76. In the tree, select 'Transactions\$EndToEndID'.
77. Click Bind.
78. In the tree, select 'Payments= Transactions\Instructed amount(InstructedAmount)'.
79. In the tree, select 'Transactions\$Amount'.
80. Click Bind.
81. In the tree, select 'Payments= Transactions\Transaction date(TransactionDate)'.
82. In the tree, select 'Transactions\Date(TransDate)'.
83. Click Bind.

## Validate created mapping

1. Click Validate.
  - Validate the new mapping to ensure that all bindings are okay.
2. Click the arrow to expand or collapse the Error List section.
3. Click Save.
4. Close the page.
5. Close the page.
6. Close the page.

## Change the status of the current version of model configuration

1. Click Change status.
  - Change the status of designed model configuration – from Draft to Completed to make it available for payment format design.
2. Click Complete.
  - Select Complete.
3. In the Description field, type a value.
  - For example, 'version 1'.
4. Click OK.
5. Select the completed version of the current configuration.
  - Note that the created configuration is saved as completed version 1.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Create a format configuration (November 2016)

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how a user in the System Administrator or Electronic Reporting Developer role can create a format configuration for Electronic reporting (ER). This format configuration will define the format of electronic documents that are used for processing payments. In this example, you will create a format configuration for sample company, Litware, Inc. To complete these steps, you must first complete the steps in the "Map model to selected datasources" procedure.

## Create a new format configuration

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Click **Reporting configurations**.
3. In the tree, select **Payments (simplified model)**.
4. Click **Create configuration** to open the drop dialog.

### NOTE

If you don't see **Create configuration**, you must enable design mode on the **Electronic reporting parameters** page.

5. In the **New** field, enter **Format based on data model PaymentModel**.
6. In the **Name** field, type **BACS (UK fictitious)**.
7. In the **Description** field, type **BACS vendor payment format (UK fictitious)**.
  - The active configuration provider is automatically entered here. This provider will be able to maintain this configuration. Other providers can use this configuration, but will not be able to maintain it.
  - A particular format of electronic document can be defined. Leave this field blank if you want to select a format at run-time.
8. In the **Data model definition** field, enter or select a value.
9. Click **Create configuration**. A new configuration has been created. The draft version can be used to store the design format for managing electronic documents.

## Design the format of an electronic document

1. Click **Designer**.
2. Click **Add root** to open the drop dialog.
3. In the tree, select **Common\File**.
4. In the **Name** field, type **Xml**.
5. In the **Encoding** field, type **UTF-8**.
6. Click **OK**.
7. Click **Add**.
8. In the tree, select **XML\Element**.
9. In the **Name** field, type **Message**.
10. Click **OK**.
11. In the tree, select **Xml\Message**.
12. Click **Add Element**.
13. In the **Name** field, type **ProcessingDate**.

14. Click **OK**.
15. Click **Add Element**.
16. In the **Name** field, type **MessageId**.
17. Click **OK**.
18. Click **Add Element**.
19. In the **Name** field, type **Payments**.
20. Click **OK**.
21. In the tree, select **Xml\Message\Payments**.
22. Click **Add Element**.
23. In the **Name** field, type **Item**.
24. Click **OK**.
25. In the tree, select **Xml\Message\Payments\Item**.
26. Click **Add**.
27. In the tree, select **XML\Attribute**.
28. In the **Name** field, type **Id**.
29. Click **OK**.
30. Click **Add**.
31. In the tree, select **XML\Element**.
32. In the **Name** field, type **Vendor**.
33. Click **OK**.
34. In the tree, select **Xml\Message\Payments\Item\Vendor**.
35. Click **Add Element**.
36. In the **Name** field, type **Name**.
37. Click **OK**.
38. Click **Add Element**.
39. In the **Name** field, type **Bank**.
40. Click **OK**.
41. In the tree, select **Xml\Message\Payments\Item\Vendor\Bank**.
42. Click **Add Element**.
43. In the **Name** field, type **RoutingNumber**.
44. Click **OK**.
45. Click **Add Element**.
46. In the **Name** field, type **AccountNumber**.
47. Click **OK**.
48. In the tree, select **Xml\Message\Payments\Item\Vendor**.
49. Click **Copy**.
50. In the tree, select **Xml\Message\Payments\Item**.
51. Click **Paste**.
52. In the **Name** field, type **Payer**.
53. In the tree, select **Xml\Message\Payments\Item**.
54. Click **Add Element**.
55. In the **Name** field, type **Currency**.
56. Click **OK**.
57. Click **Add Element**.
58. In the **Name** field, type **Description**.
59. Click **OK**.



60. Click **Add Element**.
61. In the Name field, type **TransDate**.
62. Click **OK**.
63. Click **Add Element**.
64. In the Name field, type **Amount**.
65. Click **OK**.

## Prepare format components for mapping to data model elements

1. In the tree, select **Xml\Message\ProcessingDate**.
2. Click **Add** to open the drop dialog.
3. In the tree, select **Text\DateTime**.
4. In the **Format** field, type **yyyy-MM-dd**.
5. Click **OK**.
6. In the tree, select **Xml\Message\Payments\Item\TransDate**.
7. Click **Add DateTime**.
8. In the **Format** field, type **yyyy-MM-dd**.
9. In the **DateTime** type field, select **Date**.
10. Click **OK**.
11. In the tree, select **Xml\Message\Messageld**.
12. Click **Add** to open the drop dialog.
13. In the tree, select **Text\String**.
14. Click **OK**.
15. In the tree, select **Xml\Message\Payments\Item\Vendor\Name**.
16. Click **Add String**.
17. Click **OK**.
18. In the tree, select **Xml\Message\Payments\Item\Vendor\Bank\RoutingNumber**.
19. Click **Add String**.
20. Click **OK**.
21. In the tree, select **Xml\Message\Payments\Item\Vendor\Bank\AccountNumber**.
22. Click **Add String**.
23. Click **OK**.
24. In the tree, select **Xml\Message\Payments\Item\Payer\Name**.
25. Click **Add String**.
26. Click **OK**.
27. In the tree, select **Xml\Message\Payments\Item\Payer\Bank\RoutingNumber**.
28. Click **Add String**.
29. Click **OK**.
30. In the tree, select **Xml\Message\Payments\Item\Payer\Bank\AccountNumber**.
31. Click **Add String**.
32. Click **OK**.
33. In the tree, select **Xml\Message\Payments\Item\Currency**.
34. Click **Add String**.
35. Click **OK**.
36. In the tree, select **Xml\Message\Payments\Item\Description**.
37. Click **Add String**.
38. Click **OK**.

39. In the tree, select **Xml\Message\Payments\Item\Amount**.
40. Click **Add String**.
41. Click **OK**.
42. Click **Save**.
43. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Map components of the created format to data model elements (November 2016)

2/18/2021 • 3 minutes to read • [Edit Online](#)

The following procedure shows how a user in either the System administrator or Electronic reporting developer role can map data model elements to components of the created Electronic reporting (ER) configuration, which defines an electronic document format for the payments business domain. This format will be used later to generate electronic documents for processing payments. In this example, you will create a format configuration for the sample company, 'Litware, Inc.'. These steps can be performed in any company as ER configurations are shared for all companies. To complete these steps, you must first complete the steps in the "Create a format configuration" task guide.

## Select a format configuration

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Reporting configurations.
3. In the tree, expand 'Payments (simplified model)'.
4. In the tree, select 'Payments (simplified model)\BACS (UK fictitious)'.
5. Click Designer.

## Map format components to data model elements

1. Click Expand/collapse.
2. Click the Mapping tab.
3. In the tree, expand 'model'.
4. In the tree, select 'Xml\Message\ProcessingDate\DateTime'.
5. In the tree, select 'model\ProcessingDateTime'.
6. Click Bind.
7. In the tree, select 'Xml\Message\Messageld\String'.
8. In the tree, select 'model\Messageldentification'.
9. Click Bind.
10. In the tree, expand 'model\Payments'.
11. In the tree, select 'Xml\Message\Payments\Item\Amount\String'.
12. In the tree, select 'model\Payments\InstructedAmount'.
13. Click Bind.
14. In the tree, select 'Xml\Message\Payments\Item\TransDate\DateTime'.
15. In the tree, select 'model\Payments\TransactionDate'.
16. Click Bind.
17. In the tree, select 'Xml\Message\Payments\Item\Description\String'.
18. In the tree, select 'model\Payments\Description'.
19. Click Bind.
20. In the tree, select 'Xml\Message\Payments\Item\Currency\String'.
21. In the tree, select 'model\Payments\Currency'.
22. Click Bind.
23. In the tree, select 'Xml\Message\Payments\Item\ld'.

24. In the tree, select 'model\Payments\End2EndID'.
25. Click Bind.
26. In the tree, expand 'model\Payments\Creditor'.
27. In the tree, expand 'model\Payments\Creditor\Account'.
28. In the tree, expand 'model\Payments\Creditor\Agent'.
29. In the tree, select 'Xml\Message\Payments\Item\Vendor\Name\String'.
30. In the tree, select 'model\Payments\Creditor\Name'.
31. Click Bind.
32. In the tree, select 'Xml\Message\Payments\Item\Vendor\Bank\RoutingNumber\String'.
33. In the tree, select 'model\Payments\Creditor\Agent\RoutingNumber'.
34. Click Bind.
35. In the tree, select 'Xml\Message\Payments\Item\Vendor\Bank\AccountNumber\String'.
36. In the tree, select 'model\Payments\Creditor\Account\Number'.
37. Click Bind.
38. In the tree, select 'Xml\Message\Payments\Item\Payer\Name\String'.
39. In the tree, expand 'model\Payments\Debtor'.
40. In the tree, expand 'model\Payments\Debtor\Account'.
41. In the tree, expand 'model\Payments\Debtor\Agent'.
42. In the tree, select 'model\Payments\Debtor\Name'.
43. Click Bind.
44. In the tree, select 'Xml\Message\Payments\Item\Payer\Bank\RoutingNumber\String'.
45. In the tree, select 'model\Payments\Debtor\Agent\RoutingNumber'.
46. Click Bind.
47. In the tree, select 'Xml\Message\Payments\Item\Payer\Bank\AccountNumber\String'.
48. In the tree, select 'model\Payments\Debtor\Account\Number'.
49. Click Bind.
50. In the tree, select 'Xml\Message\Payments\Item'.
51. In the tree, select 'model\Payments'.
52. Click Bind.
53. Click Save.

## Validate format mapping

1. Click Validate.
  - Validate the new mapping to ensure that all bindings are okay.
2. Close the page.

## Change status of the current version of format configuration

In the next steps, you'll change the status of the format configuration from Draft to Completed to make it available for payment document generation.

1. Click Change status.
2. Click Complete.
3. In the Description field, type a value.
  - For example, 'version 1'.
4. Click OK.
5. Select completed version of the current configuration.

- Note that the configuration is saved as completed version 1.1: version 1 of the format based on the version 1 of the data model.

## Define effective date for completed version of format

Each format version can be configured as available for usage starting from a certain date. When more than one format version is active on a certain date, the latest format (based on version number) will be selected for usage. The session date value is used for proper version selection.

## Restrict access to created format from companies

1. Expand the ISO Country/region codes section.

- Each format access can be restricted by identifying particular countries/regions in which a format is applicable. When the list of countries/regions for particular format is empty, this format can be used in any company. When some ISO country/region codes are inserted in the list of countries/regions, the format can only be use in companies if the primary address is in the country/region.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# (ER) Import configurations from RCS

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can import a new version of an Electronic reporting (ER) configuration from Microsoft Regulatory Configuration Services (RCS). In this example, you will select the version of the ER configuration that has been configured in an RCS instance and import it into the current instance for sample company, Litware, Inc. These steps can be performed in any company because ER configurations are shared among companies. To complete these steps, you must first complete the steps in the topic, [Create configuration providers and mark them as active](#). To complete these steps, you must also have access to an RCS instance containing at least one ER configuration in either **Completed** or **Shared** status.

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as **Active**. If you don't see this configuration provider, complete the steps in the topic, [Create configuration providers and mark them as active](#).
3. If you have no RCS environment provisioned to your company, select **Regulatory services – Configuration** external link and follow the instructions to provision an RCS environment.
4. Select **Electronic reporting parameters**.
5. Select the **RCS** tab.
6. If RCS environment has been already provisioned to your company, use presented on the page URLs to access it.
7. Close the page.

## Register a new ER repository.

1. In the list, mark the selected row.
2. Select Litware, Inc. provider.
3. Select Repositories.
4. Select Add to open the drop dialog.
5. In the Configuration repository type field, enter 'RCS'.
6. Select Create repository.
7. In the RCS environment display name field, enter or select a value.
8. Select the desired RCS instance. You can have several of them.
9. Select OK.

## Import ER configurations from RCS-based repository

1. Select **Show filters**.
2. Enter a filter value of "RCS" on the **Name** field using the **begins with** filter operator.
3. When you open the selected repository, on the **Connect to Regulatory Configuration Services** page, select **Select here to connect to Regulatory Configuration Services** link.
4. Select **Open**.
5. Select **Close**.
6. Select the desired version of ER configuration and select **Import** to bring it in the current instance.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Generate electronic documents for payments using a format configuration

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can use a new Electronic reporting (ER) format configuration to generate electronic documents for processing payments. These steps can be performed in the GBSI sample company.

To complete these steps, you must first complete the steps in the "Create a configuration with format of payment document" procedure.

## Change the configuration of the electronic payment method

1. Go to Accounts payable > Payment setup > Methods of payment.
2. Toggle the File format section to expand it, if needed.
3. Use the Quick Filter to find records. For example, filter on the Method of payment field with a value of 'Electronic'.
4. Click Edit.
5. Set the General electronic reporting field to Yes.
  - Select Yes to use the General electronic reporting pattern for payment files generation.
6. In the Name field, click the drop-down button to open the lookup.
7. Select BACS (UK fictitious) format configuration.
8. Click Save.
9. Close the page.

## Test the format of generated payment files

1. Go to Accounts payable > Payments > Payment journal.
2. Click New.
3. In the list, mark the selected row.
4. In the Name field, click the drop-down button to open the lookup.
5. In the list, click the link in the selected row.
  - Select VendPay.
6. Click Save.
7. Click Lines.
8. In the Company field, type 'DEMF'.
  - DEMF
9. In the Account field, specify the values 'DE-01001'.
  - DE-01001
10. In the Description field, type 'Payment'.
  - Payment
11. In the Debit field, enter a number.
  - 1000
12. Click the Payment tab.
13. In the Method of payment field, click the drop-down button to open the lookup.



14. In the list, find and select the desired record.
  - Select the Electronic value.
15. In the list, click the link in the selected row.
16. Click Save.
17. Click Generate payments.
18. In the Method of payment field, click the drop-down button to open the lookup.
19. In the list, find and select the desired record.
  - Select the Electronic value.
20. In the list, click the link in the selected row.
  - Select the Electronic value.
21. In the File name field, type a value.
  - For example, type 'payments'.
22. In the Bank account field, click the drop-down button to open the lookup.
23. In the list, click the link in the selected row.
  - Select the value GBSI OPER.
24. Click OK.
25. Click OK.
  - Analyze the created payment file in XML format. Compare it with the designed document layout and defined payment transaction attributes.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Upgrade your format by adopting a new, base version of that format

2/18/2021 • 9 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can maintain an Electronic reporting (ER) format configuration. This procedure explains how a custom version of a format can be created based on the format received from a configuration provider (CP). It also explains how to adopt a new, base version of that format.

To complete these steps, you must first complete the steps in the "Create a configuration provider and mark it as active" and "Use created format to generate electronic documents for payments" procedures. These steps can be performed in the GBSI company.

## Select format configuration for customization

1. Go to Organization administration > Workspaces > Electronic reporting.

In this example, sample company Litware, Inc. (<https://www.litware.com>) will act as a configuration provider that supports format configurations for electronic payments for a particular country. Sample company Proseware, Inc. (<http://www.proseware.com>) will act as a consumer of the format configuration that Litware, Inc. provided. Proseware, Inc. uses formats in certain regions of that country.

2. Click Reporting configurations.
3. Click Show filters.
4. Apply the following filters: Enter a filter value of "BACS (UK fictitious)" on the "Name" field using the "begins with" filter operator.

The selected format configuration BACS (UK fictitious) is owned by provider Litware, Inc.

5. Click Show filters.
6. In the list, find and select the desired record.

The version of the format with the status of Completed will be used by Proseware, Inc. for customization.

## Create a new configuration for your custom format of electronic document

Proseware, Inc. received version 1.1 of BACS (UK fictitious) configuration that contains the initial format to generate electronic payment documents from Litware, Inc. in accordance to their service subscription. Proseware, Inc. wants to start using this as a standard for their country but some customization is required to support specific regional requirements. Proseware, Inc. also wants to keep the ability to upgrade a custom format as soon as a new version of it (with changes to support new country-specific requirements) comes from Litware, Inc. and they want to perform this upgrade with the lowest cost.

To do this, Proseware, Inc. needs to create a configuration using the Litware, Inc. configuration BACS (UK fictitious) as a base.

1. Close the page.
2. Select Proseware, Inc. to make it an active provider.

3. Click Set active.
4. Click Reporting configurations.
5. In the tree, expand 'Payments (simplified model)'.
6. In the tree, select 'Payments (simplified model)\BACS (UK fictitious)'.

Select the BACS (UK fictitious) configuration from Litware, Inc. Proseware, Inc. will use version 1.1 as a base for the custom version.

7. Click Create configuration to open the drop dialog.

This lets you create a new configuration for a custom payment format.

8. In the New field, enter 'Derive from Name: BACS (UK fictitious), Litware, Inc.'.

Select the Derive option to confirm the usage of BACS (UK fictitious) as the base for creating the custom version.

9. In the Name field, type 'BACS (UK fictitious custom)'.

10. In the Description field, type 'BACS vendor payment (UK fictitious custom)'.

The active configuration provider (Proseware, Inc.) is automatically entered here. This provider will be able to maintain this configuration. Other providers can use this configuration, but will not be able to maintain it.

11. Click Create configuration.

## Customize your format for the electronic document

1. Click Designer.
2. Click Expand/collapse.
3. Click Expand/collapse.
4. In the tree, select 'Xml\Message\Payments\Item\Vendor\Bank'.
5. Click Add to open the drop dialog.
6. In the tree, select 'XML\Element'.
7. In the Name field, type 'IBAN'.
8. Click OK.
9. In the tree, select 'Xml\Message\Payments\Item\Vendor\Bank\IBAN'.
10. Click Add to open the drop dialog.
11. In the tree, select 'Text\String'.
12. Click OK.
13. In the tree, select 'Xml\Message\Payments\Item\Vendor\Name\String'.
14. In the Maximum length field, enter '60'.
15. Click the Mapping tab.
16. In the tree, expand 'model'.
17. In the tree, expand 'model\Payments'.
18. In the tree, expand 'model\Payments\Creditor'.
19. In the tree, expand 'model\Payments\Creditor\Account'.
20. In the tree, select 'model\Payments\Creditor\Account\IBAN'.
21. In the tree, select 'Xml\Message\Payments\Item = model.Payments\Vendor\Bank\IBAN\String'.
22. Click Bind.
23. Click Save.

## Validate the customized format

1. Click Validate.

Validate the customized format layout and data mapping changes to make sure that all bindings are okay.

2. Close the page.

## Change the status of the current version of the custom format configuration

Change the status of the designed format configuration from Draft to Completed to make it available for payment document generation.

1. Click Change status.

Note that the current version of the selected configuration is in Draft status.

2. Click Complete.
3. In the Description field, type a value.
4. Click OK.
5. In the list, find and select the desired record.

Note that the created configuration is saved as completed version 1.1.1. This means it is version 1 of the custom BACS (UK fictitious custom) format, which is based on version 1 of the BACS (UK fictitious) format, which is based on version 1 of the Payments (simplified model) data model.

## Test the customized format to generate payment files

Complete the steps in the "Use created format to generate electronic documents for payments" procedure in a parallel Finance and Operations session. Select the BACS (UK fictitious custom) format in electronic payment method parameters. Make sure that the created payment file contains the recently introduced XML node presenting IBAN code in accordance to regional requirements.

## Update the existing country-specific configuration

Litware, Inc. needs to update the BACS (UK fictitious) configuration and adopt new country requirements for managing the format of the electronic document. Later, this will be enclosed in a new version of this configuration that will be offered for service subscribers, including Proseware, Inc.

In real service provision related processes, each new version of BACS (UK fictitious) can be imported by Proseware, Inc. from Litware, Inc. configurations' LCS repository. In this procedure we will simulate this by updating BACS (UK fictitious) on behalf of a service provider.

1. Close the page.
2. Select Litware, inc. provider.
3. Click Set active.
4. Click Reporting configurations.
5. In the tree, expand 'Payments (simplified model)'.  
6. In the tree, select 'Payments (simplified model)\BACS (UK fictitious)'.

The draft version owned by Litware, Inc. provider BACS (UK fictitious) is selected to bring in changes to

support new country-specific requirements.

## Localize the base format of the electronic document

Assume that there are new country-specific requirements to be supported by Litware, Inc.:

- A value for the creditor's bank SWIFT code in each payment transaction.
- A limit of 100 characters for the length of text for the vendor's name in a generating file.
- New country-specific requirements
- Select the draft version of the desired configuration to introduce required changes.

1. Click Designer.
2. Click Expand/collapse.
3. Click Expand/collapse.
4. In the tree, select 'Xml\Message\Payments\Item\Vendor\Bank'.
5. Click Add to open the drop dialog.
6. In the tree, select 'XML\Element'.
7. In the Name field, type 'SWIFT'.
8. Click OK.
9. In the tree, select 'Xml\Message\Payments\Item\Vendor\Bank\SWIFT'.
10. Click Add to open the drop dialog.
11. In the tree, select 'Text\String'.
12. Click OK.
13. In the tree, select 'Xml\Message\Payments\Item\Vendor\Name\String'.
14. In the Maximum length field, enter '100'.
15. Click the Mapping tab.
16. In the tree, expand 'model'.
17. In the tree, expand 'model\Payments'.
18. In the tree, expand 'model\Payments\Creditor'.
19. In the tree, expand 'model\Payments\Creditor\Agent'.
20. In the tree, select 'model\Payments\Creditor\Agent\SWIFT'.
21. In the tree, select 'Xml\Message\Payments\Item = model.Payments\Vendor\Bank\SWIFT\String'.
22. Click Bind.
23. Click Save.

## Validate the localized format

1. Click Validate.
2. Close the page.

## Change the status of the current version of the base format configuration

Change the status of the updated base format configuration from Draft to Completed to make it available for generation of payment documents and updates of format configurations derived from it.

1. Click Change status.

Note that the current version of the selected configuration is in Draft status.

2. Click Complete.

3. In the Description field, type a value.
4. Click OK.
5. In the list, find and select the desired record.

## Change the base version for the custom format configuration

Proseware, Inc. is informed that a new version 1.2 of BACS (UK fictitious) configuration is available to generate electronic payment documents in accordance to recently announced country-specific requirements. Proseware, Inc. wants to start using it as a standard for the country.

To do this, Proseware, Inc. needs to change the base configuration version for the custom configuration BACS (UK fictitious custom). Instead of version 1.1 of BACS (UK fictitious) use new version 1.2.

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Select the Proseware, Inc. provider to mark it as active.
3. Click Set active.
4. Click Reporting configurations.
5. In the tree, expand 'Payments (simplified model)'.
6. In the tree, expand 'Payments (simplified model)\BACS (UK fictitious)'.
7. In the tree, select 'Payments (simplified model)\BACS (UK fictitious)\BACS (UK fictitious custom)'.

Select the BACS (UK fictitious custom) configuration, which is owned by Proseware, Inc.

Use the draft version of the selected configuration to introduce required changes.

8. Click Rebase.

Select the new version 1.2 of the base configuration to be applied as a new base for updating the configuration.

9. Click OK.

Note that some conflicts have been discovered between merging the custom version and a new base version representing some format changes that can't be merged automatically.

## Resolve rebase conflicts

1. Click Designer.

Note that changes to the vendor's name text length limit couldn't be resolved automatically. Therefore, this is presented in a conflicts list. For each conflict of type Update, the following options are available: - Apply a prior base value (button on top of the grid) to bring in the previous base version value (0 in our case). - Apply a base value (button on top of the grid) to bring in the new base version value (100 in our case). - Keep your own (custom) value (60 in our case). Click Apply base value to apply a country-specific limit of 100 characters for vendor's name text length.

Note that Proseware, Inc. and Litware, Inc. have custom and local versions of this format using IBAN and SWIFT codes with related components that are automatically merged in the managing format.

2. Click Apply base value.

Click Apply base value to apply the country-specific limit of 100 characters for vendor names.

3. Click Save.

Saving the format will remove resolved conflicts from the conflicts list.

4. Close the page.

## Change the status of the new version of the custom format configuration

1. Click Change status.

Change the status of the updated, custom format configuration from Draft to Completed. This will make the format configuration available for generating payment documents. Note that the current version of the selected configuration is in Draft status.

2. Click Complete.

3. In the Description field, type a value.

4. Click OK.

Note that the created configuration is saved as completed version 1.2.2: version 2 of base BACS (UK fictitious custom) format, which is based on version 2 of base BACS (UK fictitious) format, which is based on version 1 of Payments (simplified model) data model.

## Test the customized format for payment files generation

Complete the steps in the "Use created format to generate electronic documents for payments" procedure in parallel Finance and Operations session. Select the created 'BACS (UK fictitious custom)' format in electronic payment method parameters. Make sure that the created payment file contains recently introduced by Proseware, Inc. XML node presenting IBAN account code in accordance to regional requirements. The file also should contain the recently introduced by Litware, Inc. XML node presenting SWIFT bank code in accordance to country requirements.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Formula designer in Electronic reporting (ER)

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains how to use the formula designer in Electronic reporting (ER). When you design a format for a specific electronic document in ER, you can use formulas to transform data so that it meets the requirements for the document's fulfillment and formatting. These formulas resemble formulas in Microsoft Excel. Various types of functions are supported in the formulas: text, date and time, mathematical, logical, information, and data type conversion functions, and also other, business domain-specific functions.

## Formula designer overview

ER supports the formula designer. Therefore, at design time, you can configure expressions that can be used for the following tasks at runtime:

- Transform data that is received from an application database and that should be entered in an ER data model that is designed to be a data source for ER formats. (For example, these transformations might include filtering, grouping, and data type conversion.)
- Format data that must be sent to a generating electronic document in accordance with the layout and conditions of a specific ER format. (For example, the formatting might be done in accordance with the requested language or culture, or the encoding).
- Control the process of creating electronic documents. (For example, the expressions can enable or disable the output of specific elements of the format, depending on processing data. They can also interrupt the document creation process or throw messages to users.)

You can open the **Formula designer** page when you perform any of the following actions:

- Bind data source items to data model components.
- Bind data source items to format components.
- Complete maintenance of calculated fields that are part of data sources.
- Define the visibility conditions for user input parameters.
- Design a format's transformations.
- Define the enabling conditions for the format's components.
- Define the file names for the format's FILE components.
- Define the conditions for process control validations.
- Define the message text for process control validations.

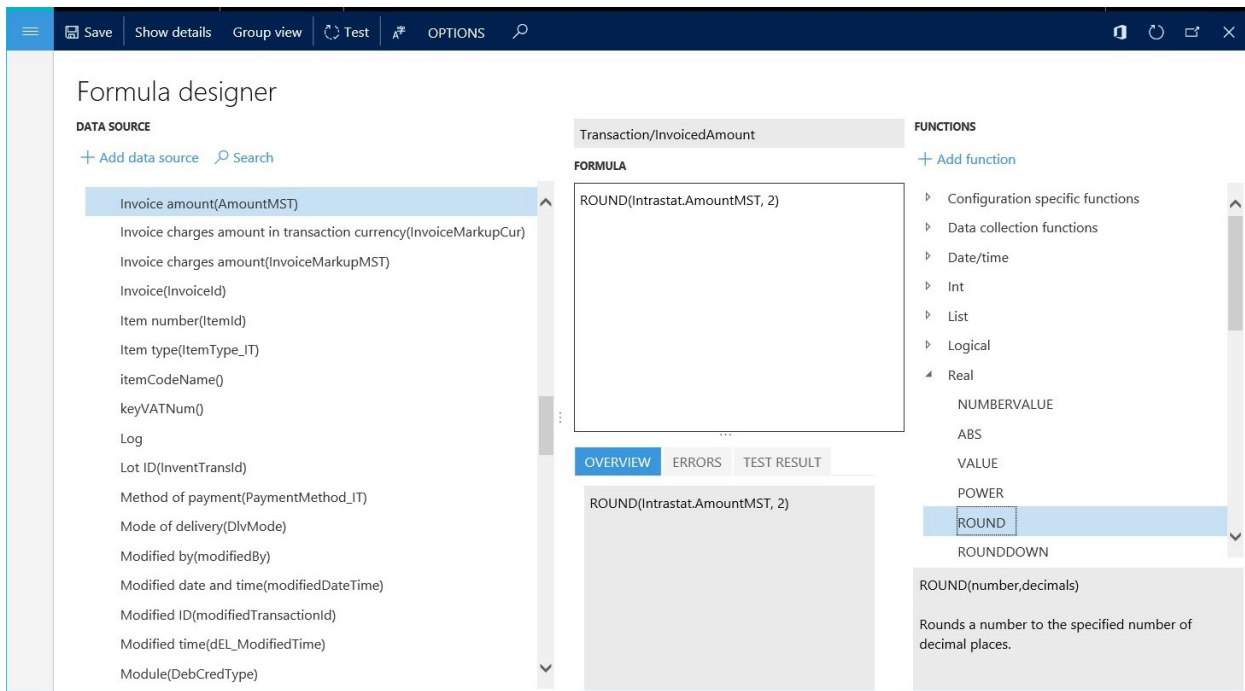
## Data binding

The ER formula designer can be used to define an expression that transforms data that is received from data sources, so that the data can be entered in the data consumer in the following ways at runtime:

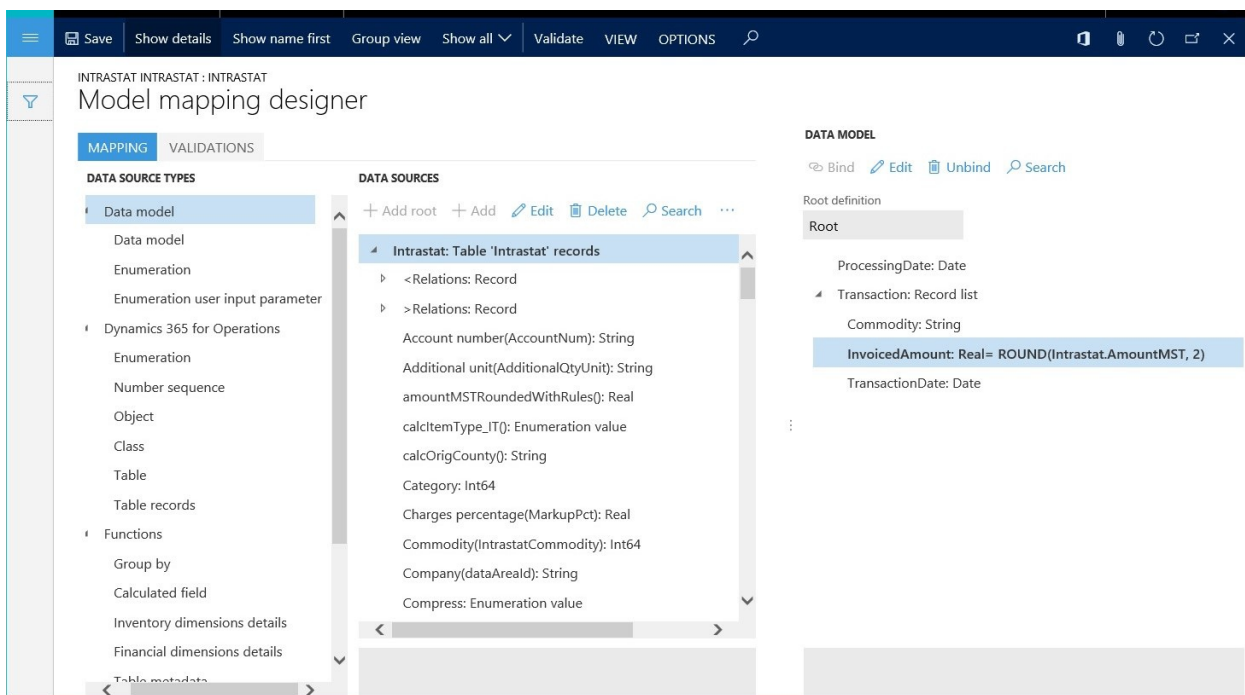
- From application data sources and runtime parameters to an ER data model
- From an ER data model to an ER format
- From application data sources and runtime parameters to an ER format

The following illustration shows the design of an expression of this type. In this example, the expression rounds the value of the **Intrastat.AmountMST** field in the Intrastat table to two decimal places and then returns the rounded value.





The following illustration shows how an expression of this type can be used. In this example, the result of the designed expression is entered in the **Transaction.InvoicedAmount** component of the **Tax reporting model** data model.

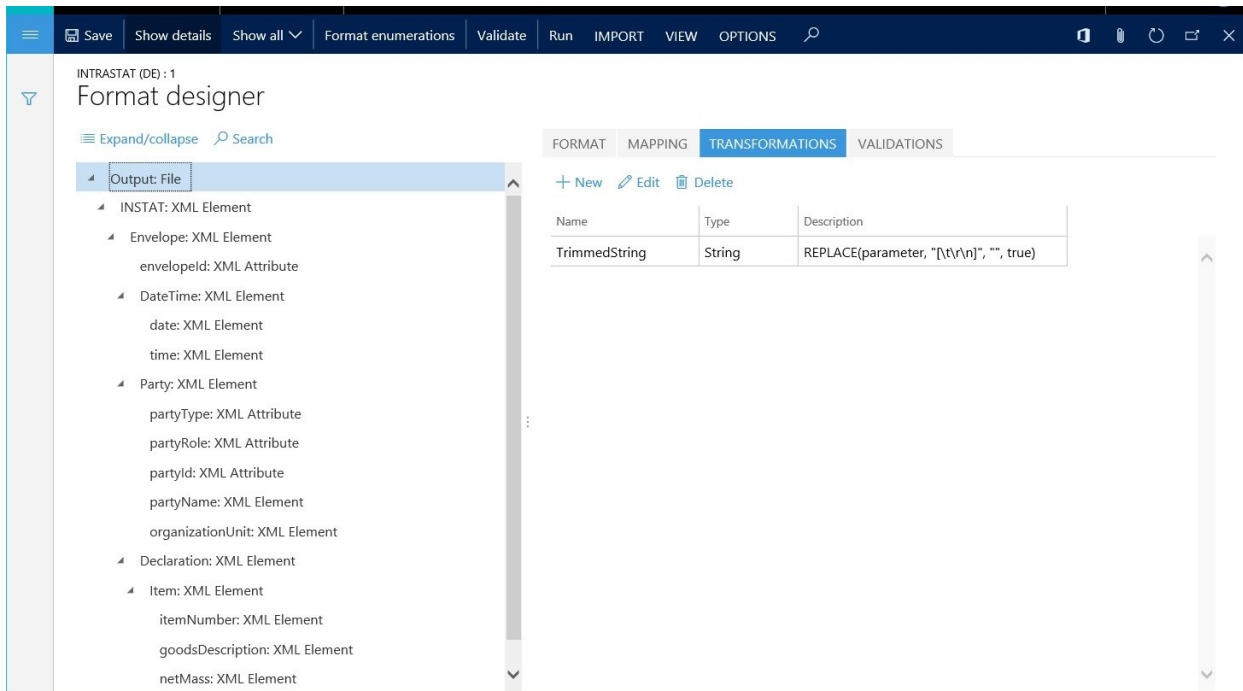


At runtime, the designed formula, `ROUND (Intrastat.AmountMST, 2)`, rounds the value of the **AmountMST** field for each record in the **Intrastat** table to two decimal places. It then enters the rounded value in the **Transaction.InvoicedAmount** component of the **Tax reporting** data model.

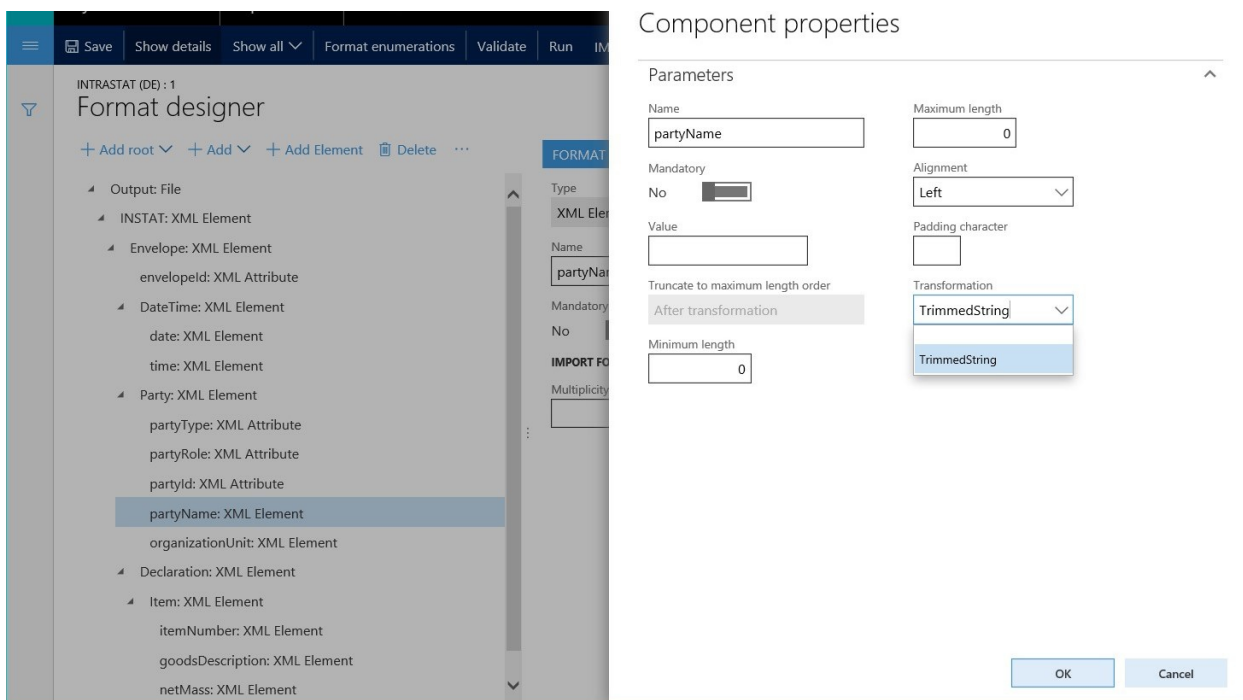
## Data formatting

The ER formula designer can be used to define an expression that formats data that is received from data sources, so that the data can be sent as part of the generating electronic document. You might have formatting that must be applied as a typical rule that should be reused for a format. In this case, you can introduce that formatting one time in the format configuration, as a named transformation that has a formatting expression. This named transformation can then be linked to many format components where the output must be formatted according to the formatting expression that you created.

The following illustration shows the design of a transformation of this type. In this example, the `TrimmedString` transformation truncates incoming data of the `String` data type by removing leading and trailing spaces. It then returns the truncated string value.

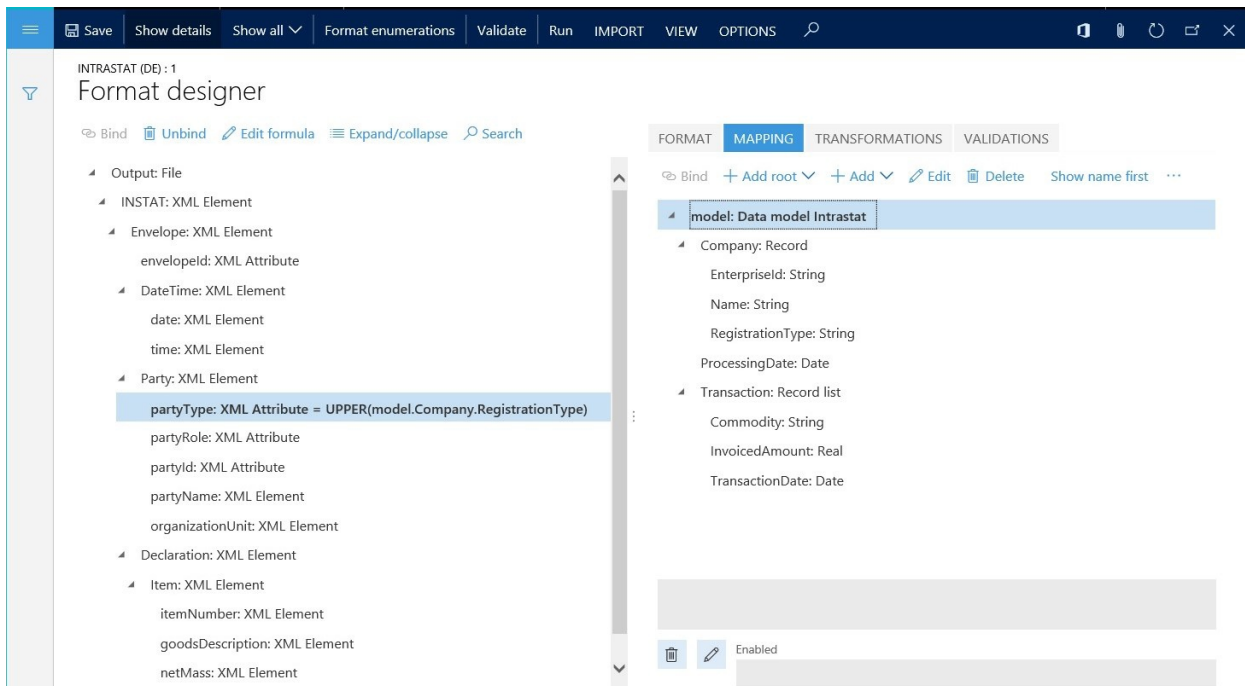


The following illustration shows how a transformation of this type can be used. In this example, several format components send text as output to the generating electronic document at runtime. All these format components refer to the `TrimmedString` transformation by name.



When format components, such as the `partyName` component in the preceding illustration, refer to the `TrimmedString` transformation, the transformation sends text as output to the generating electronic document. This text doesn't include leading and trailing spaces.

If you have formatting that must be applied individually, you can introduce that formatting as an individual expression of a binding of a specific format component. The following illustration shows an expression of this type. In this example, the `partyType` format component is bound to the data source via an expression that converts incoming data from the `Model.Company.RegistrationType` field in the data source to uppercase text. The expression then sends that text as output to the electronic document.



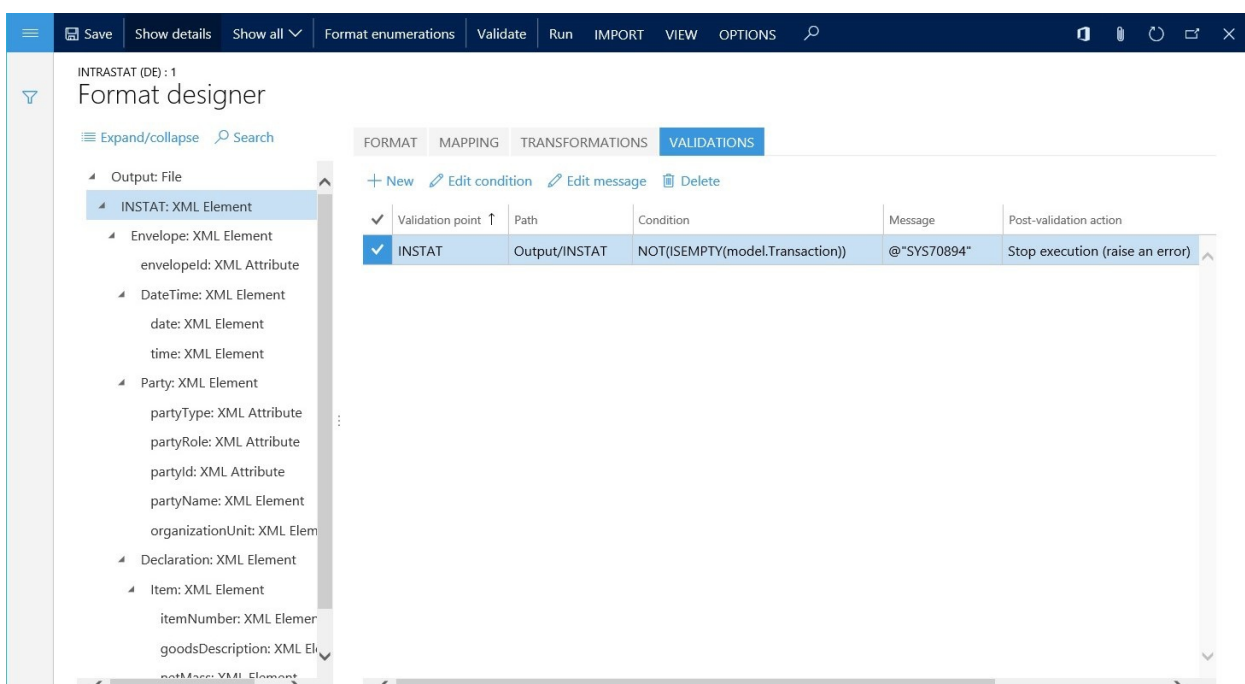
## Process flow control

The ER formula designer can be used to define expressions that control the process flow of generating electronic documents. You can perform the following tasks:

- Define conditions that determine when a document creation process must be stopped.
- Specify expressions that either create messages for the user about stopped processes or throw execution log messages about the continuing process of report generation.
- Specify the file names of generating electronic documents, and control the conditions of their creation.

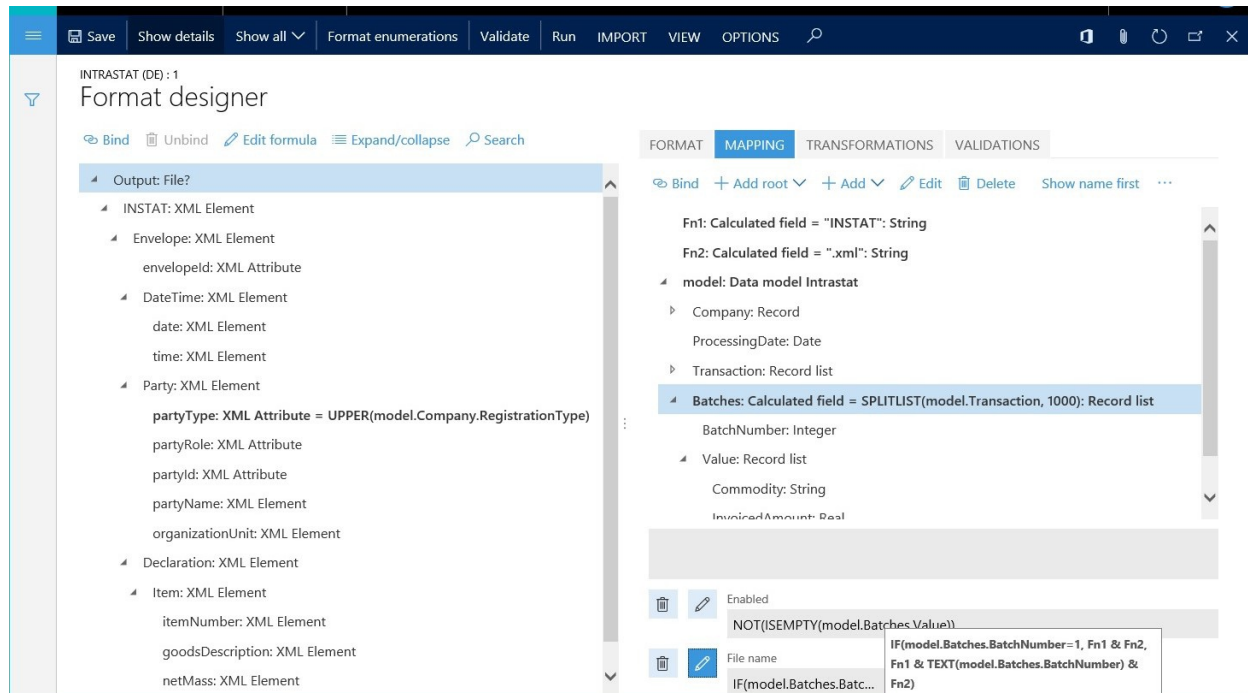
Each rule of the process flow control is designed as an individual validation. The following illustration shows a validation of this type. Here is an explanation of the configuration in this example:

- The validation is evaluated when the **INSTAT** node is created during generation of the XML file.
- If the list of transactions is empty, the validation stops the execution process and returns **FALSE**.
- The validation returns an error message that includes the text of label SYS70894 in the user's preferred language.



The ER formula designer can also be used to generate a file name for a generating electronic document and to control the file creation process. The following illustration shows the design of a process flow control of this type. Here is an explanation of the configuration in this example:

- The list of records from the **model.Intrastat** data source is divided into batches. Each batch contains up to 1,000 records.
- The output creates a zip file that contains one file in XML format for every batch that was created.
- An expression returns a file name for generating electronic documents by concatenating the file name and the file name extension. For the second batch and all subsequent batches, the file name contains the batch ID as a suffix.
- An expression enables (by returning **TRUE**) the file creation process for batches that contain at least one record.



## Document content control

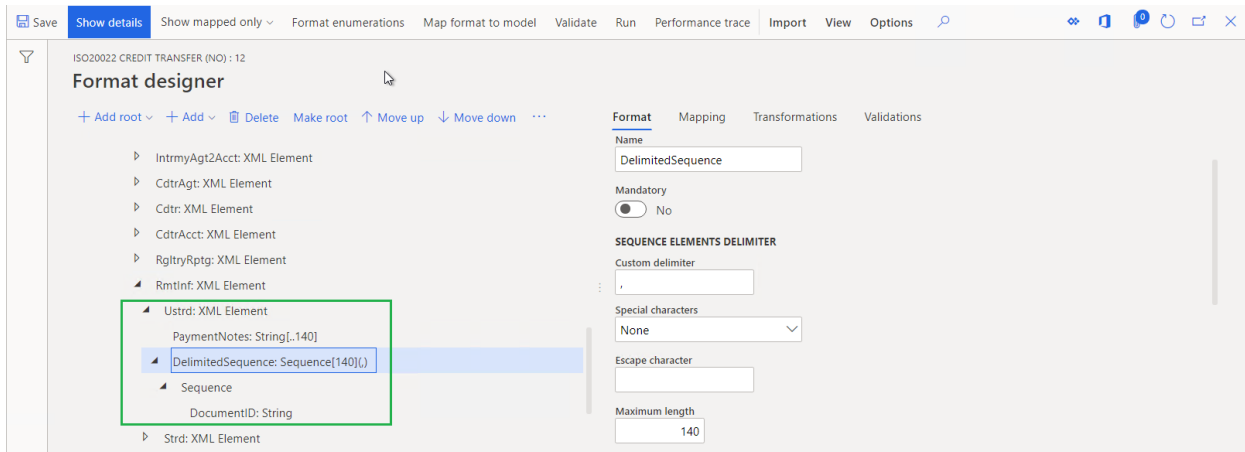
The ER formula designer can be used to configure expressions that control what data will be put into generated electronic documents at runtime. The expressions can enable or disable the output of specific elements of the format, depending on processing data and configured logic. These expressions can be entered for a single format element in the **Enabled** field on the **Mapping** tab of the **Operations designer** page. You can enter the expressions as a logic condition that returns a *Boolean* value:

- If the condition returns **True**, the current format element is run.
- If the condition returns **False**, the current format element is skipped.

The following illustration shows expressions of this type. (Version 11.12.11 of the **ISO20022 Credit transfer (NO)** format configuration that is provided by Microsoft is used as an example.) The **XMLHeader** format component is configured to describe the structure of the credit transfer message according to the ISO 20022 XML message standards. The

**XMLHeader/Document/CstmrCdtTrfInittn/PmtInf/CdtTrfTxInf/RmtInf/Ustrd** format component is configured to add the **Ustrd** XML element to the generated message and to put the remittance information in an unstructured format as text of the following XML elements:

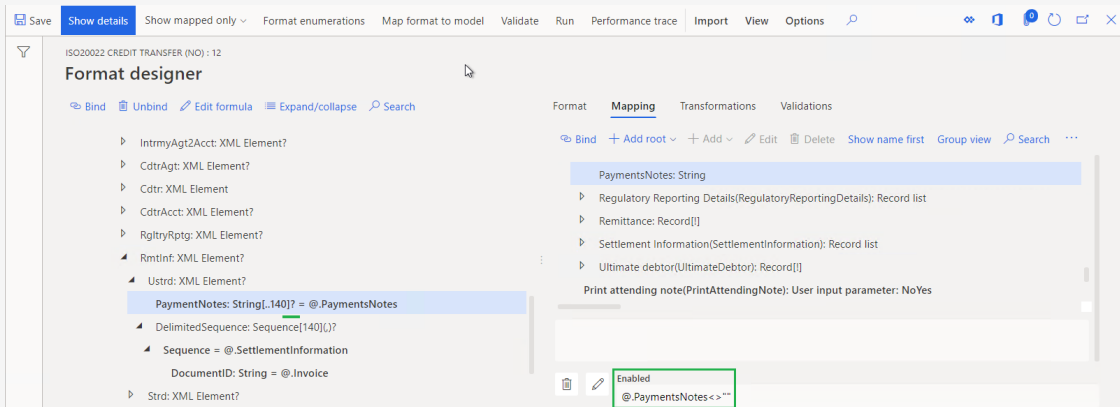
- The **PaymentNotes** component is used to generate the text of payment notes.
- The **DelimitedSequence** component generates comma-separated invoice numbers that are used to settle the current credit transfer.



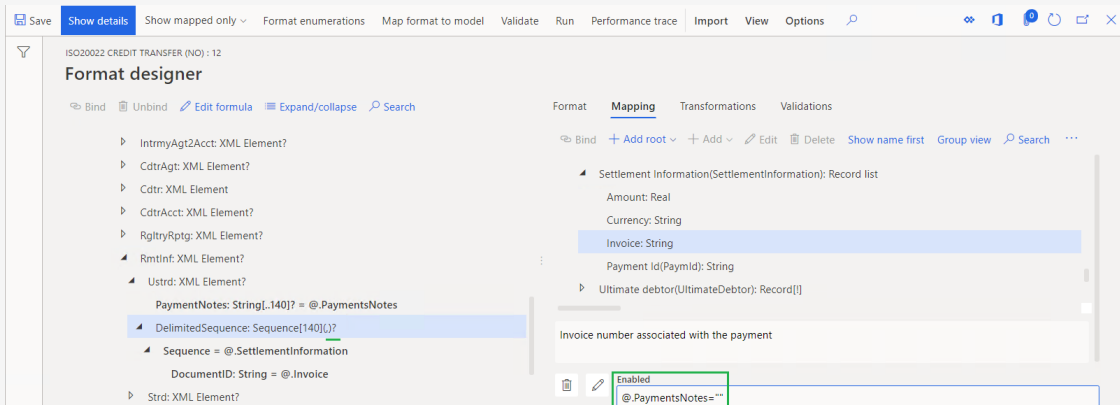
## NOTE

The **PaymentNotes** and **DelimitedSequence** components are labeled by using a question mark. A question mark indicates that the use of a component is conditional. In this case, use of the components is based on the following criteria:

- The `@.PaymentsNotes <> ""` expression that is defined for the **PaymentNotes** component enables (by returning **TRUE**) the **Ustrd** XML element to be filled with the text of payment notes, if that text isn't blank for the current credit transfer.



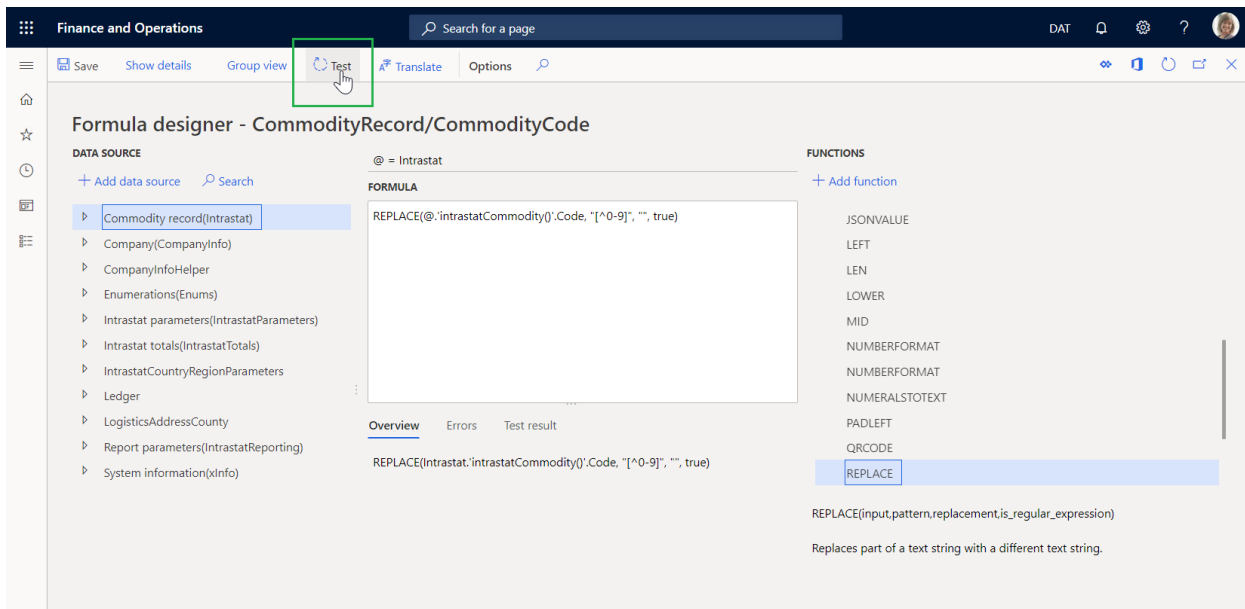
- The `@.PaymentsNotes = ""` expression that is defined for the **DelimitedSequence** component enables (by returning **TRUE**) the **Ustrd** XML element to be filled with a comma-separated list of the invoice numbers that are used to settle the current credit transfer, if the text of payment notes for that credit transfer is blank.



Based on this setup, the message that is generated for each debtor payment, the **Ustrd** XML element, will contain either the text of payment notes or, when that text is blank, a comma-separated list of the invoice numbers that are used to settle the payment.

## Validation of configured formulas

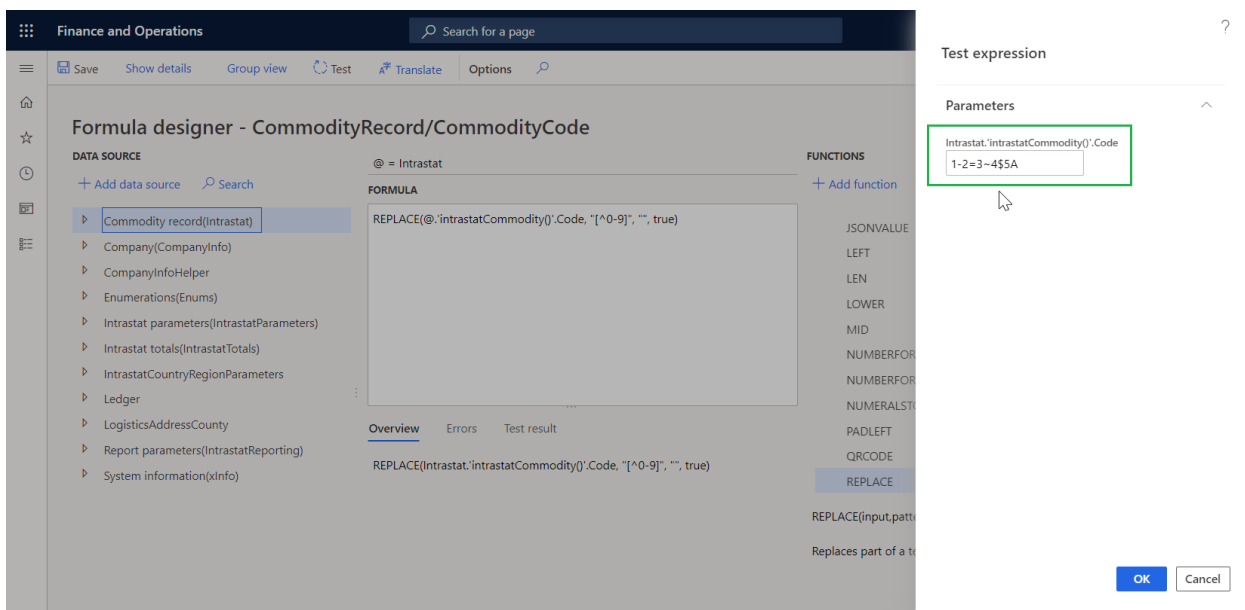
On the **Formula designer** page, select **Test** to validate how the configured formula works.



When the values of formula arguments are required, you can open the **Test expression** dialog box from the **Formula designer** page. In most cases, these arguments must be manually defined, because the configured bindings aren't run at design time. The **Test result** tab on the **Formula designer** page shows the result from execution of the configured formula.

The following example shows how you can test the formula that is configured for the foreign trade domain to make sure that the Intrastat commodity code contains only digits.

When you test this formula, you can use the **Test expression** dialog box to specify the value of the Intrastat commodity code for testing.



After you specify the Intrastat commodity code and select **OK**, the **Test result** tab on the **Formula designer** page shows the result of execution of the configured formula. You can then evaluate whether the result is acceptable. If the result isn't acceptable, you can update the formula and test it again.

Some formulas can't be tested at design time. For example, a formula might return a result of a data type that can't be shown on the Test result tab. In this case, you receive an error message that states that the formula can't be tested.

## Additional resources

- [Electronic Reporting overview](#)
- [Electronic reporting formula language](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Electronic reporting advanced formula editor

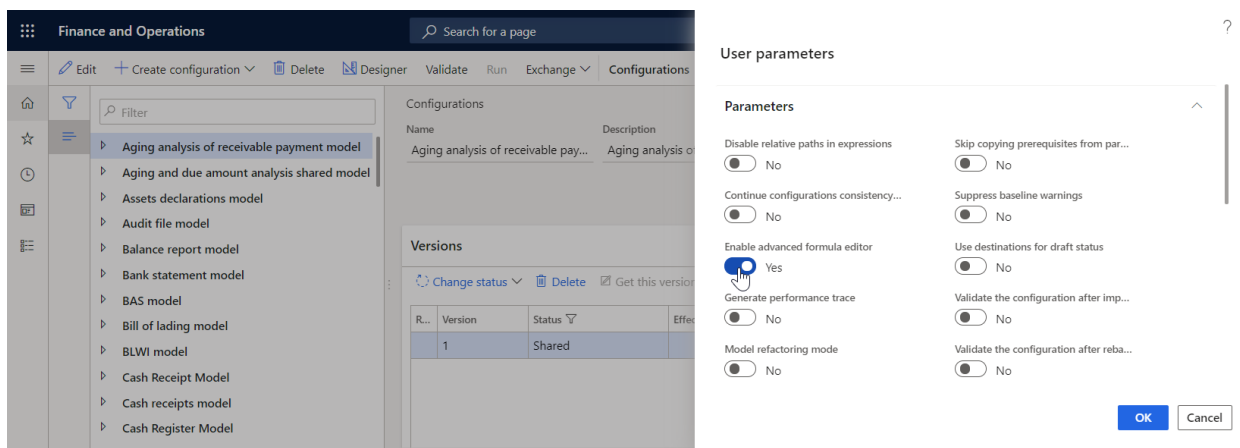
2/18/2021 • 4 minutes to read • [Edit Online](#)

In addition to the [Electronic reporting formula editor](#), you can use the advanced Electronic reporting formula editor to improve the experience of configuring Electronic reporting (ER) expressions. The advanced editor is browser-based and powered by the [Monaco editor](#). The most commonly used advanced editor features are described in this topic:

- [Code autoformatting](#)
- [IntelliSense](#)
- [Code completion](#)
- [Code navigation](#)
- [Code structuring](#)
- [Find and replace](#)
- [Data pasting](#)
- [Syntax colorization](#)

Complete the following steps to start using the advanced formula editor in your instance of Microsoft Dynamics 365 Finance.

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
3. In the **User parameters** dialog box, in the **Execution tracing** section, set the **Enable advanced formula editor** parameter to **Yes**.

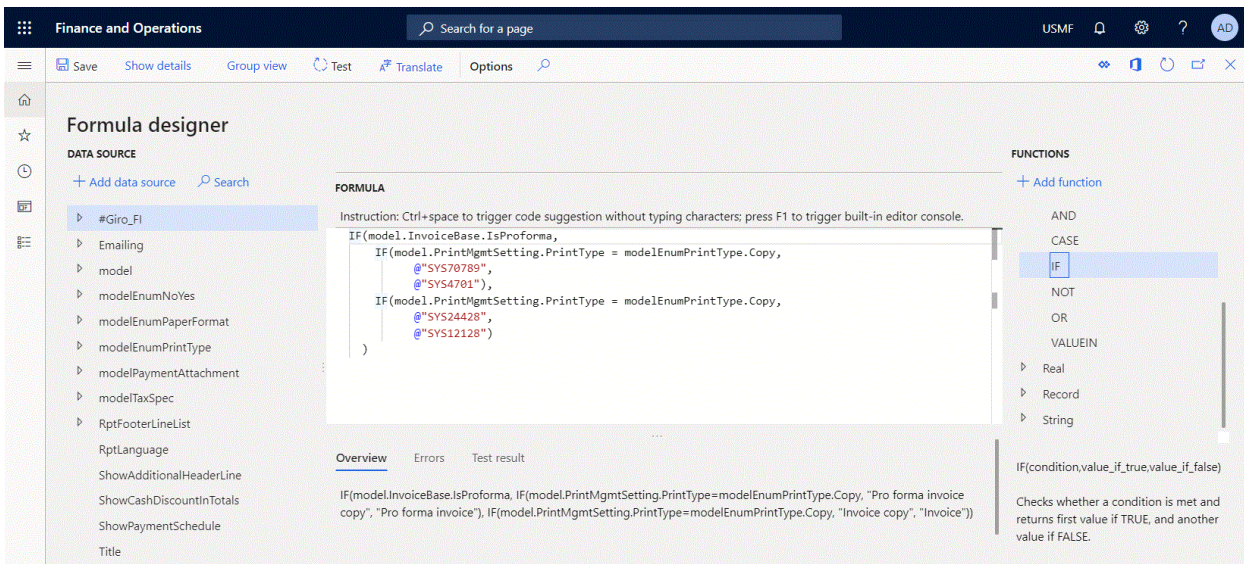


## NOTE

Be aware that this parameter is user specific and company specific.

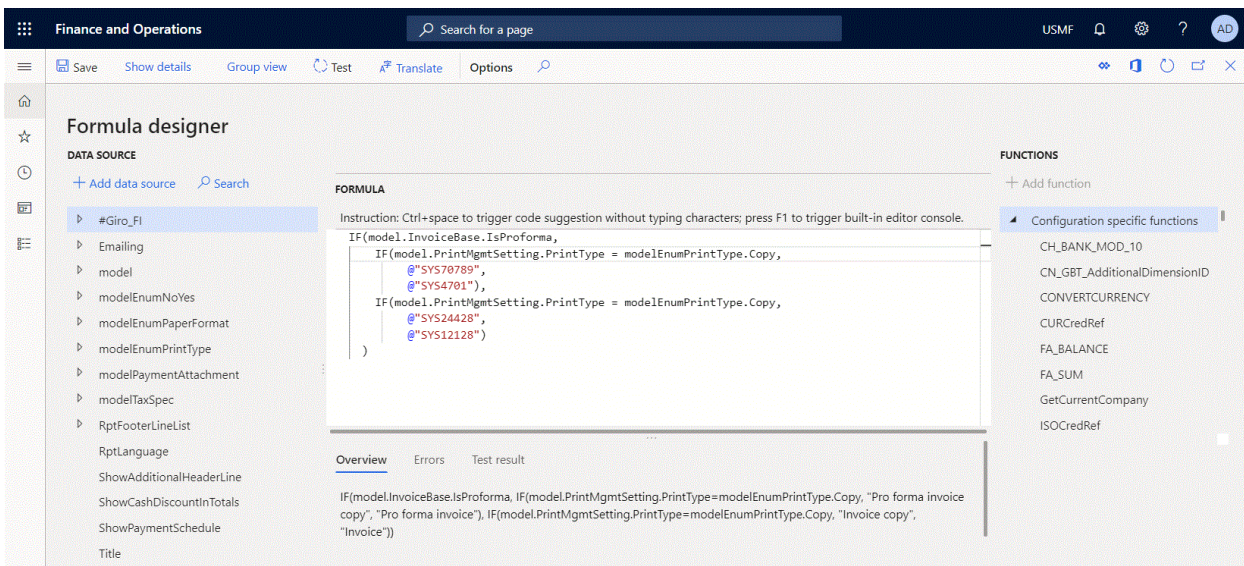
When you write a complex expression that consists of multiple rows of code, the indentation of a new entered line will be automatic based on the indentation of the previous row. You can select lines and change their indentation by typing **Tab** or **Shift+ Tab**.





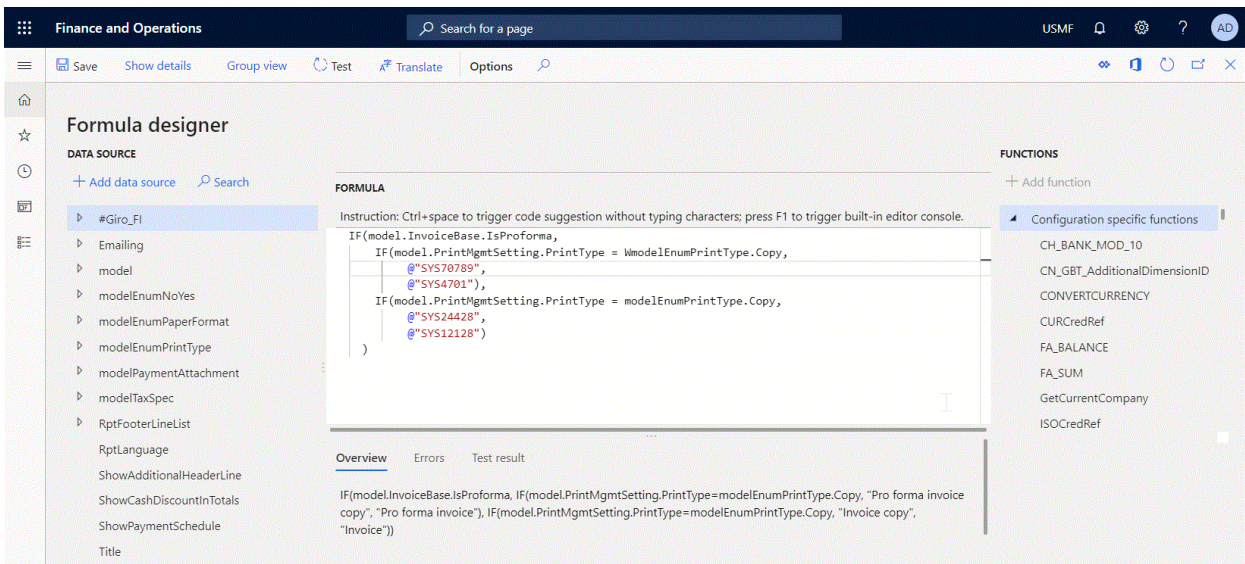
Autoformatting allows you to keep the entire expression well formatted to make further maintenance easier and to simplify understanding of the configured logic.

The editor provides word completion to help you write expression faster and avoid typos. When you start adding new text, the editor automatically offers a list of functions supported in ER functions that contain the characters you have entered. You can also trigger IntelliSense in any place of a configured expression by typing **Ctrl+Space**.



The editor automatically provides code completion by:

- Inserting a closing bracket when an opening bracket is entered, keeping the cursor inside the brackets.
- Inserting the second quotation symbol when the first one is entered, keeping the cursor inside the quotations.
- Inserting the second double quotation symbol when the first one is entered, keeping the cursor inside the quotations.

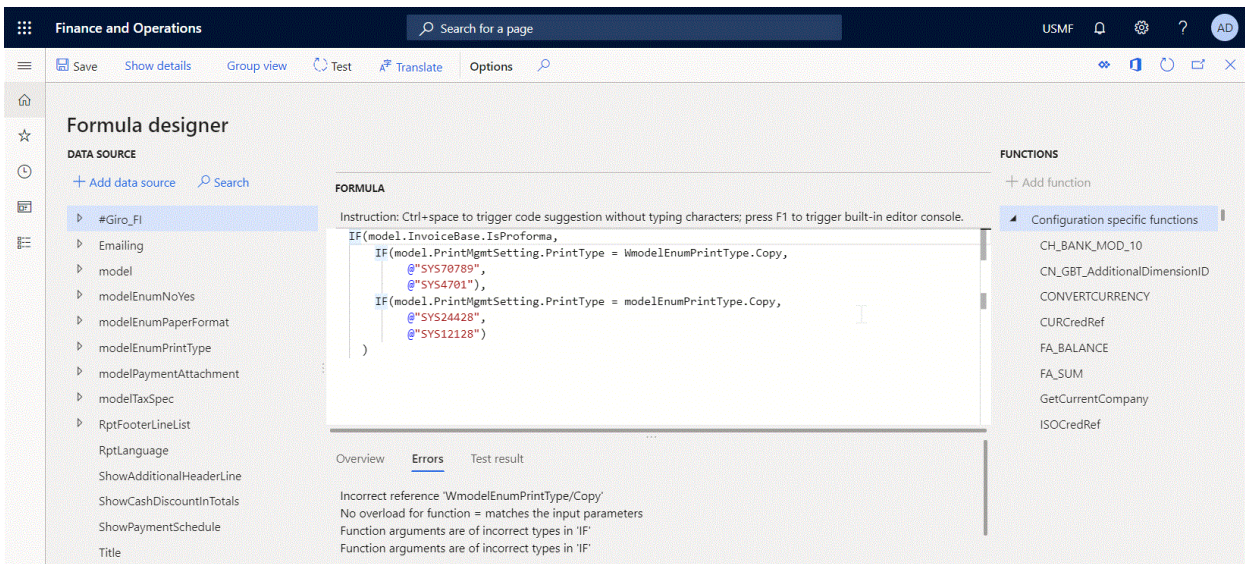


When you point to the typed bracket, the second bracket of this pair is automatically highlighted to show the construct that they support.

You can locate required symbols or lines in your expression by typing the Go to command using the command palette or the context menu.

For example, to jump to line 8, do the following:

- Press **Ctrl+G**, enter the value **8**, and then press **Enter**.
- or-
- Press **F1**, type **G**, select **Go to line**, enter the value **8**, and the press **Enter**.



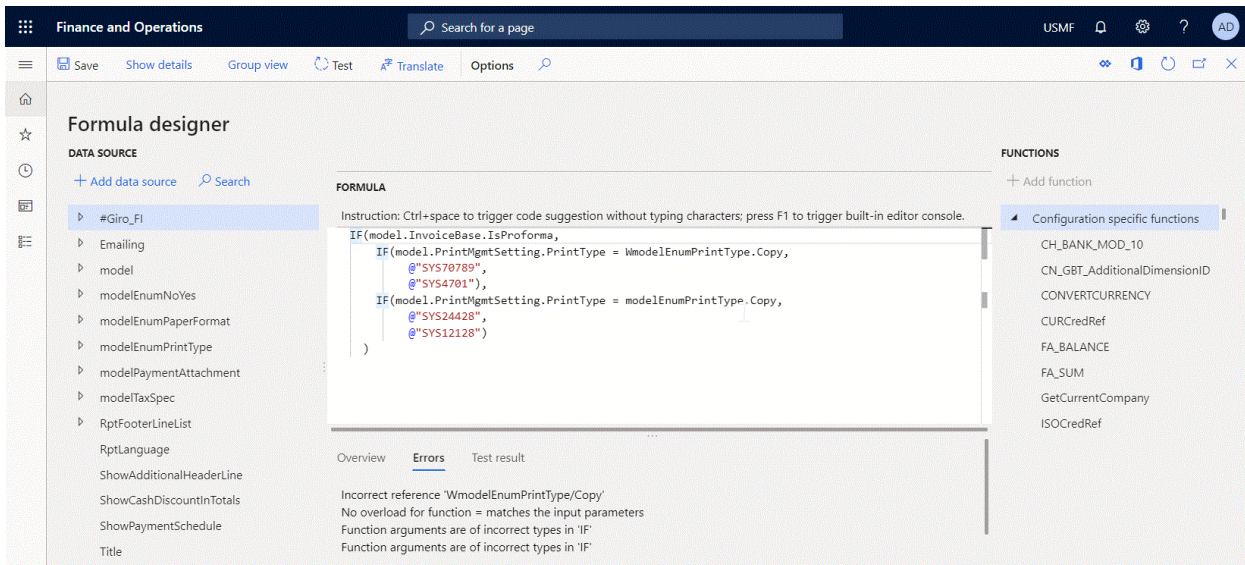
The code for some functions, such as **IF** or **CASE**, is automatically structured. You can expand and collapse any or all of the folding regions of this code to reduce the editable part of an expression in order to focus on only the piece of code that requires your attention. The toggle fold/unfold commands can be used for that.

For example, to fold all regions, do the following:

- Press **Ctrl+K**
- or-
- Press **F1**, press **FO**, select **Fold all**, and then press **Enter**

To unfold all regions, do the following:

- Press **Ctrl+J**
- or-
- Press **F1**, type **UN**, select **Unfold all**, and then press **Enter**



To find occurrences of certain text, select the text in your expression, and do the following:

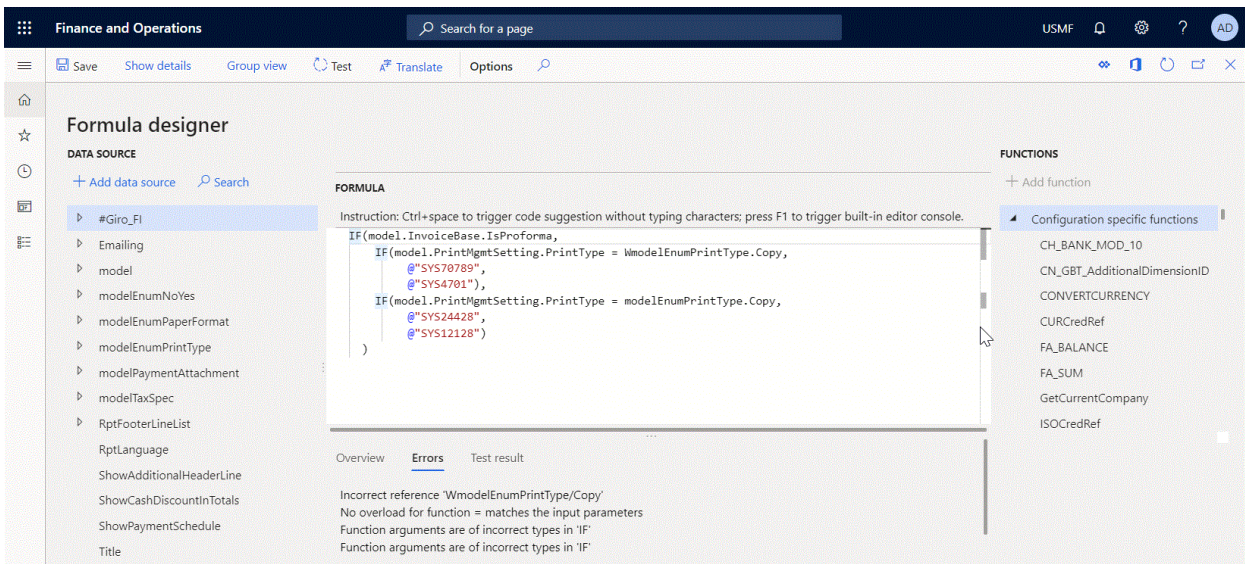
- Press **Ctrl+F** and then press **F3** to find the next occurrence of the selected text, or press **Shift+F3** to find the previous occurrence.
- or-
- Press **F1**, type **F**, and then select the required option to find the selected text.

To replace occurrences of a certain text, select the text in your expression, and do the following:

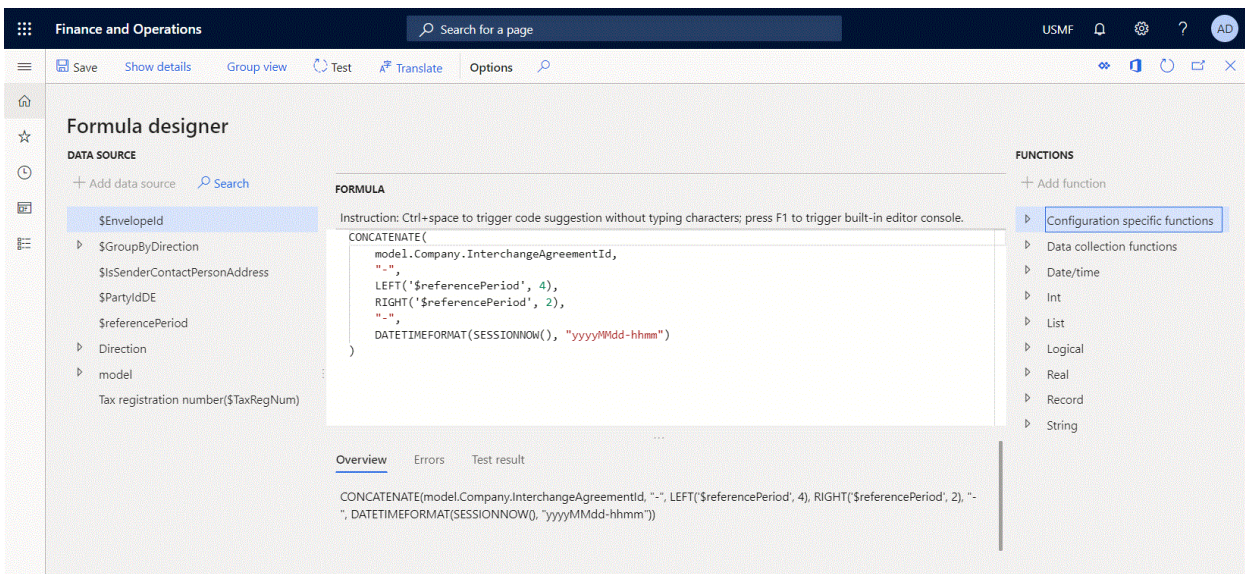
- Press **Ctrl+H**. Enter the alternative text and select the replacement option to replace either the selected text or all occurrences of this text in the current expression.
- or-
- Press **F1**, type **R**, and then select the required option to replace the selected text. Enter the alternative text and select the replacement option to replace either the selected text or all occurrences of this text in the current expression.

To change all occurrences of a certain text, select the text in your expression, and do the following:

- Press **Ctrl+F2** and then enter the alternative text.
- or-
- Press **F1**, type **C**, and then select the required option to change the selected text. Enter the alternative text.

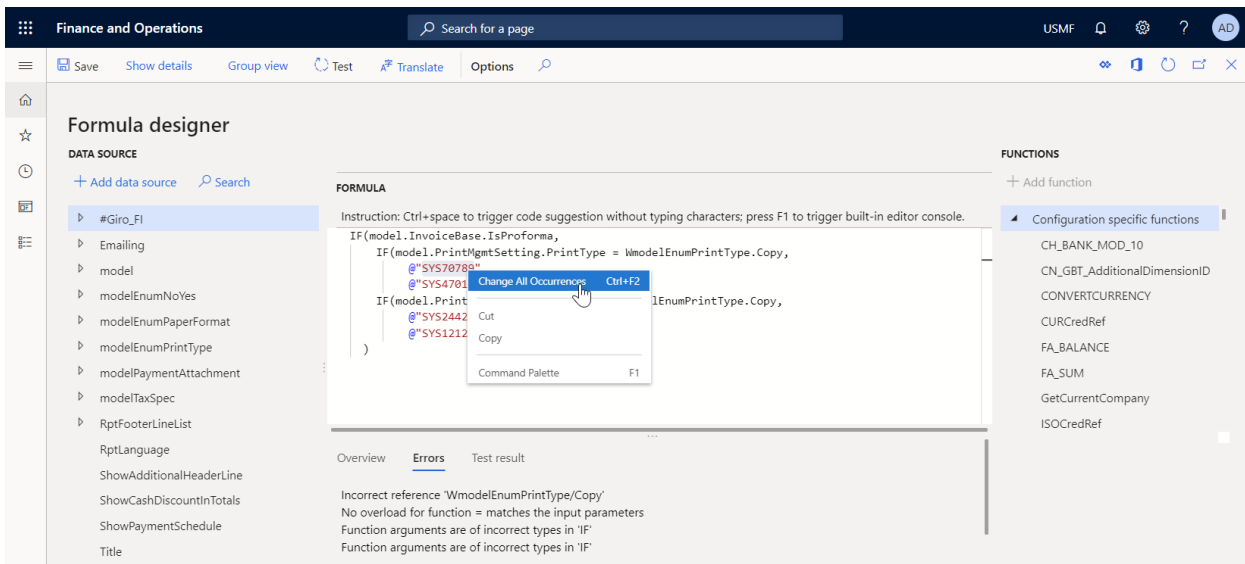


You can select **Add data source**, which pastes to the current expression a data source that is currently selected on the **Data source** left panel. Similarly, you can select **Add function**, which pastes to the current expression a function that is currently selected on the **Functions** right panel. If you use the ER formula editor, a selected function or a selected data source will always be pasted to the end of the configured expression. When you use the advanced ER formula editor, a selected function or a selected data source can be pasted to any part of the configured expression. You will need to use the cursor to specify where you want to paste the data.



Currently, different colors are used to highlight the following parts of expressions:

- The text in double brackets that can represent a label ID of a text constant.



## Limitations

The editor is currently supported in the following web browsers:

- Chrome
- Edge
- Firefox
- Opera
- Safari

## Additional resources

- [Electronic reporting \(ER\) overview](#)
- [Formula designer in Electronic reporting](#)
- [Monaco editor](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Electronic reporting formula language

2/18/2021 • 7 minutes to read • [Edit Online](#)

Electronic reporting (ER) provides a powerful data transformation experience. The language that is used to express the required data manipulations in the [ER formula designer](#) resembles the formula language in Microsoft Excel.

## Basic syntax

ER expressions can contain any or all of the following elements:

- [Constants](#)
- [Operators](#)
- [References](#)
- [Paths](#)
- [Functions](#)

When you design expressions, you can use text and numeric constants (that is, values that aren't calculated). For example, the expression `VALUE ("100") + 20` uses the numeric constant **20** and the string constant **"100"**, and it returns the numeric value **120**.

The ER formula designer supports escape sequences. Therefore, you can specify an expression string that should be handled differently. For example, the expression `"Leo Tolstoy ""War and Peace"" Volume 1"` returns the text string **Leo Tolstoy "War and Peace" Volume 1**.

The following table shows the arithmetic operators that you can use to do basic mathematical operations, such as addition, subtraction, multiplication, and division.

OPERATOR	MEANING	EXAMPLE
+	Addition	<code>1+2</code>
-	Subtraction, negation	<code>5-2</code> , <code>-1</code>
*	Multiplication	<code>7\*8</code>
/	Division	<code>9/3</code>

The following table shows the comparison operators that are supported. You can use these operators to compare two values.

OPERATOR	MEANING	EXAMPLE
=	Equal	<code>X=Y</code>
>	Greater than	<code>X&gt;Y</code>
<	Less than	<code>X&lt;Y</code>

OPERATOR	MEANING	EXAMPLE
>=	Greater than or equal to	<code>X&gt;=Y</code>
<=	Less than or equal to	<code>X&lt;=Y</code>
<>	Not equal to	<code>X&lt;&gt;Y</code>

Additionally, you can use an ampersand (&) as a text concatenation operator. In this way, you can join, or concatenate, one or more text strings into a single piece of text.

OPERATOR	MEANING	EXAMPLE
&	Concatenate	<code>"Nothing to print:" &amp; " " &amp; "no records found"</code>

### Operator precedence

The order in which the parts of a compound expression are evaluated is important. For example, the result of the expression `1 + 4 / 2` varies, depending on whether the addition or division operation is done first. You can use parentheses to explicitly define how an expression is evaluated. For example, to indicate that the addition operation should be done first, you can change the preceding expression to `(1 + 4) / 2`. If you don't explicitly indicate the order of operations in an expression, the order is based on the default precedence that is assigned to the supported operators. The following table shows the precedence that is assigned to each operator. Operators that have a higher precedence (for example, 7) are evaluated before operators that have a lower precedence (for example, 1).

PRECEDENCE	OPERATORS	SYNTAX
7	Grouping	<code>( ... )</code>
6	Member access	<code>... . ...</code>
5	Function call	<code>... ( ... )</code>
4	Multiplicative	<code>... * ...</code> <code>... / ...</code>
3	Additive	<code>... + ...</code> <code>... - ...</code>
2	Comparison	<code>... &lt; ...</code> <code>... &lt;= ...</code> <code>... = &gt; ...</code> <code>... &gt; ...</code> <code>... = ...</code> <code>... &lt;&gt; ...</code>
1	Separation	<code>... / ...</code>

If an expression includes multiple consecutive operators that have the same precedence, those operations are evaluated from left to right. For example, the expression `1 + 6 / 2 \* 3 > 5` returns **true**. We recommend that you use parentheses to explicitly indicate the desired order of operations in expressions, so that the expressions are easier to read and maintain.

All data sources of the current ER component that are available during the design of an expression can be used as named references. The current ER component can be either a model mapping or a format. For example, the current ER model mapping contains the **ReportingDate** data source, which returns a value of the *DateTime* data type. To correctly format that value in the generating document, you can reference the data source in the expression as `DATETIMEFORMAT (ReportingDate, "dd-MM-yyyy")`.

All characters in the name of a referencing data source that don't represent a letter of the alphabet must be preceded by a single quotation mark ('). If the name of a referencing data source contains at least one symbol that doesn't represent a letter of the alphabet, the name must be enclosed in single quotation marks. For example, these non-alphabetic symbols can be punctuation marks or other written symbols. Here are some examples:

- The **Today's date & time** data source must be referred to in an ER expression as `'Today''s date & time'`.
- The **name()** method of the **Customers** data source must be referred to in an ER expression as `Customers.'name()'`.

If the methods of application data sources have parameters, the following syntax is used to call those methods:

- If the **isLanguageRTL** method of the **System** data source has an **EN-US** parameter of the *String* data type, this method must be referred to in an ER expression as `System.isLanguageRTL("EN-US")`.
- Quotation marks aren't required when a method name contains only alphanumeric symbols. However, they are required for a method of a table if the name includes brackets.

When the **System** data source is added to an ER mapping that refers to the **Global** application class, the expression `System.isLanguageRTL("EN-US ")` returns the *Boolean* value **FALSE**. The modified expression `System.isLanguageRTL("AR")` returns the *Boolean* value **TRUE**.

You can limit the way that values are passed to the parameters of this type of method:

- Only constants can be passed to methods of this type. The values of the constants are defined at design time.
- Only primitive (basic) data types are supported for parameters of this type. The primitive data types include *Integer*, *Real*, *Boolean*, and *String*.

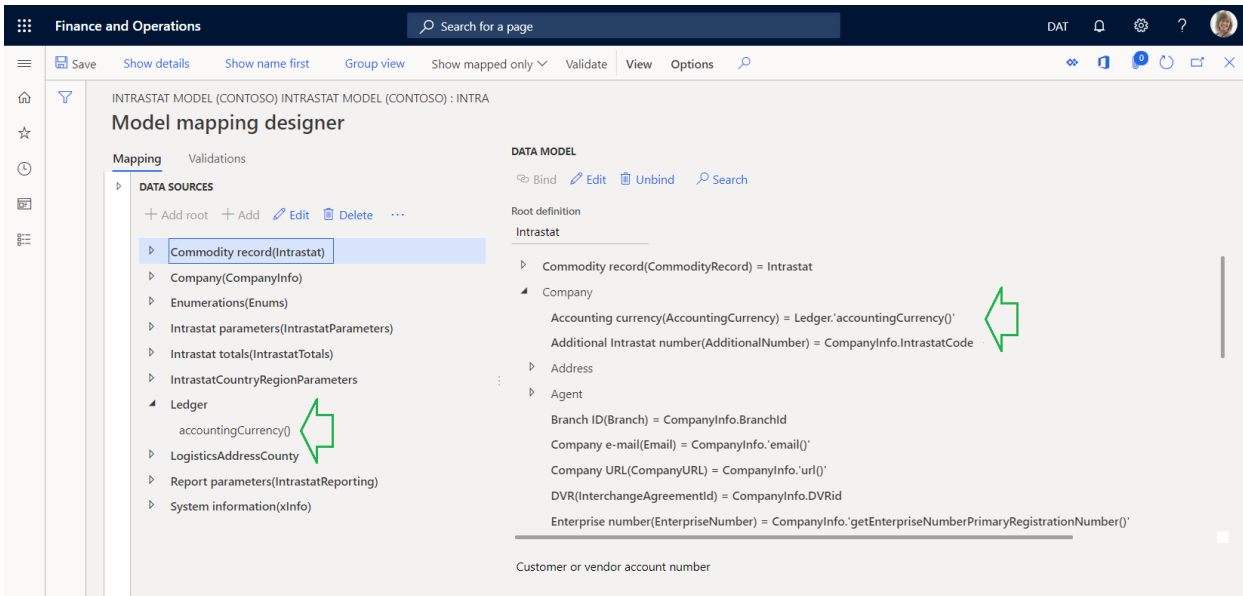
When an expression references a structured data source, you can use the path definition to select a specific primitive element of that data source. A dot character (.) is used to separate individual elements of a structured data source. For example, the current ER model mapping contains the **InvoiceTransactions** data source, and this data source returns a list of records. The **InvoiceTransactions** record structure contains the **AmountDebit** and **AmountCredit** fields, and both these fields return numeric values. Therefore, you can design the following expression to calculate the invoiced amount:

`InvoiceTransactions.AmountDebit - InvoiceTransactions.AmountCredit`. The `InvoiceTransactions.AmountDebit` construction in this expression is the path that is used to access the **AmountDebit** field of the **InvoiceTransactions** data source of the *Record list* type.

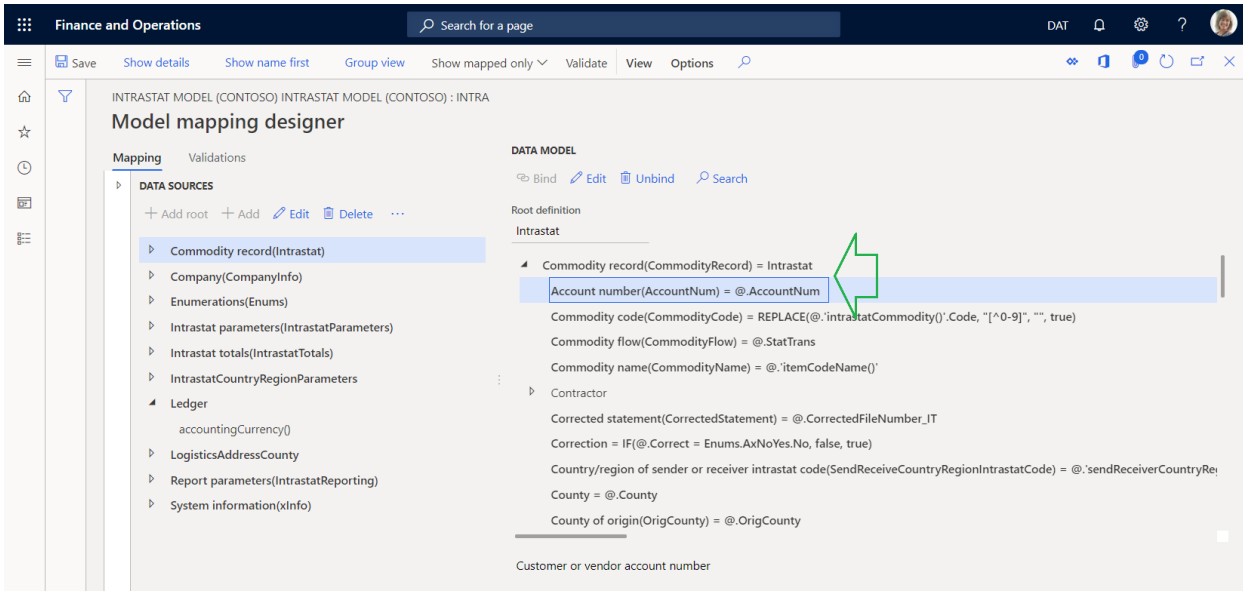
### Relative path

If the path of a structured data source starts with an "at" sign (@), it's a relative path. The "at" sign is shown instead of the remaining part of the absolute path of the hierarchical tree structure that is used. The following illustration shows an example. Here, the absolute path `Ledger.'accountingCurrency()'` indicates that the accounting currency value from the **Ledger** data source is entered in the **AccountingCurrency** field of the data model.

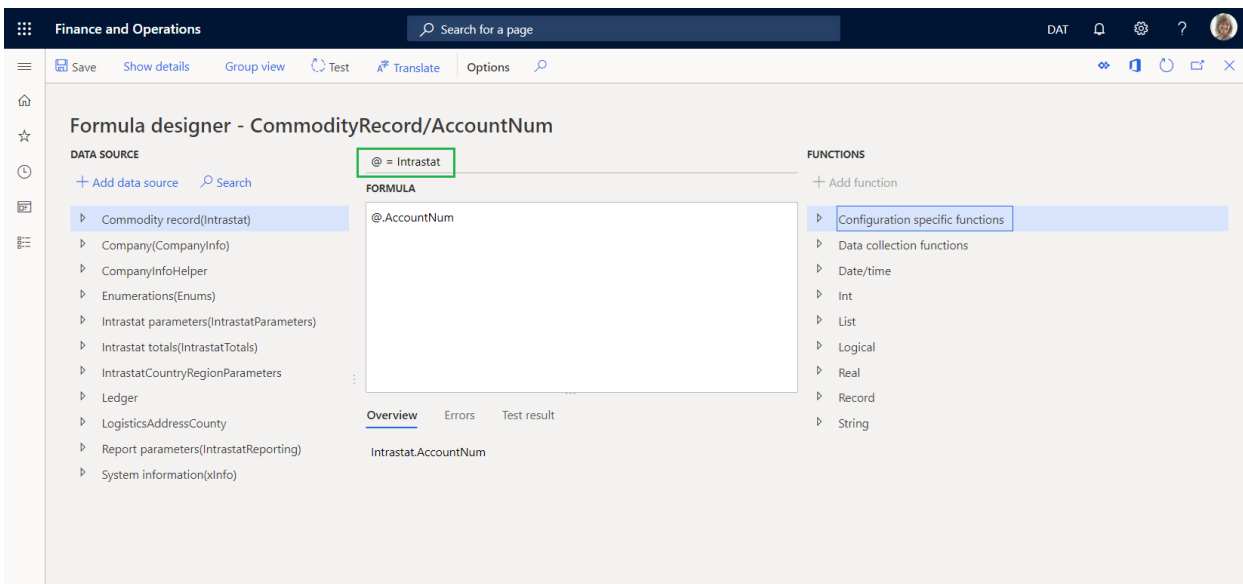




The example in the following illustration shows how a relative path is used. The relative path `@.AccountNum` indicates that the `AccountNum` field of the `Intrastat` data source (which appears one level above the `AccountNum` field in the data model's hierarchical tree) is used to enter the customer or vendor account number in the data model's `AccountNum` field.



The remaining part of the absolute path is also shown in the [ER formula editor](#).



For more information, see [Use a relative path in data bindings of ER models and formats](#).

ER built-in functions can be used in ER expressions. All data sources of the expression context (that is, the current ER model mapping or ER format) can be used as parameters of calling functions, in accordance with the list of arguments for calling functions. Constants can also be used as parameters of calling functions. For example, the current ER model mapping contains the **InvoiceTransactions** data source, and this data source returns a list of records. The **InvoiceTransactions** record structure contains the **AmountDebit** and **AmountCredit** fields, and both these fields return numeric values. Therefore, to calculate the invoiced amount, you can design the following expression that uses the built-in ER rounding function:

```
ROUND (InvoiceTransactions.AmountDebit - InvoiceTransactions.AmountCredit, 2) .
```

When you design ER model mappings and ER reports, you can use ER functions from the following categories:

- [Date and time functions](#)
- [List functions](#)
- [Logical functions](#)
- [Mathematical functions](#)
- [Record functions](#)
- [Text functions](#)
- [Data collection functions](#)
- [Other \(business domain–specific\) functions](#)
- [Type conversion functions](#)

## Functions list extension

ER lets you extend the list of functions that are used in ER expressions. Some engineering effort is required. For detailed information, see [Extend the list of Electronic reporting \(ER\) functions](#).

## Compound expressions

You can create compound expressions that use functions from different categories, provided that the data types match. When you use functions together, match the data type of the output from one function to the input data type that is required by another function. For example, to avoid a possible "list-is-empty" error in a binding of a field to an ER format element, combine functions from the [List](#) category with a function from the [Logical](#) category, as the following example shows. Here, the formula uses the [IF](#) function to test whether the **IntrastatTotals** list is empty before it returns the value of the required aggregation from that list. If the **IntrastatTotals** list is empty, the formula returns **0** (zero).

```
IF(ISEMPTY(IntrastatTotals), 0.0, IntrastatTotals.aggregated.'$AmountMSTRounded')
```

## Multiple solutions

Often, you can get the same data transformation result in multiple ways, by using functions from different categories or different functions from the same category. For example, the previous expression can also be configured by using the [COUNT](#) function from the [List](#) category.

```
IF(COUNT (IntrastatTotals)=0, 0.0, IntrastatTotals.aggregated.'$AmountMSTRounded')
```

## Additional resources

[Electronic Reporting overview](#)

Formula designer in Electronic reporting

Extend the list of Electronic reporting functions

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# List of ER functions in the Date and time category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) date and time functions can be used to extract information from date and time values, and to perform operations on them. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">AddDays</a>	This function returns a <i>DateTime</i> value that is the specified number of days before or after a specified start date.
<a href="#">DateFormat</a>	This function returns a <i>String</i> value that presents a given date value as text in the specified format and in an optionally specified culture.
<a href="#">DateTimeFormat</a>	This function returns a <i>String</i> value that presents a given date/time value as text in the specified format and in an optionally specified culture.
<a href="#">DateTimeValue</a>	This function returns a <i>DateTime</i> value that is converted from a given text value in the specified format and in an optionally specified culture to a date/time value.
<a href="#">DateToDateTime</a>	This function returns a <i>DateTime</i> value that is converted from a given date value to a date/time value in Coordinated Universal Time (Greenwich Mean Time [GMT]).
<a href="#">DateValue</a>	This function returns a <i>Date</i> value that is converted from a given text value in the specified format and in an optionally specified culture to a date value.
<a href="#">DayOfYear</a>	This function returns an <i>Integer</i> value that represents the number of days between January 1 and the specified date.
<a href="#">Days</a>	This function returns an <i>Integer</i> value that represents the number of days between one specified date and a second specified date.
<a href="#">Now</a>	This function returns a <i>DateTime</i> value that represents the current application server date and time.
<a href="#">NullDate</a>	This function returns a <i>Date</i> value that represents the <b>null</b> date (January 1, 1900).
<a href="#">NullDateTime</a>	This function returns a <i>DateTime</i> value that represents the <b>null</b> date/time value (January 1, 1900) in Coordinated Universal Time.
<a href="#">SessionNow</a>	This function returns a <i>DateTime</i> value that represents the current application session date and time.

FUNCTION	DESCRIPTION
<a href="#">SessionToday</a>	This function returns a <i>Date</i> value that represents the current application session date.
<a href="#">Today</a>	This function returns a <i>Date</i> value that represents the current application server date.

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

[Electronic reporting formula language](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ADDDAYS ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ADDDAYS` function calculates a *DateTime* value that is the specified number of days before or after a specified start date.

## Syntax

```
ADDDAYS (datetime, days)
```

## Arguments

`datetime` : *DateTime*

A date/time value that represents the start date.

`days` : *Integer*

The number of days before or after `datetime`.

## Return values

*DateTime*

The resulting date/time value.

## Usage notes

A positive value for `days` yields a future date. A negative value yields a past date.

## Example 1

`ADDDAYS (NOW(), 7)` returns the date and time seven days in the future.

## Example 2

`ADDDAYS (NOW(), -3)` returns the date and time three days in the past.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DATEFORMAT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATEFORMAT` function returns a *String* value that presents a given date value as text in the specified format and in an optionally specified *culture*. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
DATEFORMAT (date, format)
```

## Syntax 2

```
DATEFORMAT (date, format, culture)
```

## Arguments

`date` : *Date*

A date value that represents the date to format.

`format` : *String*

The format of the output string.

### NOTE

The format string is case-sensitive when you use either a standard format or a custom format. For example, the [standard](#) "d" format specifier returns the date by using the short date pattern, whereas the standard "D" format specifier returns the date by using the long date pattern. Additionally, the [custom](#) "M" format specifier returns the month from 1 through 12, whereas the custom "m" format specifier returns the minute from 0 through 59.

`culture` : *String*

The culture to use for formatting.

## Return values

*String*

The resulting string value.

## Usage notes

If the *culture* isn't defined as an argument of the called function, the value of `culture` is defined by the calling context. For example, if the `DATEFORMAT` function is called by using syntax 1 in an Electronic reporting (ER) format for a **FILE** element that is configured to use the German culture, the conversion will be done by using the German culture. The default `culture` value is **EN-US**.

## Example 1

`DATEFORMAT (TODAY (), "dd-MM-yyyy")` returns the current application server date, December 24, 2015, as the string "24-12-2015", based on the specified custom format.

## Example 2

`DATEFORMAT (SESSIONTODAY (), "d", "DE")` returns the current application session date, December 24, 2015, as the string "24-12-2015", based on the selected German culture and the specified format.

## Additional resources

### [Date and time functions](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# DATETIMEFORMAT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATETIMEFORMAT` function returns a *String* value that presents a given date/time value as text in the specified format and in an optionally specified *culture*. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
DATETIMEFORMAT (datetime, format)
```

## Syntax 2

```
DATETIMEFORMAT (datetime, format, culture)
```

## Arguments

`datetime` : *DateTime*

A date/time value that represents the date and time to format.

`format` : *String*

The format of the output string.

### NOTE

The format string is case-sensitive when you use either a standard format or a custom format. For example, the [standard](#) "d" format specifier returns the date by using the short date pattern, whereas the standard "D" format specifier returns the date by using the long date pattern. Additionally, the [custom](#) "M" format specifier returns the month from 1 through 12, whereas the custom "m" format specifier returns the minute from 0 through 59.

`culture` : *String*

The culture to use for formatting.

## Return values

*String*

The resulting string value.

## Usage notes

If the culture isn't defined as an argument of the called function, the value of `culture` is defined by the calling context. For example, if the `DATETIMEFORMAT` function is called by using syntax 1 in an Electronic reporting (ER) format for a **FILE** element that is configured to use the German culture, the conversion will be done by using the German culture. The default `culture` value is **EN-US**.

When the `DATETIMEFORMAT` function converts a given date/time value, it considers the time zone setting of the application user who is running the ER format that the function is called in the context of.

## Example 1

`DATETIMEFORMAT (NOW(), "dd-MM-yyyy")` returns the current application server date/time value, December 24, 2015, as "24-12-2015", based on the specified custom format.

## Example 2

`DATETIMEFORMAT (SESSIONNOW(), "d", "DE")` returns the current application session date/time value, December 24, 2015, as "24.12.2015", based on the selected German culture and the specified format.

## Example 3

`DATETIMEFORMAT (DATETIMEVALUE( "2019-11-12T09:00:00.0000000-07:00", "O"), "O")` returns the string value **2019-11-12T08:00:00.0000000-08:00** when the function is called during a process that was initiated by an application user who has the time zone value **(GMT-08:00) Pacific Time (US & Canada)** in the **Language and country/region preferences** section.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DATETIMEVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATETIMEVALUE` function returns a *DateTime* value that is converted from a given text value in the specified format and in an optionally specified *culture* to a date/time value. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
DATETIMEVALUE (text, format)
```

## Syntax 2

```
DATETIMEVALUE (text, format, culture)
```

## Arguments

`text` : *String*

Text that represents the value to format.

`format` : *String*

The format of the given text.

`culture` : *String*

The culture that is used for formatting of the given text.

## Return values

*DateTime*

The resulting date/time value.

## Usage notes

When the culture isn't defined as an argument of the called function, the value of `culture` is defined by the calling context. For example, if the `DATETIMEVALUE` function is called by using syntax 1 in an Electronic reporting (ER) format for a FILE element that is configured to use the German culture, the conversion will be done by using the German culture. The default `culture` value is EN-US.

## Example 1

`DATETIMEVALUE ("21-Dec-2016 02:55:00", "dd-MMM-yyyy hh:mm:ss")` returns 2:55:00 AM on December 21, 2016, based on the specified custom format and the default application's EN-US culture.

## Example 2

`DATETIMEVALUE ("21-Jan-2016 02:55:00", "dd-MMM-yyyy hh:mm:ss", "IT")` returns **2:55:00 AM on December 21, 2016**, based on the specified custom format and culture.

However, `DATETIMEVALUE ("21-Jan-2016 02:55:00", "dd-MMM-yyyy hh:mm:ss", "EN-US")` throws an exception to inform the user that the specified string isn't recognized as a valid date/time value for the specified culture.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DATEODATETIME ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATEODATETIME` function returns a *DateTime* value that is converted from a given date value to a date/time value in Coordinated Universal Time (Greenwich Mean Time [GMT]).

## Syntax

```
DATEODATETIME (date)
```

## Arguments

`date` : *Date*

A date value that represents the date to convert.

## Return values

*DateTime*

The resulting date/time value.

## Example 1

`DATEODATETIME (CompInfo. 'getCurrentDate()')` returns the date of the current Microsoft Dynamics 365 Finance session, December 24, 2015, as **12/24/2015 12:00:00 AM**. In this example, **CompInfo** is an Electronic reporting (ER) data source of the **Finance and Operations/Table** type, and it refers to the **CompanyInfo** table.

## Example 2

`DATEODATETIME (DATEVALUE ("2019-11-12T16:00:00.0000000-07:00", "0"))` returns the date/time value **11/12/2019 12:00:00 AM**.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DATEVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATEVALUE` function returns a *Date* value that is converted from a given text value in the specified format and in an optionally specified *culture* to a date value. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
DATEVALUE (text, format)
```

## Syntax 2

```
DATEVALUE (text, format, culture)
```

## Arguments

`text` : *String*

Text that represents the value to format.

`format` : *String*

The format of the given text.

`culture` : *String*

The culture that is used for formatting of the given text.

## Return values

*Date*

The resulting date value.

## Usage notes

When the culture isn't defined as an argument of the called function, the value of `culture` is defined by the calling context. For example, if the `DATEVALUE` function is called by using syntax 1 in an Electronic reporting (ER) format for a FILE element that is configured to use the German culture, the conversion will be done by using the German culture. The default `culture` value is **EN-US**.

## Example 1

`DATEVALUE ("21-Dec-2016", "dd-MMM-yyyy")` returns the date value **December 21, 2016**, based on the specified custom format and the default application's **EN-US** culture.

## Example 2

`DATEVALUE ("21-Jan-2016", "dd-MMM-yyyy", "IT")` returns the date value **January 21, 2016**, based on the specified custom format and culture.

However, `DATEVALUE ("21-Jan-2016", "dd-MMM-yyyy", "EN-US")` throws an exception to inform the user that the specified string isn't recognized as a valid date for the specified culture.

## Additional resources

### [Date and time functions](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DAYOFYEAR ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DAYOFYEAR` function returns an *Integer* value that represents the number of days between January 1 and the specified date.

## Syntax

```
DAYOFYEAR (date) as Integer
```

## Arguments

`date` : *Date*

A date value that represents the date to use for the calculation of the number of days.

## Return values

*Integer*

The resulting numeric value.

## Example 1

```
DAYOFYEAR (DATEVALUE ("01-03-2016", "dd-MM-yyyy")) returns 61.
```

## Example 2

```
DAYOFYEAR (DATEVALUE ("01-01-2016", "dd-MM-yyyy")) returns 1.
```

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# DAYS ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DAYS` function returns an *Integer* value that represents the number of days between one specified date and a second specified date.

## Syntax

```
DAYS (date 1, date 2) as Integer
```

## Arguments

`date 1` : *Date*

A date value that represents the start date for the calculation of the number of days.

`date 2` : *Date*

A date value that represents the end date for the calculation of the number of days.

## Return values

*Integer*

The resulting numeric value.

## Usage notes

The `DAYS` function returns a positive value when the first date is later than the second date, it returns **0** (zero) when the first date equals the second date, and it returns a negative value when the first date is earlier than the second date.

## Example

```
DAYS (TODAY (), DATEVALUE( DATETIMEFORMAT( ADDDAYS ( NOW(), 1), "yyyyMMdd"), "yyyyMMdd")) returns -1.
```

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NOW ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NOW` function returns a *DateTime* value that represents the current application server date and time.

## Syntax

```
NOW ( )
```

## Return values

*DateTime*

The resulting date/time value.

## Example

`DATETIMEFORMAT (NOW(), "dd-MM-yyyy")` returns the current application server date/time value, December 24, 2015, as "24-12-2015", based on the specified custom format.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NULLDATE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NULLDATE` function returns a *Date* value that represents the **null** date (January 1, 1900).

## Syntax

```
NULLDATE () as
```

## Return values

*Date*

The resulting date value.

## Example 1

`DATEFORMAT (NULLDATE(), "yyyy-MM-dd")` returns the **null** date, January 1, 1900, as "1900-01-01", based on the specified custom format.

## Example 2

The expression `IF( Invoice.DocumentDate = NULLDATE(), true, false)` returns **True** when the value of the **DocumentDate** field equals the **null** date. In this example, **Invoice** is an Electronic reporting (ER) data source of the **Finance/Table records** type, and it refers to the **CustInvoiceJour** table.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NULLDATETIME ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NULLDATETIME` function returns a *DateTime* value that represents the **null** date/time value (January 1, 1900) in Coordinated Universal Time (Greenwich Mean Time [GMT]).

## Syntax

```
NULLDATETIME ( )
```

## Return values

*DateTime*

The resulting date/time value.

## Example

`DATETIMEFORMAT( NULLDATETIME(), "0")` returns the string value `1900-01-01T00:00:00.0000000+00:00` when it's called during a process that was initiated by an application user who has the time zone value (GMT) Coordinated Universal Time in the **Language and country/region preferences** section.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SESSIONNOW ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SESSIONNOW` function returns a *DateTime* value that represents the current application session date and time.

## Syntax

```
SESSIONNOW ()
```

## Return values

*DateTime*

The resulting date/time value.

## Example

`DATETIMEFORMAT (SESSIONNOW(), "d", "DE")` returns the current application session date/time value, December 24, 2015, as "24.12.2015", based on the selected German culture and the specified format.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SESSIONTODAY ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SESSIONTODAY` function returns a *Date* value that represents the current application session date.

## Syntax

```
SESSIONTODAY ()
```

## Return values

*Date*

The resulting date value.

## Example

`DATEFORMAT (SESSIONTODAY (), "d", "DE")` returns the current application session date, December 24, 2015, as the string "24-12-2015", based on the selected German culture and the specified format.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# List of ER functions in the list category

2/18/2021 • 3 minutes to read • [Edit Online](#)

Electronic reporting (ER) list functions can be used to extract information from, and perform operations on, data sources of the *Record list* and *Container (record)* data types. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">AllItems</a>	This function runs as an in-memory selection. It returns a new flattened <i>Record list</i> value that consists of a list of records that represents all items that match the specified path.
<a href="#">AllItemsQuery</a>	This function runs as a joined SQL query. It returns a new flattened <i>Record list</i> value that consists of a list of records that represents all items that match the specified path.
<a href="#">Count</a>	This function returns an <i>Integer</i> value that represents the number of records in the specified list, if the list isn't empty. If the list is empty, this function returns 0 (zero).
<a href="#">EmptyList</a>	This function returns an empty <i>Record list</i> value by using the specified list as a source for the list structure.
<a href="#">Enumerate</a>	This function returns a new <i>Record list</i> value that consists of enumerated records of the specified list.
<a href="#">Filter</a>	This function returns the specified list as a <i>Record list</i> value after the query has been changed so that it filters for the specified condition.
<a href="#">First</a>	This function returns the first record of the specified list as a <i>Container (record)</i> value, if that list isn't empty. If the list is empty, this function throws an exception.
<a href="#">FirstOrDefault</a>	This function returns the first record of the specified list as a <i>Container (record)</i> value, if that record isn't empty. If the record is empty, this function returns a null <i>Container (record)</i> value.
<a href="#">Index</a>	This function returns a <i>Container (record)</i> value that is selected by using the specified numeric index in the specified list. If the index is out of range for the records in the specified list, this function throws an exception.
<a href="#">IsEmpty</a>	This function returns a <i>Boolean</i> value of <b>TRUE</b> if the specified list contains no records. Otherwise, it returns a <i>Boolean</i> value of <b>FALSE</b> .
<a href="#">List</a>	This function returns a <i>Record list</i> value that consists of a new list that is created from the specified arguments.

FUNCTION	DESCRIPTION
ListDistinct	This function calculates the specified expression as a selector for every record of the specified list. It returns a new <i>Record list</i> value that contains a single record for each unique selector value.
ListJoin	This function returns a <i>Record list</i> value that represents a new joined list that is created from the specified arguments.
ListOfFields	This function returns a <i>Record list</i> value that is created based on the structure of the specified argument of the <i>Enumeration</i> or <i>Container (record)</i> type.
ListOfFirstItem	This function returns a <i>Record list</i> value that consists of only the first record of the specified list.
OrderBy	This function returns the specified list as a <i>Record list</i> value after it has been sorted according to the specified arguments. These arguments can be defined as expressions.
Reverse	This function returns the specified list as a <i>Record list</i> value in reversed sort order.
Split	This function splits the specified input string into substrings and returns the result as a new <i>Record list</i> value.
SplitList	This function splits the specified list into sublists (or batches), each of which contains the specified number of records. It then returns the result as a new <i>Record list</i> value that consists of the batches.
SplitListByLimit	This function splits the specified list into a new list of sublists (batches). The number of records in each batch is dynamically calculated. The function then returns the result as a new <i>Record list</i> value that consists of the batches.
StringJoin	This function returns a <i>String</i> value that consists of concatenated values of the specified field from the specified list. The values can be separated by the specified delimiter.
Where	This function returns the specified list as a <i>Record list</i> value after it has been filtered according to the specified condition.

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

[Electronic reporting formula language](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# ALLITEMS ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ALLITEMS` function runs as an in-memory selection and returns a new flattened *Record list* value as a list of records that represents all items that match the specified path.

## Syntax

```
ALLITEMS (path)
```

## Arguments

`path` : *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The path must be defined as a valid data source path of a data source element of the *Record list* data type. Data elements such as the path string and date should raise an error in the Electronic reporting (ER) expression builder at design time.

We don't recommend that you use this function for transactional data sources that might contain a large volume of data. Instead, consider using the [ALLITEMSQUERY](#) function.

## Example 1

If you enter `SPLIT("abcdef" , 2)` as data source `DS`, the expression `COUNT( ALLITEMS (DS))` returns 3.

## Example 2

If you enter `Vend` as the data source of the *Record list* data type that refers to the `VendTable` application table, the expression `ALLITEMS (Vend.'<Relations'.ContactPerson)` returns a flattened list of records that has the `ContactPerson` table structure and contains all contact persons that can be accessed by using the `ContactPerson.ContactForParty == VendTable.Party` relation, and that is available for all vendors from the referenced vendor table.

## Additional resources

[List functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ALLITEMSQUERY ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ALLITEMSQUERY` function runs as a joined SQL query. It returns a new flattened *Record list* value that consists of a list of records that represent all items that match the specified path.

## Syntax

```
ALLITEMSQUERY (path)
```

## Arguments

`path` : *Record list*

The valid path of a data source of the *Record list* data type. It must contain at least one relation.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The specified path must be defined as a valid data source path of a data source element of the *Record list* data type. It must also contain at least one relation. Data elements such as the path *String* and *Date* should raise an error in the Electronic reporting (ER) expression builder at design time.

When this function is applied to data sources of the *Record list* data type that refer to an application object that can be directly called by using SQL (for example, an table, entity, or query), it runs as a joined SQL query. Otherwise, it runs in memory as the `ALLITEMS` function.

## Example

You define the following data sources in your model mapping:

- A `CustInv` data source of the *Table records* type that refers to the `CustInvoiceTable` table
- A `FilteredInv` data source of the *Calculated field* type that contains the expression

```
FILTER (CustInv, CustInv.InvoiceAccount = "US-001")
```

- A `JourLines` of the *Calculated field* type that contains the expression

```
ALLITEMSQUERY ( FilteredInv.'<Relations'.CustInvoiceJour.'<Relations'.CustInvoiceTrans)
```

When you run the model mapping to call the `JourLines` data source, the following SQL statement is run:

```
SELECT ... FROM CUSTINVOICETABLE T1 CROSS JOIN CUSTINVOICEJOUR T2 CROSS JOIN  
CUSTINVOICETRANS T3 WHERE...
```

## Additional resources

## List functions

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# COUNT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `COUNT` function returns an *Integer* value that represents the number of records in the specified list, if the list isn't empty. If the list is empty, this function returns **0** (zero).

## Syntax

```
COUNT (list)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Integer*

The resulting numeric value.

## Example

`COUNT (SPLIT("abcd" , 3))` returns 2, because the `SPLIT` function that is used in this example creates a list that consists of two records.

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# EMPTYLIST ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `EMPTYLIST` function returns an empty *Record list* value by using the specified list as a source for the list structure.

## Syntax

```
EMPTYLIST (list)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Record list*

The resulting list of records.

## Example

`EMPTYLIST (SPLIT ("abc", 1))` returns a new empty list that has the same structure as the list that is returned by the `SPLIT` function that is used.

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ENUMERATE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ENUMERATE` function returns a new *Record list* value that consists of enumerated records of the specified list.

## Syntax

```
ENUMERATE (list)
```

## Arguments

`list`: *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The list of enumerated records that is returned exposes the following additional elements:

- The record of fields (**Value** component)
- The current record index (**Number** component)

## Example

In the following illustration, an **Enumerated** data source is created as an enumerated list of vendor records from the **Vendors** data source that refers to the `VendTable` table.

```
└─ Enumerated: = ENUMERATE(Vendors):Record list
```

```
  └─ Number: Integer
```

```
    └─ Value: Record
```

```
      └─ Vendors: Table 'VendTable' records
```

The following illustration shows the Electronic reporting (ER) format. In this format, data bindings are created to generate output in XML format. This output presents individual vendors as enumerated nodes.

```
└─ root: XML Element
```

```
  └─ vendor: XML Element= Enumerated
```

```
    └─ name: XML Attribute= Enumerated.Value.'name()'
```

```
      └─ index: XML Attribute= Enumerated.Number
```

The following illustration shows the result when the designed format is run.

```
<?xml version="1.0" encoding="UTF-8"?>
- <root>
  <vendor index="1" name="Contoso Asia"/>
  <vendor index="2" name="Finanzamt Berlin"/>
  <vendor index="3" name="Opal Audio"/>
  <vendor index="4" name="Property Management"/>
  <vendor index="5" name="City-wide Advertising"/>
  <vendor index="6" name="Contoso Entertainment System"/>
</root>
```

## Additional resources

### List functions

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# FILTER ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `FILTER` function returns the specified list as a *Record list* value after the query has been changed so that it filters for the specified condition.

## Syntax

```
FILTER (list, condition)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`condition` : *Boolean*

A valid conditional expression that is used to filter records of the specified list.

## Return values

*Record list*

The resulting list of records.

## Usage notes

This function differs from the [WHERE](#) function, because the specified condition is applied to any Electronic reporting (ER) data source of the *Table records* type at the database level. The list and condition can be defined by using tables and relations.

If one or both arguments that are configured for this function (`list` and `condition`) don't allow this request to be translated to the direct SQL call, an exception is thrown at design time. This exception informs the user that either `list` or `condition` can't be used to query the database.

## Example 1

If **Vendor** is configured as an ER data source that refers to the VendTable table, the expression

```
FILTER (Vendors, Vendors.VendGroup = "40")
```

 returns a list of only vendors that belong to vendor group 40.

## Example 2

If **Vendor** is configured as an ER data source that refers to the VendTable table, and if **parmVendorBankGroup** is configured as an ER data source that returns a value of the *String* data type, the expression

```
FILTER ( Vendor.'<Relations'.VendBankAccount, Vendor.'<Relations'.VendBankAccount.BankGroupID = parmVendorBankGroup)
```

returns a list of only vendor accounts that belong to a specific bank group.

## Example 3

You enter data source *DS* of the *Calculated field* type, and it contains the expression `SPLIT ("A,B,C", ",")`. You then enter another expression, `FILTER( DS, DS.Value = "B")`. When you try to save this expression in the ER formula designer, the following exception is thrown: "Validation error: The list expression of FILTER function is not queryable."

## Additional resources

### List functions

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# FIRST ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `FIRST` function returns the first record of the specified list as a *Container (record)* value, if that list isn't empty. If the list is empty, this function throws an exception.

## Syntax

```
FIRST (list)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Container (record)*

The resulting record value.

## Example 1

The expression `FIRST(SPLIT("ABC",1)).Value` returns the text value "A".

## Example 2

The expression `FIRST(SPLIT("",1)).Value` throws an exception at runtime.

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# FIRSTORNULL ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `FIRSTORNULL` function returns the first record of the specified list as a *Container (record)* value, if that record isn't empty. If the record is empty, this function returns a null *Container (record)* value.

## Syntax

```
FIRSTORNULL (list)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Container (record)*

The resulting record value.

## Example

The expression `FIRSTORNULL(SPLIT("",1)).Value` returns an empty string ("").

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# INDEX ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `INDEX` function returns a *Container (record)* value that is selected by using the specified numeric index in the specified list. If the index is out of range for the records in the specified list, an exception is thrown.

## Syntax

```
INDEX (list, index)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`index` : *Integer*

A numeric index that indicates the position of the desired record in the specified list.

## Return values

*Container (record)*

The resulting record value.

## Example 1

If you enter data source `DS` of the *Calculated field* type, and it contains the expression `SPLIT ("A|B|C", "|")`, the expression `DS.Value` returns the text value "B" for the second record of this record list. The expression `INDEX (SPLIT ("A|B|C", "|"), 2).Value` also returns the text value "B".

## Example 2

If you enter data source `DS` of the *Calculated field* type, and it contains the expression `SPLIT ("A|B|C", "|")`, the expression `INDEX (SPLIT ("A|B|C", "|"), 4).Value` throws an exception at runtime.

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# IEMPTY ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `IEMPTY` function returns a *Boolean* value of **TRUE** if the specified list contains no records. Otherwise, it returns a *Boolean* value of **FALSE**.

## Syntax

```
IEMPTY (list)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Boolean*

The resulting *Boolean* value.

## Example 1

If you enter data source `DS` of the *Calculated field* type, and it contains the expression `SPLIT ("A|B|C", "|")`, the expression `IEMPTY(DS)` returns **FALSE**.

## Example 2

The expression `IEMPTY (SPLIT ("",1))` returns **TRUE**.

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# LIST ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `LIST` function returns a *Record list* value that consists of a new list of records that is created from the specified arguments.

## Syntax

```
LIST (record 1 [, record 2, ..., record N])
```

## Arguments

`record 1` : *Container (record)*

A reference to a data source of the *Record* data type. This argument is required.

`record N` : *Container (record)*

A reference to a data source of the *Record* data type. These additional arguments are optional.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The structure of the list that is created contains only the fields that are presented in the structure of every record that is mentioned in the arguments.

## Example

You enter data source **Record 1** of the *Container* type. This data source contains the following nested fields of the *Calculated field* type:

- **Code**: This field contains an expression that returns a value of the *String* type.
- **Amount**: This field contains an expression that returns a value of the *Real* type.

You then enter data source **Record 2** of the *Container* type. This data source contains the following nested fields of the *Calculated field* type:

- **Amount**: This field contains an expression that returns a value of the *Real* type.
- **IsValid**: This field contains an expression that returns a value of the *Boolean* type.

In this case, the expression `LIST('Record 1', 'Record 2')` returns a new list that contains two records. The structure of this list consists of a single **Amount** field of the *Real* type, because this field is the only field that is presented in every argument of the called function.

## Additional resources

## List functions

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# LISTJOIN ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `LISTJOIN` function returns a *Record list* value that represents a new joined list of records that is created from the specified arguments.

## Syntax

```
LIST (list 1 [, list 2, ..., list N])
```

## Arguments

`list 1` : *Record list*

A reference to a data source of the *Record list* data type. This argument is mandatory.

`list N` : *Record list*

A reference to a data source of the *Record list* data type. These additional arguments are optional.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The structure of the list that is created contains only the fields that are present in the structure of every record list that is referenced in the arguments.

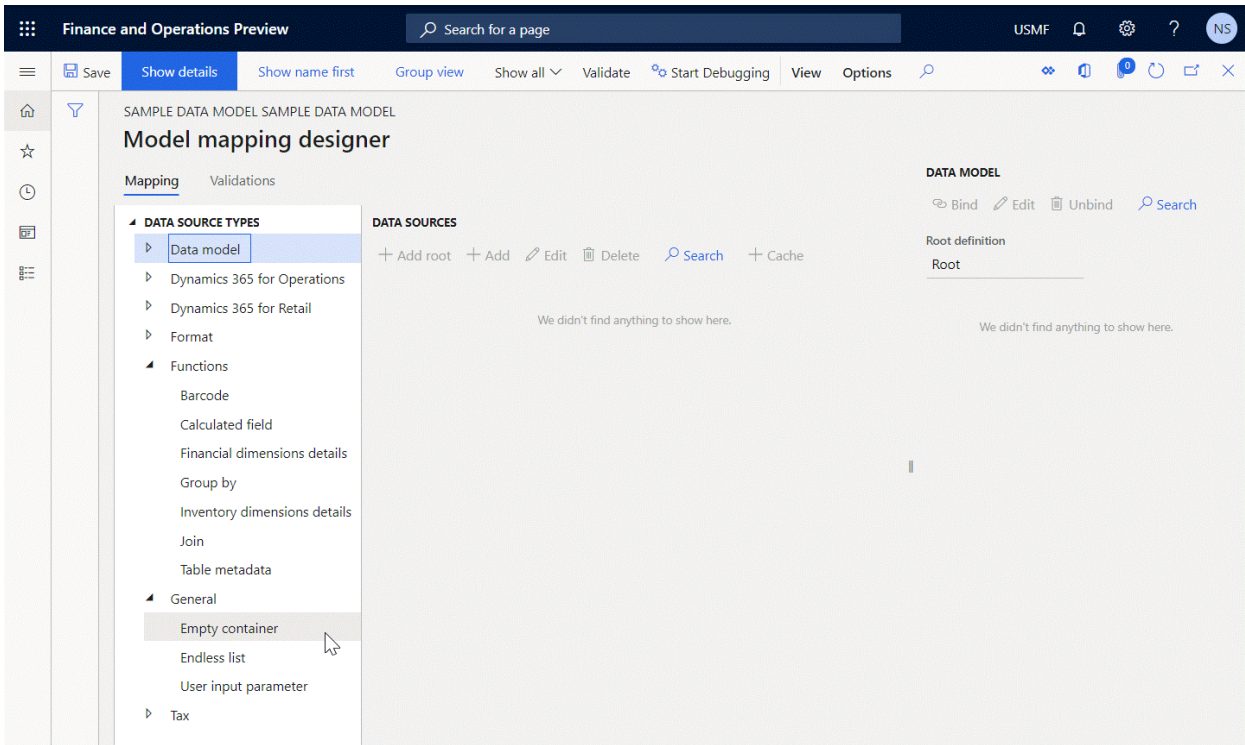
## Example

You enter data source **Record 1** of the `Container` type. This data source contains the following nested fields of the `Calculated field` type:

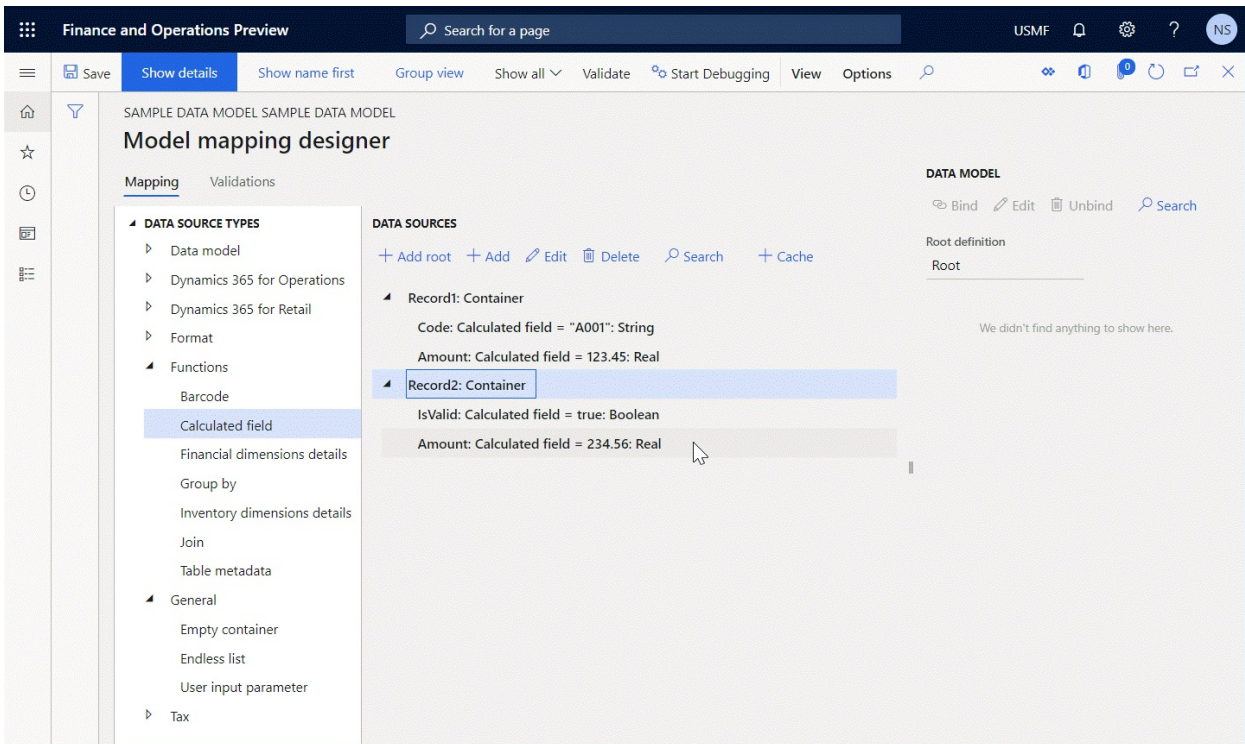
- **Code**: This field contains an expression that returns a value of the `String` type.
- **Amount**: This field contains an expression that returns a value of the `Real` type.

You then enter data source **Record 2** of the `Container` type. This data source contains the following nested fields of the `Calculated field` type:

- **Amount**: This field contains an expression that returns a value of the `Real` type.
- **IsValid**: This field contains an expression that returns a value of the `Boolean` type.



In this case, the expression `LISTJOIN(LIST('Record 1'), LIST('Record 2'))` returns a new list that contains two records.



The structure of this list consists of a single **Amount** field of the `Real` type, because this field is the only field that is presented in every argument of the called function.

## Additional resources

[List functions](#)

[Debug data sources of an executed ER format to analyze data flow and transformation](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# LISTOFFIELDS ER function

2/18/2021 • 3 minutes to read • [Edit Online](#)

The `LISTOFFIELDS` function returns a *Record list* value that is created based on the structure of the specified argument of the *Enumeration* or *Container (record)* type.

## Syntax 1

```
LISTOFFIELDS (path)
```

## Syntax 2

```
LISTOFFIELDS (path, language)
```

## Arguments

`path` : Data source reference

The valid reference path of a data source of one of the following data types:

- Model enumeration
- Format enumeration
- Application enumeration
- Container (record)

`language` : *String*

Text that represents a language code.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The list that is created consists of records that have the following fields:

- **Name** (*String* data type)
- **Label** (*String* data type)
- **Description** (*String* data type)
- **IsTranslated** (*Boolean* data type)

If the `path` argument refers to a data source of the *Container (Record)* type, for every field of the referenced container record, a new record is added to the list that is created. For every record that is created, the **Name** field returns the name of the field of the referenced container record that the current record was created for.

If the `path` argument refers to a data source of one of the *Enumeration* types, for every enumeration value of

the referenced enumeration, a new record is added to the list that is created. For every record that is created, the **Name** field returns the value of the referenced enumeration that the current record was created for, the **Description** field returns the description of that enumeration, and the **Label** field returns the label of that enumeration.

At runtime, when syntax 1 is used, the **Label** and **Description** fields must return values that are based on the language settings of the Electronic reporting (ER) format that is running:

- If the labels and descriptions for the requested language are available, the **Label** and **Description** fields return values that are based on that language, and the **IsTranslated** field returns **True**.
- If the labels and descriptions for the requested language aren't available, the **Label** and **Description** fields return values that are based on the default EN-US language, and the **IsTranslated** field returns **False**.

At runtime, when syntax 2 is used, the **Label** and **Description** fields must return values that are based on the language that is defined as the second argument of the called function:

- If the labels and descriptions for the requested language are available, the **Label** and **Description** fields return values that are based on that language, and the **IsTranslated** field returns **True**.
- If the labels and descriptions for the requested language aren't available, the **Label** and **Description** fields return values that are based on the **EN-US** language, and the **IsTranslated** field returns **False**.

## Example 1

In the following illustration, an enumeration is introduced in an ER data model.

ReportDirection		
Enumeration values		
Name	Label (*Recommended to use labels)	Description
Both	Both	
Export	Dispatches	
Import	Arrivals	

The following illustration shows these details:

- The model enumeration is inserted into a report as a data source.
- An ER expression uses the model enumeration as a parameter of the `LISTOFFIELDS` function.
- A data source of the *Record list* type is inserted into a report by using the ER expression that is created.

FORMAT   **MAPPING**   TRANSFORMATIONS   VALIDATIONS

Bind   + Add root   + Add   Edit   Delete   Show name first   Group view

- enumDirectionInReport: Data model enumeration ReportDirection
  - Arrivals(Import): Enumeration value
  - Both: Enumeration value
  - Dispatches(Export): Enumeration value
- listDirectionInReport: Calculated field = LISTOFFIELDS(enumDirectionInReport): Record list**
  - Description: String
  - Label: String
  - Name: String
- model: Data model Intrastat

The following example shows the ER format elements that are bound to the data source of the *Record list* type that was created by using the `LISTOFFIELDS` function.

- Root: XML Element
  - Directions: XML Element = listDirectionInReport
    - name: XML Attribute = listDirectionInReport.Name
    - label: XML Attribute = listDirectionInReport.Label
    - desc: XML Attribute = listDirectionInReport.Description

The following illustration shows the result when the designed format is run.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Root>
  <Directions desc="" label="Both" name="Both"/>
  <Directions desc="" label="Dispatches" name="Export"/>
  <Directions desc="" label="Arrivals" name="Import"/>
</Root>
```

**NOTE**

Based on the language settings of the parent FILE and FOLDER format elements, translated text for labels and descriptions is entered in the output of the ER format.

## Example 2

You use the *Calculated field* data source type to configure `enumType_de` and `enumType_deCH` data sources for the `enumType` data model enumeration:

- `enumType_de` = LISTOFFIELDS (enumType, "de")
- `enumType_deCH` = LISTOFFIELDS (enumType, "de-CH")

In this case, you can use the following expression to get the label of the enumeration value in Swiss German, if that translation is available. If the Swiss German translation isn't available, the label is in German.

```
IF (NOT (enumType_deCH.IsTranslated), enumType_de.Label, enumType_deCH.Label)
```

## Additional resources

[List functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# LISTOFFIRSTITEM ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `LISTOFFIRSTITEM` function returns a *Record list* value that consists of only the first record of the specified list.

## Syntax

```
LISTOFFIRSTITEM (list)
```

## Arguments

`list`: *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Record list*

The resulting list of records.

## Example

The expression `FIRST( LISTOFFIRSTITEM ( SPLIT ("ABC",1))).Value` returns the text value "A".

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ORDERBY ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ORDERBY` function returns the specified list as a *Record list* value after it has been sorted according to the specified arguments. These arguments can be defined as expressions.

## Syntax

```
ORDERBY (list , expression 1[, expression 2, ..., expression N])
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`expression 1` : *Field*

The valid path of a field of the data source that is referenced by the `list` argument of the called function. The referenced field must be a field of the primitive data type. This argument is required.

`expression N` : *Field*

The valid path of a field of the data source that is referenced by the `list` argument of the called function. The referenced field must be a field of the primitive data type. These additional arguments are optional.

## Return values

*Record list*

The resulting list of records.

## Example 1

If you enter data source `DS` of the *Calculated field* type, and it contains the expression `SPLIT ("C|B|A", "|")`, the expression `FIRST( ORDERBY( DS, DS. Value)).Value` returns the text value "A".

## Example 2

If `Vendor` is configured as an Electronic reporting (ER) data source that refers to the `VendTable` table, the expression `ORDERBY (Vendors, Vendors.'name()')` returns a list of vendors that is sorted by name in ascending order.

## Additional resources

[List functions](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# REVERSE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `REVERSE` function returns the specified list as a *Record list* value in reversed sort order.

## Syntax

```
REVERSE (list)
```

## Arguments

`list`: *Record list*

The valid path of a data source of the *Record list* data type.

## Return values

*Record list*

The resulting list of records.

## Example 1

If you enter data source `DS` of the *Calculated field* type, and it contains the expression `SPLIT ("C|B|A", "|")`, the expression `FIRST( REVERSE( ORDERBY( DS, DS. Value)))` returns the text value "C".

## Example 2

If `Vendor` is configured as an Electronic reporting (ER) data source that refers to the `VendTable` table, the expression `REVERSE (ORDERBY (Vendors, Vendors.'name()'))` returns a list of vendors that is sorted by name in descending order.

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SPLIT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SPLIT` function splits the specified input string into substrings and returns the result as a new *Record list* value.

## Syntax 1

```
SPLIT (input, length)
```

This syntax is used to split the specified input string into substrings, each of which has the specified length.

## Syntax 2

```
SPLIT (input, delimiter)
```

This syntax is used to split the specified input string into substrings, based on the specified delimiter.

## Arguments

`input` : *String*

The text to split.

`length` : *Integer*

The maximum length of a single substring.

`delimiter` : *String*

A delimiter that is used to separate substrings.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The record structure of the list that is returned consists of the **Value** field of the *String* type. Every record of the list that is returned contains generated substrings in this field.

If the `delimiter` argument is empty, the new list that is returned consists of one record that has the **Value** field of the *String* type. This field contains the input text.

If the `input` argument is empty, a new empty list is returned. If either the `input` or `delimiter` argument is unspecified (null), an application exception is thrown.

## Example 1

`SPLIT ("abcd", 3)` returns a new list that consists of two records that have the **Value** field of the *String* type. The **Value** field in the first record contains the text "abc", and the **Value** field in the second record contains the text "d".

## Example 2

`SPLIT ("XAb aBy", "aB")` returns a new list that consists of three records that have the **Value** field of the *String* type. The **Value** field in the first record contains the text "X", the **Value** field in the second record contains the text " ", and the **Value** field in the third record contains the text "y".

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SPLITLIST ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SPLITLIST` function splits the specified list into sublists (or batches), each of which contains the specified number of records. It then returns the result as a new *Record list* value that consists of the batches.

## Syntax

```
SPLITLIST (list, number)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`number` : *Integer*

The maximum number of records per batch.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The list of batches that is returned contains the following elements:

- **Value:** *List*

The list of records that belong to the current batch.

- **BatchNumber:** *Integer*

The number of the current batch in the returned list.

## Example

In the following illustration, a **Lines** data source is created as a record list that has three records. This list is divided into batches, each of which contains up to two records.

```
← Lines: = SPLITLIST(SPLIT("abcdef", 2), 2):Record list
```

```
    BatchNumber: Integer
```

```
    ← Value: Record list
```

```
        Value: String
```

The following illustration shows the designed format layout. In this format layout, bindings to the **Lines** data source are created to generate output in XML format. This output presents individual nodes for each batch and the records in it.

- └─ root: XML Element
  - └─ batch: XML Element= Lines
    - number: XML Attribute= Lines.BatchNumber
  - └─ record: XML Element= Lines.Value
    - value: XML Attribute= Lines.Value.Value

The following illustration shows the result when the designed format is run.

```
<?xml version="1.0" encoding="UTF-8"?>
- <root>
  - <batch number="1">
    <record value="ab"/>
    <record value="cd"/>
  </batch>
  - <batch number="2">
    <record value="ef"/>
  </batch>
</root>
```

## Additional resources

### List functions

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SPLITLISTBYLIMIT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SPLITLISTBYLIMIT` function splits the specified list into a new list of sublists (batches). The number of records in each batch is dynamically calculated. The function then returns the result as a new *Record list* value that consists of the batches.

## Syntax

```
SPLITLISTBYLIMIT (list, limit value, limit source)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`limit value` : *Integer or Real*

The maximum value of the limit that is used to split the original list into batches.

`limit source` : *Field*

The valid path of a field of the *Integer* or *Real* type in the specified list. The value of this field defines the step that the total sum is increased on.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The list of batches that is returned contains the following elements:

- **Value:** *List*

The list of records that belong to the current batch.

- **BatchNumber:** *Integer*

The number of the current batch in the returned list.

The limit isn't applied to a single item of the original list if the limit source exceeds the defined limit.

## Example

The following illustration shows an Electronic reporting (ER) format.

Bind Unbind Edit formula Expand/collapse

- Root: XML Element
  - ListOfItems: XML Element
    - Item: XML Element = model.Item
      - Name: XML Attribute = model.Item.Name
      - Weight: XML Attribute = model.Item.Weight

The following illustration shows the data sources that are used for the format.

FORMAT **MAPPING** TRANSFORMATIONS

Bind + Add root + Add Edit

- model: Data model Intrastat
  - Item: Record list
    - Name: String
    - Weight: Real

The following illustration shows the result when the format is run. In this case, the output is a flat list of commodity items.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Root>
  - <ListOfItems>
    <Item Weight="1" Name="Speaker"/>
    <Item Weight="2" Name="Projector"/>
    <Item Weight="3" Name="Radio receiver"/>
    <Item Weight="4" Name="Monitor"/>
    <Item Weight="5" Name="Transmitter or receiver"/>
    <Item Weight="6" Name="Parts and components"/>
    <Item Weight="11" Name="Hardware"/>
  </ListOfItems>
</Root>
```

In the following illustrations, the same format has been adjusted so that it presents the list of commodity items in batches if a single batch must include commodities and the total weight should not exceed a limit of 9.

Bind Unbind Edit formula Expand/collapse Search

- Root: XML Element
  - ListOfItems: XML Element
    - Item: XML Element = model.Item
      - Name: XML Attribute = model.Item.Name
      - Weight: XML Attribute = model.Item.Weight
  - ListOfBatches: XML Element
    - Batch: XML Element = batches
      - Id: XML Attribute = batches.BatchNumber
      - TotalWeight: XML Attribute = batches.total.aggregated.total
    - ListOfItems: XML Element
      - Item: XML Element = batches.Value
        - Name: XML Attribute = batches.Value.Name
        - Weight: XML Attribute = batches.Value.Name



4 batches: Calculated field = SPLITLISTBYLIMIT(model.Item, 9, model.Item.Weight): Record list

BatchNumber: Integer

4 total: Record list 'batches/Value' group by

4 aggregated: Record

total: Real

grouped: Record

▸ lines: Record list

4 Value: Record list

Name: String

Weight: Real

4 model: Data model Intrastat

4 Item: Record list

Name: String

Weight: Real

The following illustration shows the result when the adjusted format is run.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Root>
  - <ListOfItems>
    <Item Weight="1" Name="Speaker"/>
    <Item Weight="2" Name="Projector"/>
    <Item Weight="3" Name="Radio receiver"/>
    <Item Weight="4" Name="Monitor"/>
    <Item Weight="5" Name="Transmitter or receiver"/>
    <Item Weight="6" Name="Parts and components"/>
    <Item Weight="11" Name="Hardware"/>
  </ListOfItems>
  - <ListOfBatches>
    - <Batch TotalWeight="6.0" Id="1">
      - <ListOfItems>
        <Item Weight="Speaker" Name="Speaker"/>
        <Item Weight="Projector" Name="Projector"/>
        <Item Weight="Radio receiver" Name="Radio receiver"/>
      </ListOfItems>
    </Batch>
    - <Batch TotalWeight="9.0" Id="2">
      - <ListOfItems>
        <Item Weight="Monitor" Name="Monitor"/>
        <Item Weight="Transmitter or receiver" Name="Transmitter or receiver"/>
      </ListOfItems>
    </Batch>
    - <Batch TotalWeight="6.0" Id="3">
      - <ListOfItems>
        <Item Weight="Parts and components" Name="Parts and components"/>
      </ListOfItems>
    </Batch>
    - <Batch TotalWeight="11.0" Id="4">
      - <ListOfItems>
        <Item Weight="Hardware" Name="Hardware"/>
      </ListOfItems>
    </Batch>
  </ListOfBatches>
</Root>
```

**NOTE**

The limit isn't applied to the last item of the original list, because the value (11) of the limit source (**weight**) exceeds the defined limit (9). To ignore sublists during report generation, use either the `WHERE` function or the **Enabled** expression of the corresponding format element, as you require.

## Additional resources

### List functions

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# STRINGJOIN ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `STRINGJOIN` function returns a *String* value that consists of concatenated values of the specified field from the specified list. The values can be separated by the specified delimiter.

## Syntax

```
STRINGJOIN (list, field, delimiter)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`field` : *Field*

The valid path of a field of the *String* data type in the specified list.

`delimiter` : *String*

A delimiter that is used to separate substrings.

## Return values

*String*

The resulting text value.

## Example

If you enter `SPLIT("abc" , 1)` as data source `DS`, the expression `STRINGJOIN (DS, DS.Value, "-")` returns "a-b-c".

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# WHERE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `WHERE` function returns the specified list as a *Record list* value after it has been filtered according to the specified condition.

## Syntax

```
WHERE (list, condition)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`condition` : *Boolean*

A valid conditional expression that is used to filter records of the specified list.

## Return values

*Record list*

The resulting list of records.

## Usage notes

This function differs from the `FILTER` function, because the specified condition is applied to any Electronic reporting (ER) data source of the *Record list* type that is present in memory.

If the arguments that are configured for this function (`list` and `condition`) allow this request to be translated to the direct SQL call, a warning message is thrown at design time. This message informs the user that performance might be improved if the `FILTER` function is used instead of `WHERE`.

## Example 1

If `Vendor` is configured as an ER data source that refers to the `VendTable` table, the expression

```
WHERE (Vendors, Vendors.VendGroup = "40")
```

 returns a list of only vendors that belong to vendor group 40.

## Example 2

If you enter data source `DS` of the *Calculated field* type, and it contains the expression `SPLIT ("A|B|C", "|")`, the expression `WHERE( DS, DS.Value = "B")` returns a list of only one record that contains the text `"B"` in the `Value` field.

## Additional resources

[List functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# List of ER functions in the logical category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) logical functions can be used to work with logical values to perform more than one comparison in a single expression or test multiple conditions. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">And</a>	This function returns a <i>Boolean</i> value of <b>TRUE</b> if all the specified conditions are true. Otherwise, it returns a <i>Boolean</i> value of <b>FALSE</b> .
<a href="#">Case</a>	This function evaluates the value of the specified expression against the specified alternative options and returns the result of the first option that equals the value of the specified expression. Otherwise, it returns an optional default result, if a default result is specified as the last argument of the called function that isn't preceded by an option. The value that is returned can be a value of any of the supported data types.
<a href="#">If</a>	This function returns the first specified value if the specified condition is met. Otherwise, it returns the second specified value. The value that is returned can be a value of any of the supported data types.
<a href="#">Not</a>	This function returns the reversed logical value of the specified condition as a <i>Boolean</i> value.
<a href="#">Or</a>	This function returns a <i>Boolean</i> value of <b>FALSE</b> if all the specified conditions are false. If any specified condition is true, the function returns a <i>Boolean</i> value of <b>TRUE</b> .
<a href="#">ValueIn</a>	This function determines whether the specified input matches any value of a specified item in the specified list. It returns a <i>Boolean</i> value of <b>TRUE</b> if the specified input matches the result of running the specified expression for at least one record of the specified list. Otherwise, it returns a <i>Boolean</i> value of <b>FALSE</b> .
<a href="#">ValueInLarge</a>	This function determines whether the specified input of the <i>Int64</i> or <i>Integer</i> type matches any value of a specified item in the specified list. It returns a <i>Boolean</i> value of <b>TRUE</b> if the specified input matches the result of running the specified expression for at least one record of the specified list. Otherwise, it returns a <i>Boolean</i> value of <b>FALSE</b> .

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

## Electronic reporting formula language

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# AND ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `AND` function returns a *Boolean* value of **TRUE** if all the specified conditions are true. Otherwise, it returns a *Boolean* value of **FALSE**.

## Syntax

```
AND (condition 1[, condition 2, ..., condition N])
```

## Arguments

`condition 1` : *Boolean*

A valid conditional expression that must be tested. This argument is required.

`condition N` : *Boolean*

A valid conditional expression that must be tested. These additional arguments are optional.

## Return values

*Boolean*

The resulting *Boolean* value.

## Usage notes

In the arguments of logical functions, you can use data source references, numeric and text values, Boolean values, comparison operators, and other Electronic reporting (ER) functions. However, all the arguments must be evaluated to a *Boolean* value of **TRUE** or **FALSE**.

## Example

`AND (1=1, "a"="a")` returns **TRUE**.

`AND (1=2, "a"="a")` returns **FALSE**.

## Additional resources

[Logical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# CASE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CASE` function evaluates the value of the specified expression against the specified alternative options and returns the result of the first option that equals the value of the specified expression. Otherwise, it returns the optional default result, if a default result is specified as the last argument of the called function that isn't preceded by an option. The value that is returned can be a value of any of the supported data types.

## Syntax

```
CASE (expression, option 1, result 1[, option 2, result 2, ..., option N, result N, default result])
```

## Arguments

`expression`: *Primitive data type* (Boolean, numeric, or text)

A valid expression that returns a value of the primitive data type.

`option 1`: *Primitive data type* (Boolean, numeric, or text)

A valid expression that returns a value of the same primitive data type as the `expression` argument of the called function. This argument is required.

`result 1`: *Any of the supported data types*

The returned result that corresponds to the preceding option. This argument is required.

`option N`: *Primitive data type* (Boolean, numeric, or text)

A valid expression that returns a value of the same primitive data type as the `expression` argument of the called function. This argument is optional.

`result N`: *Any of the supported data types*

The returned result that corresponds to the preceding option. This argument is optional.

`default result`: *Any of the supported data types*

The result that should be returned if there is no match. This argument is optional.

## Return values

*Any of the supported data types*

The resulting value of any of the supported data types.

## Usage notes

An exception is thrown at runtime if there is no match and an optional default result isn't defined.

All results must be specified by using the same data type. An exception is thrown at design time if the data types of the configured results don't match.

If the first result value and the *N*th result value are values of the *Container (record)* or *Record list* data type, the

result has only the fields that exist in both values.

## Example

`CASE( DATETIMEFORMAT( NOW(), "MM"), "10", "WINTER", "11", "WINTER", "12", "WINTER", "" )` returns the string "WINTER" if the current application session date is between October and December. Otherwise, it returns a blank string.

## Additional resources

[Logical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# IF ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `IF` function returns the first specified value if the specified condition is met. Otherwise, it returns the second specified value. The value that is returned can be a value of any of the supported data types.

## Syntax

```
IF (condition, first value, second value) as any of the supported data types
```

## Arguments

`condition` : *Boolean*

A valid conditional expression that must be tested.

`first value` : *Any of the supported data types*

The result that is returned if the condition is met.

`second value` : *Any of the supported data types*

The result that is returned if the condition isn't met.

## Return values

*Any of the supported data types*

The resulting value of any of the supported data types.

## Usage notes

The `first value` and `second value` arguments must be specified by using the same data type. An exception is thrown at design time if the data types of the configured values don't match.

If the first value and the second value are values of the *Container (record)* or *Record list* data type, the result has only the fields that exist in both values.

## Example

```
IF (1=2, "condition is met", "condition is not met") returns the string "condition is not met".
```

## Additional resources

[Logical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NOT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NOT` function returns the reversed logical value of the specified condition as a *Boolean* value.

## Syntax

```
NOT (condition)
```

## Arguments

`condition`: *Boolean*

A valid conditional expression that must be reversed.

## Return values

*Boolean*

The resulting *Boolean* value.

## Example

`NOT (TRUE)` returns `FALSE`.

## Additional resources

[Logical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# OR ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `OR` function returns a *Boolean* value of **FALSE** if all the specified conditions are false. If any specified condition is true, the function returns a *Boolean* value of **TRUE**.

## Syntax

```
OR (condition 1[, condition 2, ..., condition N])
```

## Arguments

`condition 1` : *Boolean*

A valid conditional expression that must be tested. This argument is required.

`condition N` : *Boolean*

A valid conditional expression that must be tested. These additional arguments are optional.

## Return values

*Boolean*

The resulting *Boolean* value.

## Example

`OR (1=2, "a"="a")` returns **TRUE**.

## Additional resources

[Logical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# VALUEIN ER function

2/18/2021 • 3 minutes to read • [Edit Online](#)

The `VALUEIN` function determines whether the specified input matches any value of a specified item in the specified list. It returns a *Boolean* value of **TRUE** if the specified input matches the result of running the specified expression for at least one record of the specified list. Otherwise, it returns a *Boolean* value of **FALSE**.

## Syntax

```
VALUEIN (input, list, list item expression)
```

## Arguments

`input`: *Field*

The valid path of an item of a data source of the *Record list* type. The value of this item will be matched.

`list`: *Record list*

The valid path of a data source of the *Record list* data type.

`list item expression`: *Boolean*

A valid conditional expression that either points to or contains a single field of the specified list that should be used for the matching.

## Return values

*Boolean*

The resulting *Boolean* value.

## Usage notes

In general, the `VALUEIN` function is translated to a set of **OR** conditions. If the list of **OR** conditions is large and the maximum total length of an SQL statement might be exceeded, consider using the `VALUEINLARGE` function.

```
(input = list.item1.value) OR (input = list.item2.value) OR ...
```

In some cases, it can be translated to a database SQL statement by using the `EXISTS JOIN` operator.

## Example 1

In your model mapping, you define the **List** data source of the *Calculated field* type. This data source contains the expression `SPLIT ("a,b,c", ",")`.

When a data source is called, if it has been configured as the `VALUEIN ("B", List, List.Value)` expression, it returns **TRUE**. In this case, the `VALUEIN` function is translated to the following set of conditions:

```
((("B" = "a") or ("B" = "b") or ("B" = "c")), where ("B" = "b") equals TRUE.
```

When a data source is called, if it has been configured as the `VALUEIN ("B", List, LEFT(List.Value, 0))`

expression, it returns **FALSE**. In this case, the `VALUEIN` function is translated to the following condition: `("B" = "")`, which doesn't equal **TRUE**.

The upper limit for the number of characters in the text of such a condition is 32,768 characters. Therefore, you should not create data sources that might exceed this limit at runtime. If the limit is exceeded, the application stops running, and an exception is thrown. For example, this situation can occur if the data source is configured as `WHERE (List1, VALUEIN (List1.ID, List2, List2.ID))`, and the `List1` and `List2` lists contain a large volume of records.

In some cases, the `VALUEIN` function is translated to a database statement by using the `EXISTS JOIN` operator. This behavior occurs when the `FILTER` function is used and the following conditions are met:

- The **ASK FOR QUERY** option is turned off for the data source of the `VALUEIN` function that refers to the list of records. No additional conditions will be applied to this data source at runtime.
- No nested expressions are configured for the data source of the `VALUEIN` function that refers to the list of records.
- A list item of the `VALUEIN` function refers to a field of the specified data source, not to an expression or method of that data source.

Consider using this option instead of the `WHERE` function that is described earlier in this example.

## Example 2

You define the following data sources in your model mapping:

- The `In` data source of the *Table records* type. This data source refers to the `Intrastat` table.
- The `Port` data source of the *Table records* type. This data source refers to the `IntrastatPort` table.

When a data source is called that has been configured as the `FILTER (In, VALUEIN(In.Port, Port, Port.PortId))` expression, the following SQL statement is generated to return filtered records of the `Intrastat` table.

```
select ... from Intrastat
exists join TableId from IntrastatPort
where IntrastatPort.PortId = Intrastat.Port
```

For `dataAreaId` fields, the final SQL statement is generated by the using `IN` operator.

## Example 3

You define the following data sources in your model mapping:

- The `Le` data source of the *Calculated field* type. This data source contains the expression `SPLIT ("DEMF,GBSI,USMF", ",")`.
- The `In` data source of the *Table records* type. This data source refers to the `Intrastat` table, and the **Cross-company** option is turned on for it.

When a data source is called that has been configured as the `FILTER (In, VALUEIN (In.dataAreaId, Le, Le.Value))` expression, the final SQL statement contains the following condition.

```
Intrastat.dataAreaId IN ('DEMF', 'GBSI', 'USMF')
```

## Additional resources

Logical functions

VALUEINLARGE functions

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# List of ER functions in the mathematical category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) mathematical functions can be used to do many common mathematical calculations. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">Abs</a>	This function returns the absolute value (modulus) of the specified number as a <i>Real</i> value. In other words, it returns the number without its sign.
<a href="#">Power</a>	This function returns a <i>Real</i> value that represents the result of raising the specified positive number to the specified power.
<a href="#">Round</a>	This function returns the specified number as a <i>Real</i> value after it has been rounded to the specified number of decimal places.
<a href="#">RoundDown</a>	This function returns the specified number as a <i>Real</i> value after it has been rounded down to the specified number of decimal places.
<a href="#">RoundUp</a>	This function returns the specified number as a <i>Real</i> value after it has been rounded up to the specified number of decimal places.

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

[Electronic reporting formula language](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ABS ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ABS` function returns the absolute value (modulus) of the specified number as a *Real* value. In other words, it returns the number without its sign.

## Syntax

```
ABS (number)
```

## Arguments

`number` : *Real*

A numeric value that you want the modulus of.

## Return values

*Real*

The resulting numeric value.

## Example

`ABS (-1)` returns 1.

## Additional resources

[Mathematical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# POWER ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `POWER` function returns a *Real* value that represents the result of raising the specified positive number to the specified power.

## Syntax

```
POWER (number, power)
```

## Arguments

`number` : *Real* or *Integer*

A numeric value that must be raised to the specified power.

`power` : *Real* or *Integer*

A numeric value that represents the specific power.

## Return values

*Real*

The resulting numeric value.

## Example 1

`POWER (10, 2)` returns 100.

## Example 2

`POWER (4, 0.5)` returns 2.

## Additional resources

[Mathematical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ROUND ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ROUND` function returns the specified number as a *Real* value after it has been rounded to the specified number of decimal places.

## Syntax

```
ROUND (number, decimals)
```

## Arguments

`number` : *Real*

A numeric value that must be rounded.

`decimals` : *Integer*

A numeric value that represents the number of decimal places.

## Return values

*Real*

The resulting numeric value.

## Usage notes

If the value of the `decimals` argument is more than 0 (zero), the specified number is rounded to that many decimal places.

If the value of the `decimals` argument is 0 (zero), the specified number is rounded to the nearest even integer.

If the value of the `decimals` argument is less than 0 (zero), the specified number is rounded to the left of the decimal point.

## Example 1

`ROUND (1200.767, 2)` rounds to two decimal places and returns **1200.77**.

## Example 2

`ROUND (1200.767, -3)` rounds to the nearest multiple of 1,000 and returns **1000**.

## Example 3

`ROUND (1200.5, 0)` rounds to the nearest even integer and returns **1200**, while `ROUND (1201.5, 0)` does the same and returns **1202**.

## Additional resources

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ROUNDDOWN ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ROUNDDOWN` function returns the specified number as a *Real* value after it has been rounded down to the specified number of decimal places.

## Syntax

```
ROUNDDOWN (number, decimals)
```

## Arguments

`number` : *Real*

A numeric value that must be rounded down.

`decimals` : *Integer*

A numeric value that represents the number of decimal places.

## Return values

*Real*

The resulting numeric value.

## Usage notes

This function behaves like `ROUND`, but it always rounds the specified number down (toward zero).

## Example 1

`ROUNDDOWN (1200.767, 2)` rounds down to two decimal places and returns **1200.76**.

## Example 2

`ROUNDDOWN (1700.767, -3)` rounds down to the nearest multiple of 1,000 and returns **1000**.

## Additional resources

[Mathematical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ROUNDUP ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ROUNDUP` function returns the specified number as a *Real* value after it has been rounded up to the specified number of decimal places.

## Syntax

```
ROUNDUP (number, decimals)
```

## Arguments

`number` : *Real*

A numeric value that must be rounded up.

`decimals` : *Integer*

A numeric value that represents the number of decimal places.

## Return values

*Real*

The resulting numeric value.

## Usage notes

This function behaves like `ROUND`, but it always rounds the specified number up (away from zero).

## Example 1

`ROUNDUP (1200.763, 2)` rounds up to two decimal places and returns **1200.77**.

## Example 2

`ROUNDUP (1200.767, -3)` rounds up to the nearest multiple of 1,000 and returns **2000**.

## Additional resources

[Mathematical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# List of ER functions in the record category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) record functions can be used to extract information from, and perform operations on, data sources of the *Container (record)* data type. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">NullContainer</a>	This function returns a null <i>Container (record)</i> value that has the same structure as the specified record list or record.
<a href="#">EmptyRecord</a>	This function returns a null <i>Container (record)</i> value that has the same structure as the specified record list or record.

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

[Electronic reporting formula language](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# EMPTYRECORD ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `EMPTYRECORD` function returns a null *Container (record)* value that has the same structure as the specified record list or record.

## Syntax

```
EMPTYRECORD (list)
```

## Arguments

`list` : *Record list* or *Container (record)*

The valid path of a data source of either the *Record list* or *Container (record)* type.

## Return values

*Container (record)*

The resulting record value.

## Usage notes

### NOTE

A null record is a record where all fields have an empty value. An empty value is 0 (zero) for numbers, an empty string for strings, and so on.

## Example

`EMPTYRECORD (SPLIT ("abc", 1))` returns a new empty record that has the same structure as the list that is returned by the `SPLIT` function. For more information, see [SPLIT](#).

## Additional resources

[Record functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NULLCONTAINER ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NULLCONTAINER` function returns a null *Container (record)* value that has the same structure as the specified record list or record.

## Syntax

```
NULLCONTAINER (list)
```

## Arguments

`list` : *Record list* or *Container (record)*

The valid path of a data source of either the *Record list* or *Container (record)* type.

## Return values

*Container (record)*

The resulting record value.

## Usage notes

### NOTE

This function is obsolete. Use the `EMPTYRECORD` function instead. For more information, see [EMPTYRECORD](#).

## Example

`NULLCONTAINER (SPLIT ("abc", 1))` returns a new empty record that has the same structure as the list that is returned by the `SPLIT` function. For more information, see [SPLIT](#).

## Additional resources

[Record functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# List of ER functions of the text category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) text functions can be used to perform operations on data sources of the *String* data type. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">Char</a>	This function returns a <i>String</i> value that presents a single character that is referenced by the specified Unicode number.
<a href="#">Concatenate</a>	This function returns all the specified text strings as a <i>String</i> value after they have been joined into one string.
<a href="#">Format</a>	This function returns the specified string a <i>String</i> value after it has been formatted by substituting any occurrences of %N with the Mth argument.
<a href="#">GetEnumValueByName</a>	This function searches for a specific <i>Enum</i> value in the specified enumeration data source by using the enumeration name that is specified as a <i>String</i> value. If the <i>Enum</i> value is found, the function returns it.
<a href="#">GuidValue</a>	This function converts the specified input of the <i>String</i> type to a data item of the <i>GUID</i> type.
<a href="#">JsonValue</a>	This function parses data in JavaScript Object Notation (JSON) format that is accessed at the specified path, and it extracts a scalar value that is based on the specified ID. It then returns the extracted scalar value as a <i>String</i> value.
<a href="#">Left</a>	This function returns a <i>String</i> value that presents the specified number of characters from the start of the specified string.
<a href="#">Len</a>	This function returns an <i>Integer</i> value that presents the number of characters in the specified string.
<a href="#">Lower</a>	This function returns the specified text string as a <i>String</i> value after it has been converted to lowercase letters.
<a href="#">Mid</a>	This function returns a <i>String</i> value that presents the specified number of characters from the specified string, starting at the specified position.
<a href="#">NumberFormat</a>	This function returns a <i>String</i> value that presents the specified number in the specified format and in an optionally specified culture.

FUNCTION	DESCRIPTION
<a href="#">NumeralsToText</a>	This function returns the specified number as a <i>String</i> value after it has been spelled out (that is, converted to text strings) in the specified language.
<a href="#">PadLeft</a>	This function returns a <i>String</i> value of the specified length, where the start of the specified string is padded with one or more instances of the specified characters.
<a href="#">QrCode</a>	This function returns a <i>Container</i> value that presents the Quick Response code (QR code) image for the specified string in binary format.
<a href="#">Replace</a>	This function returns the specified text string as a <i>String</i> value after all or part of it has been replaced with another string.
<a href="#">Right</a>	This function returns a <i>String</i> value that presents the specified number of characters from the end of the specified string.
<a href="#">Text</a>	This function returns the specified number as a <i>String</i> value after it has been converted to a text string that is formatted according to the server locale settings of the current application instance.
<a href="#">Translate</a>	This function returns a <i>String</i> value that contains the result of the replacement the specified text in characters for another provided set of characters.
<a href="#">Trim</a>	This function returns the specified text string as a <i>String</i> value after leading and trailing spaces have been truncated, and after multiple spaces between words have been removed.
<a href="#">Upper</a>	This function returns the specified text string as a <i>String</i> value after it has been converted to uppercase letters.

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

[Electronic reporting formula language](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# CHAR ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CHAR` function returns a *String* value that presents a single character that is referenced by the specified Unicode number.

## Syntax

```
CHAR (number)
```

## Arguments

`number` : *Integer*

A number that corresponds to an expected single character.

## Return values

*String*

The resulting text value.

## Usage notes

The string that this function returns depends on the encoding that is selected in the parent **FILE** format element. For a list of the supported encodings, see [Encoding class](#).

## Example

`CHAR (255)` returns "ÿ".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# CONCATENATE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CONCATENATE` function returns all the specified text strings as a *String* value after they have been joined into one string.

## Syntax

```
CONCATENATE (text 1[, text 2, ..., text N])
```

## Arguments

`text 1` : *String*

A reference to a data source of the *String* data type. This argument is required.

`text N` : *String*

A reference to a data source of the *String* data type. These additional arguments are optional.

## Return values

*String*

The resulting text value.

## Example

```
CONCATENATE ("abc", "def") returns "abcdef".
```

## Usage notes

The expression `"abc" & "def"` also returns **"abcdef"**.

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# FORMAT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `FORMAT` function returns the specified string as a *String* value after it has been formatted by substituting any occurrences of %N with the *N*th argument.

## Syntax

```
FORMAT (string, argument 1[, argument 2, ..., argument N])
```

## Arguments

`string` : *String*

A reference to a data source of the *String* type that must be formatted. This argument is required.

`argument 1` : *String*

The first argument, which is used to replace occurrences of %1. This argument is required.

`argument N` : *String*

The *N*th argument, which is used to replace occurrences of %2, %3, and so on. These additional arguments are optional.

## Return values

*String*

The resulting text value.

## Usage notes

If an argument isn't provided for a parameter, the parameter is returned as "%N" in the string. For values of the *Real* type, the default string conversion is limited to two decimal places.

## Example

In the following illustration, the **PaymentModel** data source returns a list of customer records by using the **Customer** component. It returns the processing date value by using the **ProcessingDate** field.

▲ model: Data model PaymentModel

▲ Customer: Record list

Name: String

ProcessingDate: DateTime

In the Electronic reporting (ER) format that is designed to generate an electronic file for selected customers, **PaymentModel** is selected as a data source, and it controls the process flow. If a selected customer is stopped for the date when the report is processed, an exception is thrown to notify the user. The formula that is designed

for this type of processing control can use the following resources:

- Label SYS70894, which has the following text:
  - For the EN-US language: "Nothing to print"
  - For the DE language: "Nichts zu drucken"
- Label SYS18389, which has the following text:
  - For the EN-US language: "Customer %1 is stopped for %2."
  - For the DE language: "Debitor '%1' wird für %2 gesperrt."

Here is the expression that can be designed.

```
FORMAT (CONCATENATE (@"SYS70894", ". ", @"SYS18389"), model.Customer.Name, DATETIMEFORMAT  
(model.ProcessingDate, "d"))
```

If a report is processed for the **Litware Retail** customer on December 17, 2015, in the **EN-US** culture and the **EN-US** language, this formula returns the following text, which can be presented to the user as an exception message:

*Nothing to print. Customer Litware Retail is stopped for 12/17/2015.*

If the same report is processed for the **Litware Retail** customer on December 17, 2015, in the **DE** culture and the **DE** language, the formula returns the following text, which uses a different date format:

*Nichts zu drucken. Debitor 'Litware Retail' wird für 17.12.2015 gesperrt.*

#### NOTE

The following syntax is applied in ER formulas for labels:

- For labels from resources in the Microsoft Dynamics 365 Finance app: @X, where X is the label ID in the Application Object Tree (AOT)
- For labels that reside in ER configurations: @"GER\_LABEL:X", where X is the label ID in the ER configuration

## Additional resources

### Text functions

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# GETENUMVALUEBYNAME ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `GETENUMVALUEBYNAME` function searches for a specific *Enum* value in the specified enumeration data source by using the enumeration name that is specified as a *String* value. If the *Enum* value is found, the function returns it. Otherwise, the function returns the **null** enumeration value.

## Syntax

```
GETENUMVALUEBYNAME (enumeration data source path, enumeration value text)
```

## Arguments

enumeration data source path : *Enumeration*

The valid path of a data source of one of the following enumeration types:

- Electronic reporting (ER) model enumeration
- ER format enumeration
- Microsoft Dynamics 365 Finance enumeration

enumeration value text : *String*

A string value that represents the name of a single enumeration value.

## Return values

Nullable *Enum*

The resulting enumeration value.

## Usage notes

No exception is thrown if an *Enum* value isn't found by using the name of the enumeration value that is specified as a *String* value.

## Example 1

In the following illustration, the **ReportDirection** enumeration is introduced in a data model. Notice that labels are defined for the enumeration values.

Report direction(ReportDirection)

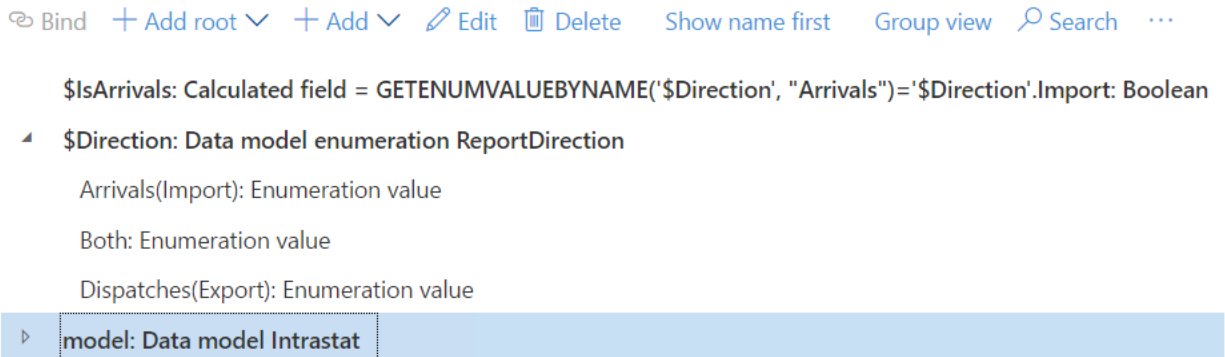
### Enumeration values

+ New  Delete

Name	Label (*Recommended to use labels)	Description
Both	Both	
Export	Dispatches	
Import	Arrivals	

The following illustration shows these details:

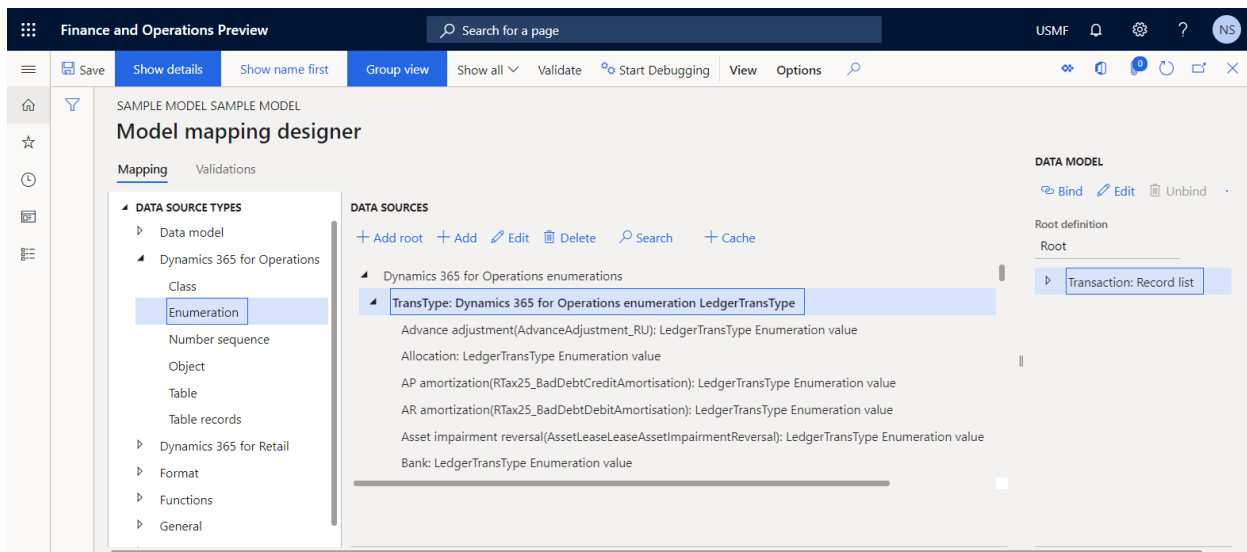
- The **\$Direction** data source is configured in an ER report. This data source is configured based on the **ReportDirection** model enumeration.
- The `$IsArrivals` expression is designed to use the model enumeration–based **\$Direction** data source as a parameter of this function.
- The value of this comparison expression is **TRUE**.



## Example 2

The `GETENUMVALUEBYNAME` and `LISTOFFIELDS` functions let you fetch values and labels of supported enumerations as text values. (The supported enumerations are application enumerations, data model enumerations, and format enumerations.)

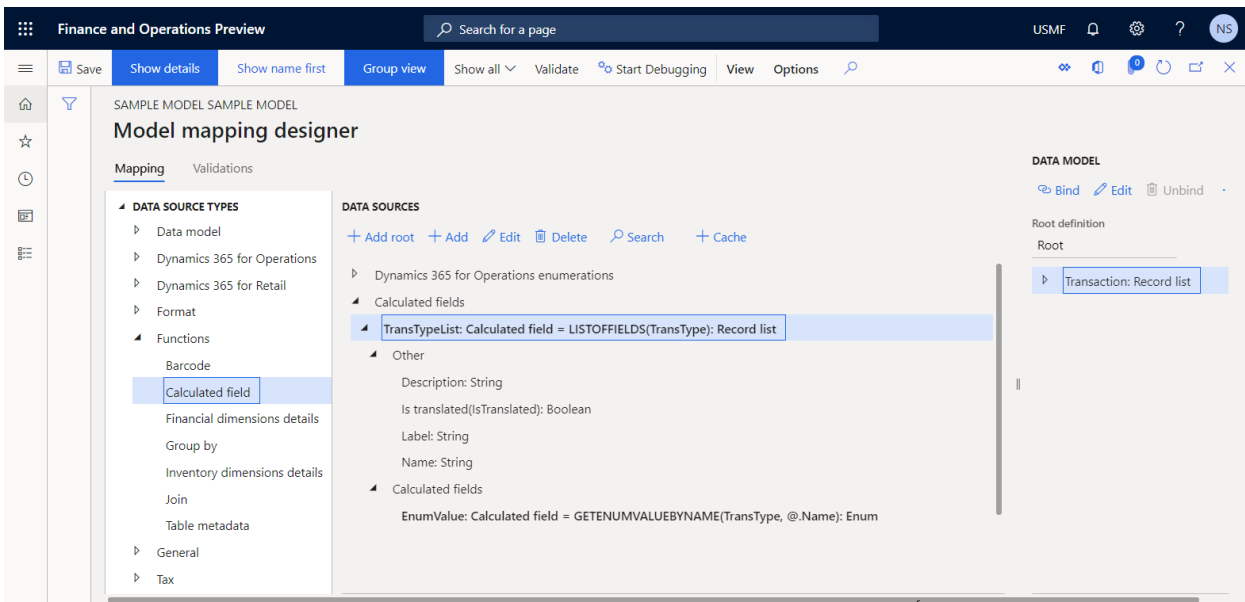
In the following illustration, the **TransType** data source is introduced in a model mapping. This data source refers to the **LedgerTransType** application enumeration.



The following illustration shows the **TransTypeList** data source that is configured in a model mapping. This data source is configured based on the **TransType** application enumeration. The `LISTOFFIELDS` function is used to return all enumeration values as a list of records that contain fields. In this way, the details of every enumeration value are exposed.

### NOTE

The `EnumValue` field is configured for the **TransTypeList** data source by using the `GETENUMVALUEBYNAME(TransType, TransTypeList.Name)` expression. This field returns an enumeration value for every record in this list.

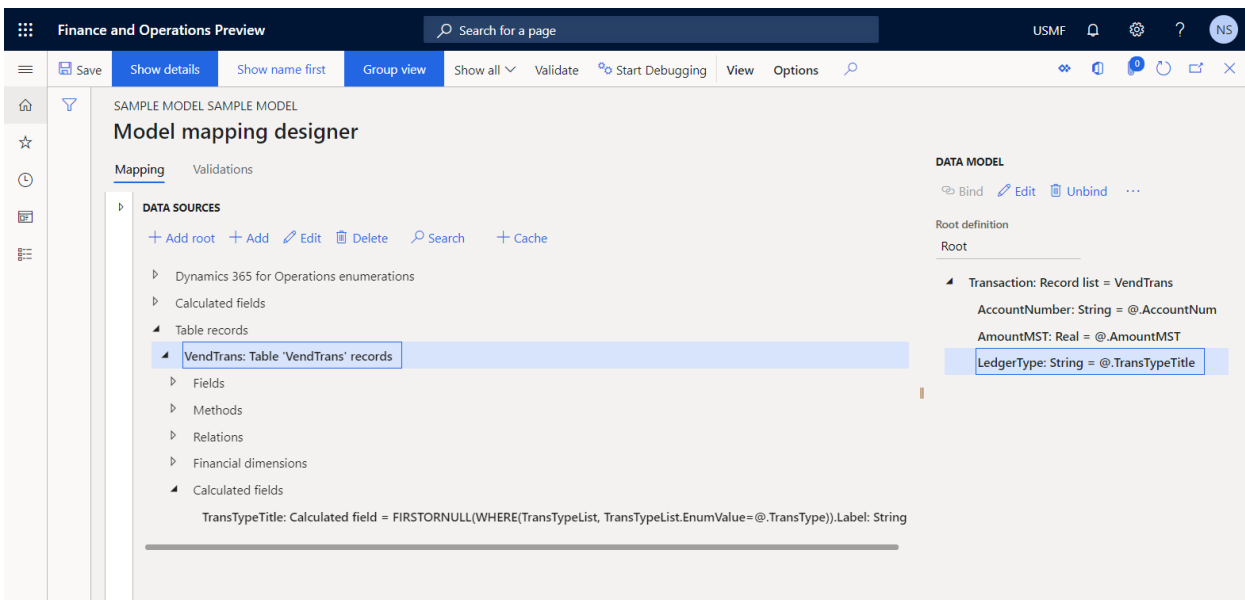


The following illustration shows the **VendTrans** data source that is configured in a model mapping. This data source returns vendor transaction records from the **VendTrans** application table. The ledger type of every transaction is defined by the value of the **TransType** field.

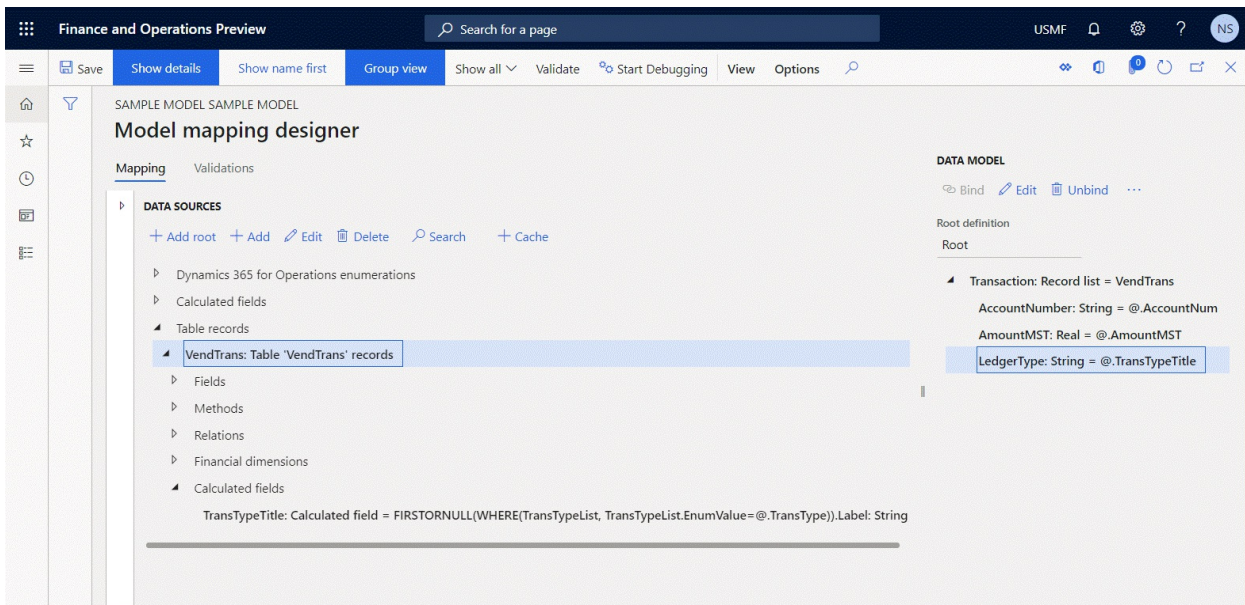
#### NOTE

The **TransTypeTitle** field is configured for the **VendTrans** data source by using the `FIRSTORNULL(WHERE(TransTypeList, TransTypeList.EnumValue = @.TransType)).Label` expression. This field returns the label of an enumeration value of the current transaction as text, if this enumeration value is available. Otherwise, it returns a blank string value.

The **TransTypeTitle** field is bound to the **LedgerType** field of a data model that enables this information to be used in every ER format that uses the data model as a source of data.



The following illustration shows how you can use the [data source debugger](#) to test the configured model mapping.



The **LedgerType** field of a data model exposes labels of transaction types as expected.

If you plan to use this approach for a large amount of transactional data, you must consider execution performance. For more information, see [Trace the execution of ER formats to troubleshoot performance issues](#).

## Additional resources

[Text functions](#)

[Trace the execution of ER formats to troubleshoot performance issues](#)

[LISTOFFIELDS ER function](#)

[FIRSTORNULLE ER function](#)

[WHERE ER function](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# GUIDVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `GUIDVALUE` function converts the specified input of the *String* type to a data item of the *GUID* type.

## Syntax

```
GUIDVALUE (input)
```

## Arguments

`input`: *String*

The valid path of a data source of the *String* type.

## Return values

*GUID*

The resulting globally unique identifier (GUID) value.

## Usage notes

To do a conversion in the opposite direction (that is, to convert specified input of the *GUID* data type to a data item of the *String* data type), you can use the [TEXT](#) function.

## Example

You define the following data sources in your model mapping:

- A `myID` data source of the *Calculated field* type that contains the expression `GUIDVALUE ("AF5CCDAC-F728-4609-8C8B-A4B30B0C0AA0")`
- A `Users` data source of the *Table records* type that refers to the `UserInfo` table

You can then use an expression such as `FILTER (Users, Users.objectId = myID)` to filter the `UserInfo` table by the `objectId` field of the *GUID* data type.

## Additional resources

### [Text functions](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# JSONVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `JSONVALUE` function parses data in JavaScript Object Notation (JSON) format that is accessed at the specified path, and it extracts a scalar value that has the specified ID. It then returns the extracted scalar value as a *String* value.

## Syntax

```
JSONVALUE (input, path)
```

## Arguments

`input`: *String*

The valid path of a data source of the *String* type that contains JSON data.

`path`: *String*

The identifier of a scalar value of JSON data.

## Return values

*String*

The resulting text value.

## Example

The `JsonField` data source contains the following data in JSON format: `{"BuildNumber":"7.3.1234.1", "KeyThumbprint":"7366E"}`. In this case, the expression `JSONVALUE (JsonField, "BuildNumber")` returns the following value of the *String* data type: `"7.3.1234.1"`.

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# LEFT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `LEFT` function returns a *String* value that presents the specified number of characters from the start of the specified string.

## Syntax

```
LEFT (text, number)
```

## Arguments

`text` : *String*

A *String* value that represents the original text.

`number` : *Integer*

The number of characters that must be returned from the start of the original text.

## Return values

*String*

The resulting text value.

## Example

`LEFT ("Sample", 3)` returns "Sam".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# LEN ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `LEN` function returns the number of characters in the specified string as an *Integer* value.

## Syntax

```
LEN (text)
```

## Arguments

`text` : *String*

A *String* value that specifies the text.

## Return values

*Integer*

The resulting numeric value.

## Example

`LEN ("Sample")` returns 6.

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# LOWER ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `LOWER` function returns the specified text string as a *String* value after it has been converted to lowercase letters.

## Syntax

```
LOWER (text)
```

## Arguments

`text` : *String*

A *String* value that specifies the text.

## Return values

*String*

The resulting text value.

## Example

```
LOWER ("Sample") returns "sample".
```

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# MID ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `MID` function returns a *String* value that presents the specified number of characters from the specified string, starting at the specified position.

## Syntax

```
MID (text, starting position, number of characters)
```

## Arguments

`text` : *String*

A *String* value that specifies the text to return characters from.

`starting position` : *Integer*

An *Integer* value that specifies the position of the first character that must be returned from the specified text.

`number of characters` : *Integer*

An *Integer* value that specifies the number of characters that must be returned, starting at the specified starting position.

## Return values

*String*

The resulting text value.

## Usage notes

If the value of the `starting position` argument is less than 0 (zero), the characters that are returned are counted from the first position in the specified string.

If the value of the `starting position` argument exceeds length of the specified string, an empty string is returned.

## Example

`MID ("Sample", 2, 3)` returns **"amp"**.

## Additional resources

[Text functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NUMBERFORMAT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NUMBERFORMAT` function returns a *String* value that presents the specified number in the specified format and in an optionally specified *culture*. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
NUMBERFORMAT (number, format)
```

## Syntax 2

```
NUMBERFORMAT (number, format, culture)
```

## Arguments

`number` : *Integer* or *Real*

A numeric value that specifies the number that must be formatted.

`format` : *String*

A *String* value that represents the format.

`culture` : *String*

A *String* value that represents the culture to use for formatting.

## Return values

*String*

The resulting text value.

## Usage notes

If the culture isn't defined as an argument of the called function, the context that this function is run in determines the culture that is used to format numbers.

## Example 1

For the **EN-US** culture, `NUMBERFORMAT (0.45, "p")` returns "45.00 %", and `NUMBERFORMAT (10.45, "#")` returns "10".

## Example 2

`NUMBERFORMAT (10/3, "F2", "de")` returns 3,33, whereas `NUMBERFORMAT (10/3, "F2", "en-us")` returns 3.33.

## Additional resources

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NUMERALSTOTEXT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NUMERALSTOTEXT` function returns the specified number as a *String* value after it has been spelled out (that is, converted to text strings) in the specified language.

## Syntax

```
NUMERALSTOTEXT (number, language, currency, print currency name flag, decimal points)
```

## Arguments

`number` : *Integer or Real*

A numeric value that specifies the number that must be spelled out.

`language` : *String*

A *String* value that represents the language code.

`currency` : *String*

A *String* value that represents the currency code.

`print currency name flag` : *Boolean*

A *Boolean* value that indicates whether a currency name must be added to the spelled-out text.

`decimal points` : *Integer*

An *Integer* value that indicates the number of decimal places that the spelled-out text should have.

## Return values

*String*

The resulting text value.

## Usage notes

The language code is optional. If it's defined as an empty string, the language code for the running context is used. The default language code is **EN-US**. The language code for the running context is defined in a **Folder** or **File** element of the Electronic reporting (ER) format that is running.

The currency code is optional. If it's defined as an empty string, the company currency for the running context is used.

### NOTE

The `print currency name flag` and `decimal points` arguments are analyzed only for the following language codes: **CS**, **ET**, **HU**, **LT**, **LV**, **PL**, and **RU**. Additionally, the `print currency name flag` argument is analyzed only for companies where the country's or region's context supports declension of currency names.

## Example 1

`NUMERALSTOTEXT (1234.56, "EN-US", "", false, 2)` returns "One Thousand Two Hundred Thirty Four and 56".

## Example 2

`NUMERALSTOTEXT (120, "PL", "", false, 0)` returns "Sto dwadzieścia".

## Example 3

`NUMERALSTOTEXT (120.21, "RU", "EUR", true, 2)` returns "Сто двадцать евро 21 евроцент".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# PADLEFT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `PADLEFT` function returns a *String* value of the specified length, where the start of the specified string is padded with the specified characters.

## Syntax

```
PADLEFT (text, length, padding chars)
```

## Arguments

`text` : *String*

A *String* value that represents the original text.

`length` : *Integer*

An *Integer* value that represents the final number of characters in the padded string.

`padding chars` : *String*

The characters to use for padding.

## Return values

*String*

The resulting text value.

## Example

`PADLEFT ("1234", 10, " ")` returns the text string " 1234".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# QR CODE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `QR CODE` function returns a *Container* value that presents the Quick Response code (QR code) image for the specified string in binary format.

## Syntax

```
QR CODE (text)
```

## Arguments

`text` : *String*

A *String* value that represents the original text.

## Return values

*Container*

The resulting binary stream.

## Example

You can configure an Electronic reporting (ER) format to generate an outbound document in Microsoft Office format (Excel workbooks or Word documents) by using a predefined template. This template may contain a **Picture** object (Excel workbook) or a **Picture Content Control** (Word document) as a placeholder for a QR code image. You need to add to the configured ER format a **Cell** element that will be used to fill this placeholder in. To specify what information will be stored in a QR code, you need to define a binding for this **Cell** element. For example, you can configure such binding as containing the following expression:

```
QR CODE (model.ListOfShelfLabels.LabelText)`
```

When you run the configured ER format, the text value of the **LabelText** field of the **model.ListOfShelfLabels** data source will be used to generate a QR code image. This image will replace a QR code image placeholder in the document template using to generate an outbound document. When this image of the generated document is scanned, it returns the text that was taken from the **LabelText** field of the **model.ListOfShelfLabels** data source. For more information, see [Embed images and shapes in documents that you generate by using ER](#).

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# REPLACE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `REPLACE` function returns the specified text string as a *String* value after all or part of it has been replaced with another string.

## Syntax

```
REPLACE (text, pattern, replacement, regular expression flag)
```

## Arguments

`text` : *String*

The valid path of a data source of the *String* type.

`pattern` : *String*

If the `regular expression flag` argument is **FALSE**, this argument contains the text that must be replaced.

If the `regular expression flag` argument is **TRUE**, this argument contains a regular expression that defines both a search pattern and the replacement text.

`replacement` : *String*

If the `regular expression flag` argument is **FALSE**, this argument contains the text to use as a replacement.

If the `regular expression flag` argument is **TRUE**, this argument isn't used.

`regular expression flag` : *Boolean*

A *Boolean* value that indicates whether a regular expression is used to do the replacement.

## Return values

*String*

The resulting text value.

## Usage notes

If the `regular expression flag` argument is **TRUE**, this function returns the specified string after it has been changed by applying the regular expression that is specified by the `pattern` argument. The regular expression is used to find the characters that must be replaced.

If the `regular expression flag` argument is **FALSE**, this function returns the specified string after the set of characters that are defined in the `pattern` argument have been replaced by characters of the `replacement` argument.

## Example 1

```
REPLACE ("+1 923 456 4971", "[^0-9]", "", true)
```

 applies a regular expression that removes all non-numeric

symbols, and it returns "19234564971".

## Example 2

`REPLACE ("abcdef", "cd", "GH", false)` replaces the pattern "cd" with the string "GH" and returns "abGHef".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# RIGHT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `RIGHT` function returns a *String* value that presents the specified number of characters from the end of the specified string.

## Syntax

```
RIGHT (text, number)
```

## Arguments

`text` : *String*

A *String* value that represents the original text.

`number` : *Integer*

The number of characters that must be returned from the end of the original text.

## Return values

*String*

The resulting text value.

## Example

```
RIGHT ("Sample", 3) returns "ple".
```

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# TEXT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `TEXT` function returns the specified number as a *String* value after it has been converted to a text string that is formatted according to the server locale settings of the current application instance.

## Syntax

```
TEXT (number)
```

## Arguments

`number`: *Integer* or *Real*

A number that must be converted to a text string.

## Return values

*String*

The resulting text value.

## Usage notes

For values of the *Real* type, the string conversion is limited to two decimal places.

## Example

If the server locale of the Microsoft Dynamics 365 Finance instance is defined as EN-US, `TEXT (NOW ())` returns the current Finance session date, December 17, 2015, as the text string "12/17/2015 07:59:23 AM".

`TEXT (1/3)` returns "0.33".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# TRANSLATE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `TRANSLATE` function returns a *String* value that contains the result of the character replacement of specified text in characters of another provided set.

## Syntax

```
TRANSLATE (text , pattern, replacement)
```

## Arguments

`text` : *String*

The valid path of a data source of the *String* type.

`pattern` : *String*

The text that must be replaced.

`replacement` : *String*

The text to use as a replacement.

## Return values

*String*

The resulting text value.

## Usage notes

The `TRANSLATE` function replaces one character at a time. The function replaces the first character of the `text` argument with the first character of the `pattern` argument and then the second character and follows the same flow until finished. When a character from the `text` and `pattern` arguments match, it is replaced by a character from the `replacement` argument that is located in the same position as the character from the `pattern` argument. If a character appears multiple times in the `pattern` argument, the `replacement` argument mapping that corresponds to the first occurrence of this character is used.

## Example 1

`TRANSLATE ("abcdef", "cd", "GH")` replaces the "c" character of the specified "abcdef" text with the "G" character of the `replacement` text due to the following:

- The "c" character is presented in the `pattern` text in the first position.
- The first position of the `replacement` text contains the "G" character.

## Example 2

`TRANSLATE ("abcdef", "ccd", "GH")` returns "abGdef".

## Example 3

`TRANSLATE ("abccba", "abc", "123")` returns `"123321"`.

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# TRIM ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `TRIM` function returns the specified text string as a *String* value after leading and trailing spaces have been truncated, and after multiple spaces between words have been removed.

## Syntax

```
TRIM (text )
```

## Arguments

`text` : *String*

The valid path of a data source of the *String* type.

## Return values

*String*

The resulting text value.

## Example

`TRIM (" Sample text ")` returns "Sample text".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# UPPER ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `UPPER` function returns the specified text string as a *String* value after it has been converted to uppercase letters.

## Syntax

```
UPPER (text )
```

## Arguments

`text` : *String*

The valid path of a data source of the *String* type.

## Return values

*String*

The resulting text value.

## Example

`UPPER ("Sample")` returns "SAMPLE".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# List of ER functions in the data collection category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) data collection functions are used to do counting and summing in an ER format that is being run, based on data of the output that has already been generated in **Text** or **Xml** format. This approach is used to help improve performance of an ER format that is run, to enter values of running totals in generated documents, and for other purposes. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">CollectedList</a>	This function returns a <i>Record list</i> value that contains the list of values that were returned by the <b>Collected data key value</b> property of format elements and collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified conditions. Each condition consists of a key range and a key value.
<a href="#">CountIF</a>	This function returns an <i>Integer</i> value that represents the number of format elements that was collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified condition. The condition consists of a key range and a key value.
<a href="#">CountIFs</a>	This function returns an <i>Integer</i> value that represents the number of format elements that was collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified conditions. Each condition consists of a key range and a key value.
<a href="#">FormatElementName</a>	This function returns a <i>String</i> value that represents the name of the current ER format's element.
<a href="#">SumIF</a>	This function returns a <i>Real</i> value that represents the sum of values that were returned by bindings of format elements and collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified condition. The condition consists of a key range and a key value.
<a href="#">SumIFs</a>	This function returns a <i>Real</i> value that represents the sum of values that were returned by bindings of format elements and collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified conditions. Each condition consists of a key range and a key value.

## Additional resources

[Electronic Reporting overview](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# COLLECTEDLIST ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `COLLECTEDLIST` function a *Record list* value that contains the list of values that were returned by the **Collected data key value** property of format elements and collected when the format elements were used to generate outbound documents during the format run, and that satisfies the specified conditions. Each condition consists of a key range and a key value.

## Syntax

```
COLLECTEDLIST (condition 1 range, condition 1 value[, condition 2 range, condition 2 value, ..., condition N range, condition N value])
```

## Arguments

`condition 1 range`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of an Electronic reporting (ER) format component. This argument is mandatory.

`condition 1 value`: *String*

A value that is returned by the expression that has been configured in the **Collected data key value** property of an ER format component. This argument is mandatory.

`condition N range`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of an ER format component. These additional arguments are optional.

`condition N value`: *String*

A value that is returned by the expression that has been configured in the **Collected data key value** property of an ER format component. These additional arguments are optional.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The **Collected data key name** and **Collected data key value** properties can be configured for either the **Sequence** component or the **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

This function returns an empty list when the **Collect output details** option of the current **Common\File** component is turned off.

In `condition range` arguments, the wildcard character "\*" can be used to represent any multiple characters.

In `condition value` arguments, the wildcard character "\*" can be used to represent any multiple characters.

## Example

For more information about how to use this function, see the [ER Use data of format output for counting and summing](#) task guide, which is part of the **Acquire/Develop IT service/solution components** business process.

## Additional resources

[Data collection functions](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# COUNTIF ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `COUNTIF` function returns an *Integer* value that represents the number of format elements that was collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified condition. The condition consists of a key range and a key value.

## Syntax

```
COUNTIF (condition range, condition value)
```

## Arguments

`condition range`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of an Electronic reporting (ER) format component.

`condition value`: *String*

A value that is returned by the expression that has been configured in the **Collected data key value** property of an ER format component.

## Return values

*Integer*

The resulting numeric value.

## Usage notes

The **Collected data key name** and **Collected data key value** properties can be configured for either the **Sequence** component or the **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

This function returns a 0 (zero) value when the **Collect output details** option of the current **Common\File** component is turned off.

In the `condition range` argument, the wildcard character "\*" can be used to represent any multiple characters.

In the `condition value` argument, the wildcard character "\*" can be used to represent any multiple characters.

## Example

For more information about how to use this function, see the [ER Use data of format output for counting and summing](#) task guide, which is part of the **Acquire/Develop IT service/solution components** business process.

## Additional resources

[Data collection functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# COUNTIFS ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `COUNTIFS` function returns an *Integer* value that represents the number of format elements that was collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified conditions. Each condition consists of a key range and a key value.

## Syntax

```
COUNTIFS (condition 1 range, condition 1 value[, condition 2 range, condition 2 value, ..., condition N range, condition N value])
```

## Arguments

`condition 1 range`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of an Electronic reporting (ER) format component. This argument is mandatory.

`condition 1 value`: *String*

A value that is returned by the expression that has been configured in the **Collected data key value** property of an ER format component. This argument is mandatory.

`condition N range`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of an ER format component. These additional arguments are optional.

`condition N value`: *String*

A value that is returned by the expression that has been configured in the **Collected data key value** property of an ER format component. These additional arguments are optional.

## Return values

*Integer*

The resulting numeric value.

## Usage notes

The **Collected data key name** and **Collected data key value** properties can be configured for either the **Sequence** component or the **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

This function returns a 0 (zero) value when the **Collect output details** option of the current **Common\File** component is turned off.

In `condition range` arguments, the wildcard character "\*" can be used to represent any multiple characters.

In `condition value` arguments, the wildcard character "\*" can be used to represent any multiple characters.



## Example

For more information about how to use this function, see the [ER Use data of format output for counting and summing](#) task guide, which is part of the **Acquire/Develop IT service/solution components** business process.

## Additional resources

[Data collection functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# FORMATELEMENTNAME ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `FORMATELEMENTNAME` function returns a *String* value that represents the name of the current Electronic reporting (ER) format's element.

## Syntax

```
FORMATELEMENTNAME ()
```

## Return values

*String*

The resulting text value.

## Usage notes

This function can be called in ER expressions that were configured for the **Collected data key name** and **Collected data key value** properties of an ER format component from the **Text** group that resides under the **Common\File** component where the **Collect output details** option is turned on.

## Example

For more information about how to use this function, see the [ER Use data of format output for counting and summing](#) task guide, which is part of the **Acquire/Develop IT service/solution components** business process.

## Additional resources

[Data collection functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SUMIF ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SUMIF` function returns a *Real* value that represents the sum of values that were returned by bindings of format elements and collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified condition. The condition consists of a key range and a key value.

## Syntax

```
SUMIF (key name for summing, condition range, condition value)
```

## Arguments

`key name for summing`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of the Electronic reporting (ER) format component for which the value of the binding must be used for summing purposes.

The **Collected data key value** property can be configured for either a **Sequence** component or an **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

## Return values

*Real*

The resulting numeric value.

## Usage notes

This function returns a 0 (zero) value when the **Collect output details** option of the current **Common\File** component is turned off.

In the `condition range` argument, the wildcard character "\*" can be used to represent any multiple characters.

In the `condition value` argument, the wildcard character "\*" can be used to represent any multiple characters.

## Example

For more information about how to use this function, see the [ER Use data of format output for counting and summing](#) task guide, which is part of the **Acquire/Develop IT service/solution components** business process.

For more information and examples about using this function, see [Defer the execution of sequence elements in ER formats](#) and [Defer the execution of XML elements in ER formats](#).

## Additional resources

[Data collection functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SUMIFS ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SUMIFS` function returns a *Real* value that represents the sum of values that were returned by bindings of format elements and collected when the format elements were used to generate an outbound document during the format run, and that satisfies the specified conditions. Each condition consists of a key range and a key value.

## Syntax

```
SUMIFS (key name for summing, condition 1 range, condition 1 value[, condition 2 range, condition 2 value, ..., condition N range, condition N value])
```

## Arguments

`key name for summing`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of the Electronic reporting (ER) format component for which the value of the binding must be used for summing purposes.

The **Collected data key name** property can be configured for either a **Numeric** component or a **String** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

`condition 1 range`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of an ER format component. This argument is mandatory.

The **Collected data key name** property can be configured for either a **Sequence** component or an **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

`condition 1 value`: *String*

A value that is returned by the expression that has been configured in the **Collected data key value** property of an ER format component. This argument is mandatory.

The **Collected data key value** property can be configured for either a **Sequence** component or an **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

`condition N range`: *String*

A value that is returned by the expression that has been configured in the **Collected data key name** property of an ER format component. These additional arguments are optional.

The **Collected data key name** property can be configured for either a **Sequence** component or an **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

`condition N value`: *String*

A value that is returned by the expression that has been configured in the **Collected data key value** property of an ER format component. These additional arguments are optional.

The **Collected data key value** property can be configured for either a **Sequence** component or an **XML Element** component of an ER format that resides under the **Common\File** component where the **Collect output details** option is turned on.

## Return values

*Real*

The resulting numeric value.

## Usage notes

This function returns a 0 (zero) value when the **Collect output details** option of the current **Common\File** component is turned off.

In the `condition range` arguments, the wildcard character "\*" can be used to represent any multiple characters.

In the `condition value` arguments, the wildcard character "\*" can be used to represent any multiple characters.

## Example

For more information about how to use this function, see the [ER Use data of format output for counting and summing](#) task guide, which is part of the **Acquire/Develop IT service/solution components** business process.

## Additional resources

[Data collection functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# List of ER functions in the business domain-specific category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) domain-specific functions can be used to perform calculations and data access requests that are specific to the implementation of Microsoft Dynamics 365 Finance. This topic provides a summary of these functions.

## List of supported functions

FUNCTION	DESCRIPTION
<a href="#">CH_Bank_Mod_10</a>	This function returns a <i>String</i> value that represents a creditor reference as an MOD10 expression, based on the digits of the specified invoice number.
<a href="#">CN_GBT_AdditionalDimensionID</a>	This function returns a <i>String</i> value that represents a single financial dimension ID that is taken from the specified string. The specified string presents all dimensions as a comma-separated list of IDs.
<a href="#">ConvertCurrency</a>	This function returns a <i>Real</i> value that represents the result of converting the specified monetary amount from the specified source currency to the specified target currency by using the settings of the specified company on the specified date.
<a href="#">CurCredRef</a>	This function returns a <i>String</i> value that represents a creditor reference, based on the digits of the specified invoice number.
<a href="#">FA_Balance</a>	This function returns a <i>Container (record)</i> value that consists of data for the fixed asset balance for the specified fixed asset item, value model code, reporting year, and reporting date.
<a href="#">FA_Sum</a>	This function returns a <i>Container (record)</i> value that consists of data for the fixed asset amounts for the specified fixed asset item, value model code, and period of dates.
<a href="#">GetCurrentCompany</a>	This function returns a <i>String</i> value that represents the code for the legal entity (company) that a user is currently signed in to.
<a href="#">ISOCredRef</a>	This function returns a <i>String</i> value that represents an International Organization for Standardization (ISO) creditor reference, based on the digits and alphabetic symbols of the specified invoice number.

FUNCTION	DESCRIPTION
<a href="#">IsValidCharacterISO7064</a>	This function returns a <i>Boolean</i> value of <b>TRUE</b> if the specified string represents a valid international bank account number (IBAN). Otherwise, it returns a <i>Boolean</i> value of <b>FALSE</b> .
<a href="#">Mod_97</a>	This function returns a <i>String</i> value that represents a creditor reference as a MOD97 expression, based on the digits of the specified invoice number.
<a href="#">NumSeqValue</a>	This function returns a <i>String</i> value that represents the new generated value of a number sequence, based on the specified number sequence, scope, and scope ID. The scope ID equals the company code that is supplied by the context that the ER format is run under.
<a href="#">RoundAmount</a>	This function returns a <i>Real</i> value that represents the result of rounding the specified amount to the specified number of decimal places according to the specified rounding rule.
<a href="#">TableName2ID</a>	This function returns a numeric representation of the table ID for the specified table name as an <i>Integer</i> value.

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

[Electronic reporting formula language](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# CH\_BANK\_MOD\_10 ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CH_BANK_MOD_10` function returns a *String* value that represents a creditor reference as an MOD10 expression, based on the digits of the specified invoice number.

## Syntax

```
CH_BANK_MOD_10 (invoice number digits)
```

## Arguments

`invoice number digits` : *String*

A text value that represents the digits of an invoice number.

## Return values

*String*

The resulting text value.

## Example

`CH_BANK_MOD_10 ("VEND-200002")` returns 3.

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# CN\_GBT\_ADDITIONALDIMENSIONID ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CN_GBT_ADDITIONALDIMENSIONID` function returns a *String* value that represents a single financial dimension ID that is taken from the specified string. The specified string presents all dimensions as a comma-separated list of IDs.

## Syntax

```
CN_GBT_ADDITIONALDIMENSIONID (text, number)
```

## Arguments

`text` : *String*

A *String* value that presents all dimensions as a comma-separated list of IDs.

`number` : *Integer*

An *Integer* value that defines the sequence code of the requested dimension in the specified string.

## Return values

*String*

The resulting text value.

## Example

```
CN_GBT_AdditionalDimensionID ("AA,BB,CC,DD,EE,FF,GG,HH", 3) returns "CC".
```

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# CONVERTCURRENCY ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CONVERTCURRENCY` function returns a *Real* value that represents the result of converting the specified monetary amount from the specified source currency to the specified target currency by using the settings of the specified company on the specified date.

## Syntax

```
CONVERTCURRENCY (amount, source currency, target currency, date, company)
```

## Arguments

`amount` : *Integer or Real*

A numeric value that represents the monetary amount that must be converted.

`source currency` : *String*

The code of the source currency.

`target currency` : *String*

The code of the target currency.

`date` : *Date*

A *Date* value that represents the date that is used to determine the exchange rate for the conversion.

`company` : *String*

A *String* value that represents the code of a company that supplies the settings that are used for the conversion.

## Return values

*Real*

The resulting numeric value.

## Example

`CONVERTCURRENCY (1, "EUR", "USD", TODAY(), "DEMF")` returns the equivalent of one euro in US dollars on the current session date, based on settings for the **DEMF** company.

## Additional resources

[Other \(business domain-specific\) functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# CURCREDREF ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CURCREDREF` function returns a *String* value that represents a creditor reference, based on the digits of the specified invoice number.

## Syntax

```
CURCREDREF (invoice number digits)
```

## Arguments

`invoice number digits` : *String*

A text value that represents the digits of an invoice number.

## Return values

*String*

The resulting text value.

## Example

```
CURCredRef ("VEND-200002") returns "2200002".
```

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# FA\_BALANCE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `FA_BALANCE` function returns a *Container (record)* value that consists of data for the fixed asset balance for the specified fixed asset item, value model code, reporting year, and reporting date.

## Syntax

```
FA_BALANCE (fixed asset code, value model code, reporting year, reporting date)
```

## Arguments

`fixed asset code` : *String*

A *String* value that represents the code of a fixed asset item that the balance is calculated for.

`value model code` : *String*

A *String* value that represents the code of a value model that the balance is calculated for.

`reporting year` : *Enumeration value*

An enumeration value of the **AssetYear** application enumeration that defines a period for the balance calculation.

`reporting date` : *Date*

A *Date* value that defines a date for the balance calculation.

## Return values

*Container (record)*

The resulting record value.

## Example

`FA_BALANCE ("COMP-000001", "Current", AxEnumAssetYear.ThisYear, SESSSIONTODAY ())` returns the data container of balances for fixed asset **COMP-000001** that has been prepared for the **Current** value model on the current application session date.

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# FA\_SUM ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `FA_SUM` function returns a *Container (record)* value that consists of data for the fixed asset amounts for the specified fixed asset item, value model code, and period of dates.

## Syntax

```
FA_SUM (fixed asset code, value model code, start date, end date)
```

## Arguments

`fixed asset code` : *String*

A *String* value that represents the code of a fixed asset item that the balance is calculated for.

`value model code` : *String*

A *String* value that represents the code of a value model that the balance is calculated for.

`start date` : *Date*

A *Date* value that represents the start date of a period that the fixed asset amounts are calculated for.

`end date` : *Date*

A *Date* value that represents the end date of a period that the fixed asset amounts are calculated for.

## Return values

*Container (record)*

The resulting record value.

## Example

`FA_SUM ("COMP-000001", "Current", Date1, Date2)` returns the data container for fixed asset **COMP-000001** that has been prepared for the **Current** value model and for a period from **Date1** to **Date2**.

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# GETCURRENTCOMPANY ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `GETCURRENTCOMPANY` function returns a *String* value that represents the code for the legal entity (company) that a user is currently signed in to.

## Syntax

```
GETCURRENTCOMPANY ()
```

## Return values

*String*

The resulting text value.

## Example

`GETCURRENTCOMPANY ()` returns **USMF** for a user who is signed in to the **Contoso Entertainment System USA** company.

## Additional resources

[Other \(business domain–specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# ISOCREDREF ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ISOCREDREF` function returns a *String* value that represents an International Organization for Standardization (ISO) creditor reference, based on the digits and alphabetic symbols of the specified invoice number.

## Syntax

```
ISOCREDREF (invoice number digits)
```

## Arguments

`invoice number digits` : *String*

A text value that represents the digits of an invoice number.

## Return values

*String*

The resulting text value.

## Usage notes

### NOTE

To eliminate symbols from alphabets that aren't ISO-compliant, the `invoice number digits` argument must be translated before it's passed to this function.

## Example

`ISOCredRef ("VEND-200002")` returns "RF23VEND-200002".

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ISVALIDCHARACTERISO7064 ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ISVALIDCHARACTERISO7064` function returns a *Boolean* value of **TRUE** if the specified string represents a valid international bank account number (IBAN). Otherwise, it returns a *Boolean* value of **FALSE**.

## Syntax

```
ISVALIDCHARACTERISO7064 (text)
```

## Arguments

`text` : *String*

A text value that represents an IBAN.

## Return values

*String*

The resulting text value.

## Example

```
ISVALIDCHARACTERISO7064 ("AT61 1904 3002 3457 3201") returns TRUE.
```

```
ISVALIDCHARACTERISO7064 ("AT61") returns FALSE.
```

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# MOD\_97 ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `MOD_97` function returns a *String* value that represents a creditor reference as a MOD97 expression, based on the digits of the specified invoice number.

## Syntax

```
MOD_97 (invoice number digits)
```

## Arguments

`invoice number digits` : *String*

A text value that represents the digits of an invoice number.

## Return values

*String*

The resulting text value.

## Example

`MOD_97 ("VEND-200002")` returns "20000285".

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NUMSEQVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NUMSEQVALUE` function returns a *String* value that represents the new generated value of a number sequence, based on the specified number sequence, scope, and scope ID. The scope ID equals the company code that is supplied by the context that the Electronic reporting (ER) format is run under.

## Syntax 1

```
NUMSEQVALUE (number sequence code)
```

## Syntax 2

```
NUMSEQVALUE (number sequence record ID)
```

## Syntax 3

```
NUMSEQVALUE (number sequence code, scope type, scope ID)
```

## Arguments

`number sequence code` : *String*

A text value that represents the code of the number sequence that a new value is required in.

`number sequence record ID` : *Int64*

An *Int64* value that represents the record ID of a record in the `NumberSequenceTable` table that contains the definition of the number sequence that a new value is required in.

`scope type` : *Enum value*

An enumeration value of the `ERExpressionNumberSequenceScopeType` enumeration that defines the scope of the number sequence that a new value is required in. The available scope types are **Shared**, **Legal entity**, and **Company**.

`scope ID` : *String*

A *String* value that identifies the scope, based on the specified scope type.

## Return values

*String*

The resulting text value.

## Usage notes

For the **Shared** scope type, specify an empty string as the scope ID.

For the **Company** and **Legal entity** scope types, specify the company code as the scope ID. If you specify an empty string as the scope ID for these scope types, the current company code is used.

When syntax 1 is used, the number sequence is requested for the **Company** scope type, and the company code is supplied by the context that the ER format is run under.

## Example 1

In your ER format, you define the **AskNumSeq** data source of the *User input parameter* type. This data source refers to the **Description** extended data type (EDT). Next, you define the **NumSeq** data source of the *Calculated field* type. This data source contains the expression `NUMSEQVALUE (AskNumSeq)`. When the **NumSeq** data source is called, it returns the new generated value of the number sequence that was specified at runtime by entering its code in the dialog box. The number sequence is requested for the **Company** scope type. The company code is supplied by the context that the ER format is run under.

## Example 2

The following data sources are defined in your model mapping:

- The **LedgerParms** data source of the *Table* type. This data source refers to the LedgerParameters table.
- The **NumSeq** data source of the *Calculated field* type. This data source contains the expression

```
NUMSEQVALUE ( LedgerParameters.'numRefJournalNum()'.NumberSequenceId )
```

When the **NumSeq** data source is called, it returns the new generated value of the number sequence that has been configured in the General ledger parameters for the company that supplies the context that the ER format is run under. This number sequence uniquely identifies journals and acts as a batch number that links the transactions together.

## Example 3

The following data sources are defined in your model mapping:

- The **enumScope** data source of the Microsoft Dynamics 365 Finance *enumeration* type. This data source refers to the **ERExpressionNumberSequenceScopeType** enumeration.
- The **NumSeq** data source of the *Calculated field* type. This data source contains the expression

```
NUMSEQVALUE ("Gene_1", enumScope.Company, "")
```

When the **NumSeq** data source is called, it returns the new generated value of the **Gene\_1** number sequence that has been configured for the company that supplies the context that the ER format is run under.

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ROUNDAMOUNT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `ROUNDAMOUNT` function returns a *Real* value as the result of the rounding of the specified number to the nearest multiple of another number according to the specified rounding rule.

## Syntax

```
ROUNDAMOUNT (number, decimals, round rule)
```

## Arguments

`number` : *Int* or *Real*

A numeric value that must be rounded.

`decimals` : *Int* or *Real*

The number that the value of the `number` parameter must be rounded to a multiple of.

`round rule` : *Enum value*

An enumeration value of the `RoundOffType` enumeration that defines the rounding rule. This enumeration offers the following values:

- Normal (Ordinary)
- Downward (RoundDown)
- Rounding-up (RoundUp)

## Return values

*Real*

The resulting numeric value is a multiple of the value specified by the `decimals` parameter and is closest to the value specified by the `number` parameter.

## Usage notes

When the `number` parameter is zero, this function always returns zero.

When the `decimals` parameter is zero, this function rounds to the default round-off value. When the `round rule` parameter is set to `RoundOffType.Ordinary`, the default round-off value is `0.01`. Otherwise, the default round-off value is `1.0`.

When the `round rule` parameter is set to `RoundOffType.Ordinary`, this function rounds to the nearest round-off amount.

When the `round rule` parameter is set to `RoundOffType.RoundDown`, this function rounds towards zero to the nearest round-off amount.

When the `round rule` parameter is set to `RoundOffType.RoundUp`, this function rounds away from zero to the nearest round-off amount.

When the `round rule` parameter is set to **RoundOffType.Ordinary**, this function behaves like the [MROUND](#) Excel function and the [ROUND](#) X++ function.

## Remarks

To round a numeric value to a specified number of decimal places, use the [ROUND](#) function.

## Example

If the `model.RoundOff` parameter is set to **RoundOffType.Ordinary**,

```
ROUNDAMOUNT (7.45, 1.05, model.RoundOff) returns 7.35.
```

If the `model.RoundOff` parameter is set to **RoundOffType.RoundDown**,

```
ROUNDAMOUNT (7.45, 1.05, model.RoundOff) returns 7.35.
```

If the `model.RoundOff` parameter is set to **RoundOffType.RoundUp**,

```
ROUNDAMOUNT (7.45, 1.05, model.RoundOff) returns 8.4.
```

## Additional resources

[Other \(business domain-specific\) functions](#)

[Mathematical functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# TABLENAME2ID ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `TABLENAME2ID` function returns a numeric representation of the table ID for the specified table name as an *Integer* value.

## Syntax

```
TABLENAME2ID (text)
```

## Arguments

`text` : *String*

A text value that represents a valid table name.

## Return values

*Integer*

The resulting numeric value.

## Usage notes

Execution of this function can have different results in different instances of Microsoft Dynamics 365 Finance, even if the same company name is used.

## Example

`TABLENAME2ID ("Intrastat")` returns 1510.

## Additional resources

[Other \(business domain-specific\) functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# List of ER functions in the type conversion category

2/18/2021 • 2 minutes to read • [Edit Online](#)

Electronic reporting (ER) type conversion functions can be used to convert values between types. This topic provides a summary of these functions.

## Type conversion functions

FUNCTION	DESCRIPTION
<a href="#">Int64Value</a>	This function returns an <i>Int64</i> value that represents the specified string.
<a href="#">IntValue</a>	This function returns an <i>Int</i> value that represents the specified string.
<a href="#">NumberValue</a>	This function returns a <i>Real</i> value that is converted from the specified <i>String</i> value. During the conversion, the specified decimal and digit grouping separators are considered.
<a href="#">Value</a>	This function returns a <i>Real</i> value that is converted from the specified <i>String</i> value.

## Type conversion functions in the container category

The following table describes the type conversion functions in the [container](#) category.

FUNCTION	DESCRIPTION
<a href="#">Base64StringToContainer</a>	This function converts the specified input of the <i>String</i> type to a data item of the <i>Container</i> type.

## Type conversion functions in the date and time category

The following table describes the type conversion functions in the [date and time](#) category.

FUNCTION	DESCRIPTION
<a href="#">DateTimeValue</a>	This function returns a <i>DateTime</i> value that is converted from a given <i>String</i> value in the specified format and in an optionally specified culture to a date/time value.
<a href="#">DateToDateTime</a>	This function returns a <i>DateTime</i> value that is converted from a given <i>Date</i> value to a date/time value in Coordinated Universal Time (Greenwich Mean Time [GMT]).
<a href="#">DateValue</a>	This function returns a <i>Date</i> value that is converted from a given <i>String</i> value in the specified format and in an optionally specified culture to a date value.

## Type conversion functions in the list category

The following table describes the type conversion functions in the [list category](#).

FUNCTION	DESCRIPTION
<a href="#">List</a>	This function returns a <i>Record list</i> value as a new list that is created from specified arguments of the <i>Container (record)</i> type.
<a href="#">ListOfFields</a>	This function returns a <i>Record list</i> value that is created based on the structure of a given argument of the <i>Enumeration</i> or <i>Container (record)</i> type.
<a href="#">Split</a>	This function splits the specified <i>String</i> value into substrings and returns the result as a new <i>Record list</i> value.
<a href="#">StringJoin</a>	This function returns a <i>String</i> value that consists of concatenated values of the specified field from the specified <i>Record list</i> value. The values can be separated by the specified delimiter.

## Type conversion functions in the text category

The following table describes the type conversion functions in the [text category](#).

FUNCTION	DESCRIPTION
<a href="#">Char</a>	This function returns a <i>String</i> value that represents a single character that is referenced by the specified Unicode number.
<a href="#">GuidValue</a>	This function converts the specified input of the <i>String</i> type to a data item of the <i>GUID</i> type.
<a href="#">NumberFormat</a>	This function returns a <i>String</i> value that represents the specified number in the specified format and in an optionally specified culture.
<a href="#">QrCode</a>	This function returns a <i>Container</i> value that presents the Quick Response code (QR code) image for the specified string in binary format.
<a href="#">Text</a>	This function returns a <i>String</i> value that represents the specified number after it has been converted to a text string that is formatted according to the server locale settings of the current application instance.

## Additional resources

[Electronic Reporting overview](#)

[Formula designer in Electronic reporting](#)

[Electronic reporting formula language](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# INT64VALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `INT64VALUE` function returns an *Int64* value that represents the specified string.

## Syntax 1

```
INT64VALUE (text)
```

## Syntax 2

```
INT64VALUE (number)
```

## Arguments

`text` : *String*

A text value that must be converted to an *Int64* number.

`number` : *Real or Integer*

A numeric *Real* or *Integer* value that must be converted to an *Int64* number.

## Return values

*Int64*

The resulting numeric value.

## Usage notes

Any decimal places are truncated.

## Example 1

```
INT64VALUE ("22565422744")
```

 returns the *Int64* value 22565422744.

## Example 2

```
INT64VALUE ( VALUE("22565422744.77"))
```

 returns the *Int64* value 22565422744.

## Additional resources

[Type conversion functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# INTVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `INTVALUE` function returns an *Int* value that represents the specified string.

## Syntax 1

```
INTVALUE (text)
```

## Syntax 2

```
INTVALUE (number)
```

## Arguments

`text` : *String*

A text value that must be converted to an *Int* number.

`number` : *Real or Integer*

A numeric *Real* or *Integer* value that must be converted to an *Int* number.

## Return values

*Int*

The resulting numeric value.

## Usage notes

Any decimal places are truncated.

## Example 1

```
INTVALUE ("100.77")
```

 returns the *Int* value 100.

## Example 2

```
INTVALUE (-100.77)
```

 returns the *Int* value -100.

## Additional resources

[Type conversion functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# NUMBERVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NUMBERVALUE` function returns a *Real* value that is converted from the specified *String* value. During the conversion, the specified decimal and digit grouping separators are considered.

## Syntax

```
NUMBERVALUE (text, decimal separator, digit grouping separator)
```

## Arguments

`text` : *String*

A text value that must be converted to a *Real* number.

`decimal separator` : *String*

A decimal separator. It's used to separate the integer and fractional parts of a decimal number.

`digit grouping separator` : *String*

A digit grouping separator. It's used as the thousands separator.

## Return values

*Real*

The resulting numeric value.

## Example

`NUMBERVALUE( "1 234,56", ",", " ")` returns 1234.56.

## Additional resources

[Type conversion functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# VALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `VALUE` function returns a *Real* value that is converted from the specified string.

## Syntax

```
VALUE (text)
```

## Arguments

`text` : *String*

A string value that must be converted to a numeric value.

## Return values

*Real*

The resulting numeric value.

## Usage notes

Commas and dot characters (.) are considered decimal separators, and a leading hyphen (-) is used as a negative sign. An exception is thrown at runtime if the specified string contains other non-numeric characters.

## Example 1

`VALUE ("1 234,56")` throws an exception.

## Example 2

`VALUE ("1234,56")` returns **1234.56**.

## Additional resources

[Type conversion functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DATETIMEVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATETIMEVALUE` function returns a *DateTime* value that is converted from a given text value in the specified format and in an optionally specified `culture` to a date/time value. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
DATETIMEVALUE (text, format)
```

## Syntax 2

```
DATETIMEVALUE (text, format, culture)
```

## Arguments

`text` : *String*

Text that represents the value to format.

`format` : *String*

The format of the given text.

`culture` : *String*

The culture that is used for formatting of the given text.

## Return values

*DateTime*

The resulting date/time value.

## Usage notes

When the culture isn't defined as an argument of the called function, the value of `culture` is defined by the calling context. For example, if the `DATETIMEVALUE` function is called by using syntax 1 in an Electronic reporting (ER) format for a FILE element that is configured to use the German culture, the conversion will be done by using the German culture. The default `culture` value is **EN-US**.

## Example 1

`DATETIMEVALUE ("21-Dec-2016 02:55:00", "dd-MMM-yyyy hh:mm:ss")` returns 2:55:00 AM on December 21, 2016, based on the specified custom format and the default application's **EN-US** culture.

## Example 2

`DATETIMEVALUE ("21-Jan-2016 02:55:00", "dd-MMM-yyyy hh:mm:ss", "IT")` returns **2:55:00 AM on December 21, 2016**, based on the specified custom format and culture.

However, `DATETIMEVALUE ("21-Jan-2016 02:55:00", "dd-MMM-yyyy hh:mm:ss", "EN-US")` throws an exception to inform the user that the specified string isn't recognized as a valid date/time value for the specified culture.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DATE TODATETIME ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATE TODATETIME` function returns a *DateTime* value that is converted from a given date value to a date/time value in Coordinated Universal Time (Greenwich Mean Time [GMT]).

## Syntax

```
DATE TODATETIME (date)
```

## Arguments

`date` : *Date*

A date value that represents the date to convert.

## Return values

*DateTime*

The resulting date/time value.

## Example 1

`DATE TODATETIME (CompInfo. 'getCurrentDate()')` returns the date of the current Microsoft Dynamics 365 Finance session, December 24, 2015, as **12/24/2015 12:00:00 AM**. In this example, **CompInfo** is an Electronic reporting (ER) data source of the **Finance and Operations/Table** type, and it refers to the CompanyInfo table.

## Example 2

`DATE TODATETIME (DATEVALUE ("2019-11-12T16:00:00.0000000-07:00", "O"))` returns the date/time value **11/12/2019 12:00:00 AM**.

## Additional resources

[Date and time functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# DATEVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `DATEVALUE` function returns a *Date* value that is converted from a given text value in the specified format and in an optionally specified *culture* to a date value. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
DATEVALUE (text, format)
```

## Syntax 2

```
DATEVALUE (text, format, culture)
```

## Arguments

`text` : *String*

Text that represents the value to format.

`format` : *String*

The format of the given text.

`culture` : *String*

The culture that is used for formatting of the given text.

## Return values

*Date*

The resulting date value.

## Usage notes

When the culture isn't defined as an argument of the called function, the value of `culture` is defined by the calling context. For example, if the `DATEVALUE` function is called by using syntax 1 in an Electronic reporting (ER) format for a **FILE** element that is configured to use the German culture, the conversion will be done by using the German culture. The default `culture` value is EN-US.

## Example 1

`DATEVALUE ("21-Dec-2016", "dd-MMM-yyyy")` returns the date value **December 21, 2016**, based on the specified custom format and the default application's EN-US culture.

## Example 2

`DATEVALUE ("21-Jan-2016", "dd-MMM-yyyy", "IT")` returns the date value **January 21, 2016**, based on the specified custom format and culture.

However, `DATEVALUE ("21-Jan-2016", "dd-MMM-yyyy", "EN-US")` throws an exception to inform the user that the specified string isn't recognized as a valid date for the specified culture.

## Additional resources

### [Date and time functions](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# LIST ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `LIST` function returns a *Record list* value that consists of a new list of records that is created from the specified arguments.

## Syntax

```
LIST (record 1 [, record 2, ..., record N])
```

## Arguments

`record 1` : *Container (record)*

A reference to a data source of the *Record* data type. This argument is required.

`record N` : *Container (record)*

A reference to a data source of the *Record* data type. These additional arguments are optional.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The structure of the list that is created contains only the fields that are presented in the structure of every record that is mentioned in the arguments.

## Example

You enter data source **Record 1** of the *Container* type. This data source contains the following nested fields of the *Calculated field* type:

- **Code**: This field contains an expression that returns a value of the *String* type.
- **Amount**: This field contains an expression that returns a value of the *Real* type.

You then enter data source **Record 2** of the *Container* type. This data source contains the following nested fields of the *Calculated field* type:

- **Amount**: This field contains an expression that returns a value of the *Real* type.
- **IsValid**: This field contains an expression that returns a value of the *Boolean* type.

In this case, the expression `LIST('Record 1', 'Record 2')` returns a new list that contains two records. The structure of this list consists of a single **Amount** field of the *Real* type, because this field is the only field that is presented in every argument of the called function.

## Additional resources

## List functions

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# LISTOFFIELDS ER function

2/18/2021 • 3 minutes to read • [Edit Online](#)

The `LISTOFFIELDS` function returns a *Record list* value that is created based on the structure of the specified argument of the *Enumeration* or *Container (record)* type.

## Syntax 1

```
LISTOFFIELDS (path)
```

## Syntax 2

```
LISTOFFIELDS (path, language)
```

## Arguments

`path`: Data source reference

The valid reference path of a data source of one of the following data types:

- Model enumeration
- Format enumeration
- Application enumeration
- Container (record)

`language`: *String*

Text that represents a language code.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The list that is created consists of records that have the following fields:

- **Name** (*String* data type)
- **Label** (*String* data type)
- **Description** (*String* data type)
- **IsTranslated** (*Boolean* data type)

If the `path` argument refers to a data source of the *Container (Record)* type, for every field of the referenced container record, a new record is added to the list that is created. For every record that is created, the **Name** field returns the name of the field of the referenced container record that the current record was created for.

If the `path` argument refers to a data source of one of the *Enumeration* types, for every enumeration value of

the referenced enumeration, a new record is added to the list that is created. For every record that is created, the **Name** field returns the value of the referenced enumeration that the current record was created for, the **Description** field returns the description of that enumeration, and the **Label** field returns the label of that enumeration.

At runtime, when syntax 1 is used, the **Label** and **Description** fields must return values that are based on the language settings of the Electronic reporting (ER) format that is running:

- If the labels and descriptions for the requested language are available, the **Label** and **Description** fields return values that are based on that language, and the **IsTranslated** field returns **True**.
- If the labels and descriptions for the requested language aren't available, the **Label** and **Description** fields return values that are based on the default **EN-US** language, and the **IsTranslated** field returns **False**.

At runtime, when syntax 2 is used, the **Label** and **Description** fields must return values that are based on the language that is defined as the second argument of the called function:

- If the labels and descriptions for the requested language are available, the **Label** and **Description** fields return values that are based on that language, and the **IsTranslated** field returns **True**.
- If the labels and descriptions for the requested language aren't available, the **Label** and **Description** fields return values that are based on the **EN-US** language, and the **IsTranslated** field returns **False**.

## Example 1

In the following illustration, an enumeration is introduced in an ER data model.

ReportDirection		
Enumeration values		
+ New <span style="float:right">Delete</span>		
Name	Label (*Recommended to use labels)	Description
Both	Both	
Export	Dispatches	
Import	Arrivals	

The following illustration shows these details:

- The model enumeration is inserted into a report as a data source.
- An ER expression uses the model enumeration as a parameter of the `LISTOFFIELDS` function.
- A data source of the *Record list* type is inserted into a report by using the ER expression that is created.

FORMAT **MAPPING** TRANSFORMATIONS VALIDATIONS

Bind + Add root + Add Edit Delete Show name first Group view

- enumDirectionInReport: Data model enumeration ReportDirection
  - Arrivals(Import): Enumeration value
  - Both: Enumeration value
  - Dispatches(Export): Enumeration value
  - listDirectionInReport: Calculated field = LISTOFFIELDS(enumDirectionInReport): Record list**
    - Description: String
    - Label: String
    - Name: String
- model: Data model Intrastat

The following example shows the ER format elements that are bound to the data source of the *Record list* type that was created by using the `LISTOFFIELDS` function.

```

Root: XML Element
├── Directions: XML Element = listDirectionInReport
│   ├── name: XML Attribute = listDirectionInReport.Name
│   ├── label: XML Attribute = listDirectionInReport.Label
│   └── desc: XML Attribute = listDirectionInReport.Description

```

The following illustration shows the result when the designed format is run.

```

<?xml version="1.0" encoding="UTF-8"?>
- <Root>
  <Directions desc="" label="Both" name="Both"/>
  <Directions desc="" label="Dispatches" name="Export"/>
  <Directions desc="" label="Arrivals" name="Import"/>
</Root>

```

**NOTE**

Based on the language settings of the parent FILE and FOLDER format elements, translated text for labels and descriptions is entered in the output of the ER format.

## Example 2

You use the *Calculated field* data source type to configure `enumType_de` and `enumType_deCH` data sources for the `enumType` data model enumeration:

- `enumType_de` = `LISTOFFIELDS (enumType, "de")`
- `enumType_deCH` = `LISTOFFIELDS (enumType, "de-CH")`

In this case, you can use the following expression to get the label of the enumeration value in Swiss German, if that translation is available. If the Swiss German translation isn't available, the label is in German.

```
IF (NOT (enumType_deCH.IsTranslated), enumType_de.Label, enumType_deCH.Label)
```

## Additional resources

[List functions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SPLIT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `SPLIT` function splits the specified input string into substrings and returns the result as a new *Record list* value.

## Syntax 1

```
SPLIT (input, length)
```

This syntax is used to split the specified input string into substrings, each of which has the specified length.

## Syntax 2

```
SPLIT (input, delimiter)
```

This syntax is used to split the specified input string into substrings, based on the specified delimiter.

## Arguments

`input` : *String*

The text to split.

`length` : *Integer*

The maximum length of a single substring.

`delimiter` : *String*

A delimiter that is used to separate substrings.

## Return values

*Record list*

The resulting list of records.

## Usage notes

The record structure of the list that is returned consists of the **Value** field of the *String* type. Every record of the list that is returned contains generated substrings in this field.

If the `delimiter` argument is empty, the new list that is returned consists of one record that has the **Value** field of the *String* type. This field contains the input text.

If the `input` argument is empty, a new empty list is returned. If either the `input` or `delimiter` argument is unspecified (null), an application exception is thrown.

## Example 1

`SPLIT ("abcd", 3)` returns a new list that consists of two records that have the **Value** field of the *String* type. The **Value** field in the first record contains the text "abc", and the **Value** field in the second record contains the text "d".

## Example 2

`SPLIT ("XAb aBy", "aB")` returns a new list that consists of three records that have the **Value** field of the *String* type. The **Value** field in the first record contains the text "X", the **Value** field in the second record contains the text " ", and the **Value** field in the third record contains the text "y".

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# STRINGJOIN ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `STRINGJOIN` function returns a *String* value that consists of concatenated values of the specified field from the specified list. The values can be separated by the specified delimiter.

## Syntax

```
STRINGJOIN (list, field, delimiter)
```

## Arguments

`list` : *Record list*

The valid path of a data source of the *Record list* data type.

`field` : *Field*

The valid path of a field of the *String* data type in the specified list.

`delimiter` : *String*

A delimiter that is used to separate substrings.

## Return values

*String*

The resulting text value.

## Example

If you enter `SPLIT("abc" , 1)` as data source `DS`, the expression `STRINGJOIN (DS, DS.Value, "-")` returns "a-b-c".

## Additional resources

[List functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# CHAR ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `CHAR` function returns a *String* value that presents a single character that is referenced by the specified Unicode number.

## Syntax

```
CHAR (number)
```

## Arguments

`number` : *Integer*

A number that corresponds to an expected single character.

## Return values

*String*

The resulting text value.

## Usage notes

The string that this function returns depends on the encoding that is selected in the parent **FILE** format element. For a list of the supported encodings, see [Encoding class](#).

## Example

`CHAR (255)` returns "ÿ".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# GUIDVALUE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `GUIDVALUE` function converts the specified input of the *String* type to a data item of the *GUID* type.

## Syntax

```
GUIDVALUE (input)
```

## Arguments

`input` : *String*

The valid path of a data source of the *String* type.

## Return values

*GUID*

The resulting globally unique identifier (GUID) value.

## Usage notes

To do a conversion in the opposite direction (that is, to convert specified input of the *GUID* data type to a data item of the *String* data type), you can use the [TEXT](#) function.

## Example

You define the following data sources in your model mapping:

- A `myID` data source of the *Calculated field* type that contains the expression `GUIDVALUE ("AF5CCDAC-F728-4609-8C8B- A4B30B0C0AA0")`
- A `Users` data source of the *Table records* type that refers to the `UserInfo` table

You can then use an expression such as `FILTER (Users, Users.objectId = myID)` to filter the `UserInfo` table by the `objectId` field of the *GUID* data type.

## Additional resources

### [Text functions](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# NUMBERFORMAT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `NUMBERFORMAT` function returns a *String* value that presents the specified number in the specified format and in an optionally specified *culture*. For information about the supported formats, see [standard](#) and [custom](#).

## Syntax 1

```
NUMBERFORMAT (number, format)
```

## Syntax 2

```
NUMBERFORMAT (number, format, culture)
```

## Arguments

`number` : *Integer or Real*

A numeric value that specifies the number that must be formatted.

`format` : *String*

A *String* value that represents the format.

`culture` : *String*

A *String* value that represents the culture to use for formatting.

## Return values

*String*

The resulting text value.

## Usage notes

If the culture isn't defined as an argument of the called function, the context that this function is run in determines the culture that is used to format numbers.

## Example 1

For the EN-US culture, `NUMBERFORMAT (0.45, "p")` returns "45.00 %", and `NUMBERFORMAT (10.45, "#")` returns "10".

## Example 2

`NUMBERFORMAT (10/3, "F2", "de")` returns 3,33, whereas `NUMBERFORMAT (10/3, "F2", "en-us")` returns 3.33.

## Additional resources

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# QRCODE ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `QRCODE` function returns a *Container* value that presents the Quick Response code (QR code) image for the specified string in binary format.

## Syntax

```
QRCODE (text)
```

## Arguments

`text` : *String*

A *String* value that represents the original text.

## Return values

*Container*

The resulting binary stream.

## Example

You can configure an Electronic reporting (ER) format to generate an outbound document in Microsoft Office format (Excel workbooks or Word documents) by using a predefined template. This template may contain a **Picture** object (Excel workbook) or a **Picture Content Control** (Word document) as a placeholder for a QR code image. You need to add to the configured ER format a **Cell** element that will be used to fill this placeholder in. To specify what information will be stored in a QR code, you need to define a binding for this **Cell** element. For example, you can configure such binding as containing the following expression:

```
QRCODE (model.ListOfShelfLabels.LabelText)`
```

When you run the configured ER format, the text value of the **LabelText** field of the **model.ListOfShelfLabels** data source will be used to generate a QR code image. This image will replace a QR code image placeholder in the document template using to generate an outbound document. When this image of the generated document is scanned, it returns the text that was taken from the **LabelText** field of the **model.ListOfShelfLabels** data source. For more information, see [Embed images and shapes in documents that you generate by using ER](#).

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# TEXT ER function

2/18/2021 • 2 minutes to read • [Edit Online](#)

The `TEXT` function returns the specified number as a *String* value after it has been converted to a text string that is formatted according to the server locale settings of the current application instance.

## Syntax

```
TEXT (number)
```

## Arguments

`number` : *Integer or Real*

A number that must be converted to a text string.

## Return values

*String*

The resulting text value.

## Usage notes

For values of the *Real* type, the string conversion is limited to two decimal places.

## Example

If the server locale of the Microsoft Dynamics 365 Finance instance is defined as **EN-US**, `TEXT (NOW ())` returns the current Finance session date, December 17, 2015, as the text string "12/17/2015 07:59:23 AM".

`TEXT (1/3)` returns "0.33".

## Additional resources

[Text functions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage ER model mapping in separate ER configurations

2/18/2021 • 5 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the System administrator or Electronic reporting developer role can manage Electronic reporting (ER) model mappings in separate ER configurations. In this task guide, you will create required ER configurations for the sample company, Litware, Inc. To complete this task guide, you must first complete the steps in the task guide, "ER Create a configuration provider" and mark it as active.

Because ER configurations are shared among companies, you can complete this task guide using the company data set of your choice. The functionality for this task guide is available if you have installed one of the following hotfixes: <https://fix.lcs.dynamics.com/Issue/Resolved?kb=4012872> for the Dynamics AX 7.0 version or <https://fix.lcs.dynamics.com/Issue/Resolved?kb=4012871> for the Dynamics 365 for Operations version.

1. Go to Organization administration > Workspaces > Electronic reporting.
  - Verify that the configuration provider for the sample company Litware, Inc. is available and marked as active. If you don't see this configuration provider, you must first complete the steps in the task guide, Create a configuration provider, and mark it as active.

## Add a new ER model configuration

1. Click Reporting configurations.
  - Add a new model configuration. The name must be unique in the configurations tree.
2. Click Create configuration to open the drop dialog.
3. In the Name field, type 'Sample data model'.
  - Sample data model
4. Click Create configuration.
5. Click Designer.
6. Click New to open the drop dialog.
7. In the Name field, type 'Root'.
  - Root
8. Click Add.
9. Click New to open the drop dialog.
10. In the Name field, type 'Company'.
  - Company
11. Click Add.
12. In the Description field, enter the text, Description of the legal entity or company in which a user logged at run-time.
  - Description of the legal entity or company in which a user logged at run-time.
13. Click Root reference.
14. Click OK.
15. Click Save.
16. Close the page.
17. Click Change status.
18. Click Complete.
19. Click OK.

## Add a new ER model-mapping configuration

1. Click Create configuration to open the drop dialog.
2. In the New field, enter 'Model Mapping based on data model Sample data model'.
3. In the Name field, type 'Sample mapping'.
  - Sample mapping
4. Click Create configuration.
5. Expand the Prerequisites section.
  - The Implementations prerequisites group has been added automatically. The group contains the prerequisite component that refers to the parent data model configuration and is marked as Implementation. This means that this Sample-mapping model-mapping configuration is considered the implementation of the data model, Sample data model. Therefore, this component will force ER to download the model-mapping configuration, Sample mapping from an ER repository when the model configuration, Sample data model, is downloaded.
6. Click Designer.
  - The created model-mapping configuration contains a new blank mapping with the same name as the created configuration. When a selected parent model configuration contains model mappings, they will be copied to a new model-mapping configuration.
7. Click Designer.
8. In the tree, select 'Dynamics 365 for Operations\Table'.
9. Click Add root.
10. In the Name field, type 'Company'.
  - Company
11. In the Table field, type 'CompanyInfo'.
  - CompanyInfo
12. Click OK.
13. In the tree, expand 'Company'.
14. In the tree, expand 'Company\find()'.
15. In the tree, select 'Company\find()\Name'.
16. Click Bind.
17. Click Save.
18. Close the page.
19. Close the page.
20. On the Action Pane, click Configurations.
21. Click User parameters.
22. Select Yes in the Run settings field.
23. Click OK.
24. Click Edit.
25. Select Yes in the Run Draft field.

## Add a new ER format configuration

1. In the tree, select 'Sample data model'.
2. Click Create configuration to open the drop dialog.
3. In the New field, enter 'Format based on data model Sample data model'.
4. In the Name field, type 'Sample format'.
  - Sample format
5. Click Create configuration.

6. Click Designer.
7. Click Add root to open the drop dialog.
8. In the tree, select 'Text\String'.
9. Click OK.
10. Click the Mapping tab.
11. In the tree, expand 'model'.
12. In the tree, select 'model\Company'.
13. Click Bind.
14. Click Save.
15. Close the page.
  - Run the draft version of the created format for testing purposes.
16. Click Run.
  - On the Versions FastTab, click Run.
17. Click OK.
  - Review the output that contains the name of the company in which the user who is running this format configuration is logged into. The created model-mapping configuration is used by this format configuration because there is only one configuration available that contains required model mappings.

## Add alternative ER model-mapping configuration

1. In the tree, select 'Sample data model'.
2. Click Create configuration to open the drop dialog.
3. In the New field, enter 'Model Mapping based on data model Sample data model'.
4. In the Name field, type 'Sample mapping (alternative)'.
  - Sample mapping (alternative)
5. Click Create configuration.
6. Click Designer.
7. Click Designer.
8. In the tree, select 'Dynamics 365 for Operations\Table'.
9. Click Add root.
10. In the Name field, type 'Company'.
  - Company
11. In the Table field, type 'CompanyInfo'.
  - CompanyInfo
12. Click OK.
13. Click Edit.
14. In the tree, select 'String\CONCATENATE'.
15. Click Add function.
16. In the tree, expand 'Company'.
17. In the tree, expand 'Company\find()'.
18. In the tree, select 'Company\find()\Name'.
19. Click Add data source.
20. In the Formula field, type a value.
  - CONCATENATE(Company.'find()'.Name, ";",
21. In the tree, select 'Company\find()\Company(DataArea)'.
22. Click Add data source.

23. In the Formula field, type a value.
  - `CONCATENATE(Company.find().Name, ";", Company.find().DataArea)`
24. Click Save.
25. Close the page.
26. Click Save.
27. Close the page.
28. Close the page.
29. Select Yes in the Run Draft field.

## Use an existing ER model-mapping configuration

1. In the tree, select 'Sample data model\Sample format'.
2. Click Run.
  - The selected draft version of the ER format configuration can't be executed because there is more than one model-mapping configuration available for the undefined data model that has been selected as the data source of the running ER format.
  - Next, you will define the alternative model-mapping configuration as the one from which model mappings will be used as data sources for running ER format.
3. In the tree, select 'Sample data model\Sample mapping (alternative)'.
4. Select Yes in the Default for model-mapping field.
5. In the tree, select 'Sample data model\Sample format'.
6. Click Run.
7. Click OK.
  - The default model-mapping configuration is used by this format configuration for generating the electronic document (the created output contains the company code).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Select data model definitions when you create formats

2/18/2021 • 3 minutes to read • [Edit Online](#)

To complete the steps in this procedure, you must first complete the procedure, ER Create a configuration provider and mark it as active.

This procedure shows how a model's root item can be selected as a data model definition for inserting an Electronic reporting (ER) format configuration that is designed to generate electronic documents. In this procedure, you will add a new ER format configuration for the sample company Litware, Inc.

This procedure is intended for users who have the System administrator or Electronic reporting developer role assigned to them. The steps can be completed by using any dataset.

1. Go to Organization administration > Workspaces > Electronic reporting.
  - Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as Active. If you don't see this configuration provider, complete the steps in the procedure, Create a configuration provider and mark it as active.
2. Click Reporting configurations.

## Add a new ER data model configuration

1. Click Create configuration to open the drop dialog.
  - We add a new ER model configuration containing a data model that is designed to be used as data source for generation ER reports.
2. In the Name field, type 'Payment model (fictitious)'.
  - Payment model (fictitious)
3. Click Create configuration.
4. Click Designer.
  - Open the ER designer to specify the structure of data model of this configuration.
  - Assume that we design the data model for payments business domain to support 2 payment methods – credit transfer and direct debit ones.
5. Click New to open the drop dialog.
6. In the Name field, type 'Payments – credit transfer'.
  - Payments – credit transfer
7. Click Add.
8. Click New to open the drop dialog.
9. In the New node as a field, enter 'Model root'.
10. In the Name field, type 'Payments – direct debit'.
  - Payments – direct debit
11. Click Add.
12. Click Save.
13. Close the page.
14. Click Change status.
  - Complete the draft version of the model to make it available in new model mappings and formats.
15. Click Complete.

16. Click OK.

## Start to enter a new ER format configuration

1. Click Create configuration to open the drop dialog.
2. In the New field, enter 'Format based on data model Payment model (fictitious)'.
3. In the Data model definition field, enter or select a value.
  - Note that all root items of the selected data model are currently available for selection as a data model definition. You can continue to design your format by using any of the required root items of the data model. A missing model mapping for the selected root item doesn't prevent you from continuing.
4. Close the page.

## Add a new ER model mapping configuration

1. Click Create configuration to open the drop dialog.
2. In the New field, enter 'Model Mapping based on data model Payment model (fictitious)'.
3. In the Name field, type 'Payment model mappings (fictitious)'.
  - Payment model mappings (fictitious)
4. In the Data model definition field, enter or select a value.
5. Click Create configuration.

## Design ER model mappings

1. Click Designer.
  - Use the ER designer to specify the model mappings for the required root items.
2. Click Designer.
  - Simulate setting of selected model mapping for the selected model's root item.
3. In the tree, select 'Dynamics 365 for Operations\Table records'.
4. Click Add root.
5. In the Name field, type 'Ledger'.
6. In the Table field, type 'LedgerJournalTrans'.
  - LedgerJournalTrans
7. Click OK.
8. Click Save.
9. Close the page.
10. Close the page.

## Start to enter another new ER format configuration

1. In the tree, select 'Payment model (fictitious)'.
2. Click Create configuration to open the drop dialog.
3. In the New field, enter 'Format based on data model Payment model (fictitious)'.
4. In the Data model definition field, enter or select a value.
  - Note that now only one root item is available to map to the application data sources. When at least one model mapping is introduced, only the model's root items that are mapped to application data sources can be selected as a model definition while the ER format is added.
5. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# RCS enhanced filtering options for finding configurations in the RCS/Global repository

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes enhanced filtering capabilities for Regulatory Configuration Services (RCS) Global repository, which have been improved to include the ability to filter with the following criteria:

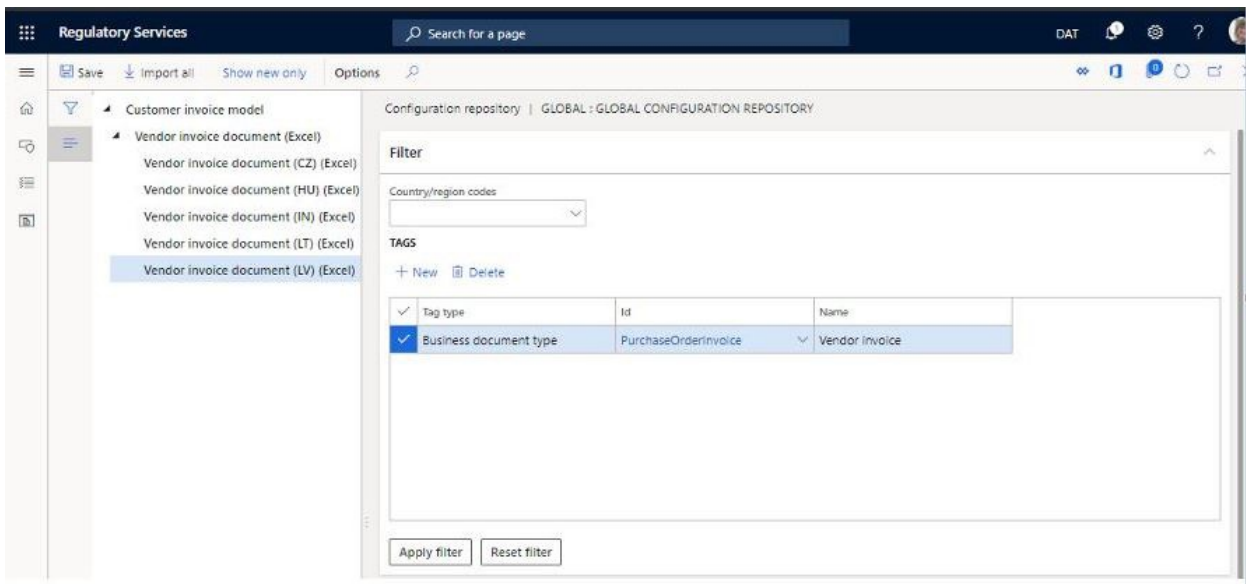
- **Country/region** - Based on ISO country codes
- **Tags** types for:
  - Functional area
  - Feature area
  - Industry
  - Business document

To make it easier to discover specific or related configurations you can apply filters, either individually or as a group. For example, to find a single type of 'configurable business documents that are related to vendor invoices, you could apply a **Business document type** filter to search for that type of document.

The screenshot shows the 'Filter' section of the 'Configuration repository' interface. It includes a 'Country/region codes' dropdown, a 'TAGS' section with '+ New' and 'Delete' buttons, and a table with columns 'Tag type', 'Id', and 'Name'. The 'Tag type' column has a dropdown menu open showing options: 'Functional area', 'Feature area', 'Industry', and 'Business document type'. The 'Functional area' option is selected. Below the table are 'Apply filter' and 'Reset filter' buttons.

Tag type	Id	Name
Functional area		

You can further refine the search by selecting document type, for example 'vendor invoice' and clicking **Apply filter**. The following example shows the results when filtering on **Business document type** with the document type added.



Filtered results can be imported into a users RCS repository or a Dynamics 365 Finance environment, either individually or as a set. To do this, select the group of configurations, and click **Import**.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Generate reports by adding content as raw XML

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can use the new **RAW XML** format element to design Electronic reporting (ER) formats that generate outgoing documents in XML format. In some cases, you might prefer to add raw XML data to these reports for one or more of the following reasons:

- It's more convenient to use raw XML for the original design and ongoing maintenance of a report, because the XML structure can be automatically generated by executing a runtime expression. Therefore, multiple bindings don't have to be determined for multiple format elements at design time. It is possible when the data sources that you're using contain information that can be used to make XML elements while the report is generated.
- No other method can be used to fill the report with XML content that was previously received and stored in the system. For example, the XML response that is generated might have to contain the content of an XML request that was sent earlier.
- No other method can be used to insert characters into the generated document based on their numeric codes. For some languages and characters, codes of this type don't exist. Examples include the Greek letter rho (ρ) and HTML entity codes such as &eacute; for an *e* that has an acute accent (é).

## NOTE

Be aware that the framework doesn't control whether the XML content that is placed to the generated document by using the **RAW XML** format element is correct.

To learn more about this feature, play the **ER Use raw XML data to generate XML reports (Part 1: Design data model)** and **ER Use raw XML data to generate XML reports (Part 2: Design and run report)** task guides, which are part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process, and can be downloaded from the [Microsoft Download Center](#). These task guides walk you through the process of configuring an ER format to insert raw XML data into generated files.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cross-company data sources in Electronic reporting (ER)

2/18/2021 • 4 minutes to read • [Edit Online](#)

You can design Electronic reporting (ER) formats to generate outgoing documents in various formats. When a document is generated, an ER format calls data sources that were configured in a corresponding ER model mapping. To configure access to application tables for record retrieval, you can use ER data sources of the **Table records** type. When the accessing table is a shared table (that is, a table where data is saved without a company identifier), this data source returns all records. When the accessing table is a company-dependent table (that is, a table where data is saved per company), this data source returns only the records that have been saved for the current company (that is, the company context that the ER format is running under).

Every data source of the **Table records** type in a model mapping can be now marked as a cross-company data source. Therefore, you can use data sources of the **Table records** type to access cross-company data in application tables.

If you mark a data source as cross-company, the following behavior occurs:

- For a data source that refers to any shared table except CompanyInfo, the data source returns all records that exist in the referenced table.
- For a data source that refers to the CompanyInfo table, even though CompanyInfo is a shared table, the data source returns the records that contain the identifier of a company from the defined scope.
- For any company-dependent table, the data source returns the records of the referenced table that contain the identifier of a company from the defined scope.

In the system query dialog box, when the **Ask for query** option is turned on for any data source that is marked as cross-company, you can manually select one or more companies to include on the **Company range** tab.

## IMPORTANT

Like other filters, the company filter is persisted as a last-used value for queries when you run an ER format. The filter isn't automatically changed if you change the cross-company value for a data source. To use a different cross-company value for another data source, delete the corresponding user-specific selection.

For every data source that is marked as cross-company, you can select the records that you want by using the **FILTER** and **WHERE** functions in ER expressions. The **dataAreaID** field can also be used as a company identifier. Currently, the **dataAreaID** field is limited to the following types of conditions when the **FILTER** function is used:

- Only conditions that have a single **dataAreaID** field comparison are supported.
- Only comparisons that have expressions that don't depend on records list items are allowed.

Therefore, the following expression is valid.

```
FILTER (MyTable, MyTable.dataAreaID = $StringUserInputParameter)
While shown below expressions will not pass the validation:
FILTER (MyTable, MyTable.dataAreaID = MyTable2RecordsList.MyField)
FILTER (MyTable,
    OR(
        MyTable.dataAreaID = $StringUserInputParameter1,
        MyTable.dataAreaID = $StringUserInputParameter2
    )
)
```

By default, the scope includes all companies of the current application. However, it can be restricted. To restrict the scope of cross-company data access for a single ER format, assign a specific organization hierarchy to the format. When a hierarchy is defined for an ER format, only records for legal entities that are presented in the assigned hierarchy are returned, even though the format calls cross-company data sources. When a reference to a hierarchy that no longer exists is defined for an ER format, the default scope is applied, and the format calls cross-company data sources. In this situation, records for all application companies are returned.

Note that when the **Use draft** option is turned on for the assigned to a single ER format organization hierarchy, the legal entities from the draft version of this hierarchy will be used to identify the scope for cross-company data sources. If the draft version does not exist, the legal entities from the last published version of this organization hierarchy will be used for this.

Note that when the **Use draft** option is turned off for the assigned to a single ER format organization hierarchy, the legal entities from the last published version of this organization hierarchy will be used to identify the scope for cross-company data sources. Date effectiveness of organization hierarchies is not supported yet in the ER framework.

The hierarchy can be assigned to a format in a specific page that can be accessed from the ER workspace or by using the **Organization administration > Electronic reporting > Legal entity filter for formats** menu item. To access the page, the **Maintain legal entity filters for format** privilege (ERMaintainFormatMappingLegalEntityFilters) must be granted to a user. The scope restriction of hierarchy-based legal entities for the format is applied in addition to the restriction that the user can manually specify in the system query dialog box. The intersection of these restrictions is used when the format is run.

To learn more about this feature, play the task guide, **ER Access records of company dependent tables in cross-company mode**, which is part of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process, and can be downloaded from the [Microsoft Download Center](#). This task guide walks you through the process of configuring an ER model mapping and ER format to access application tables in cross-company mode.

Download the following files to complete the task guide:

- [ER model configuration - CrossCompanyDataAccessModel.xml](#)
- [ER format configuration - CrossCompanyDataAccessFormat.xml](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# ER Design a configuration for generating reports in OPENXML format (November 2016)

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how a user in the System Administrator or Electronic Reporting Developer role can create a new Electronic reporting (ER) configuration that contains a template for generating electronic documents in OPENXML format. This configuration will be used for processing vendor payments.

In this example, you will create a configuration for sample company, Litware, Inc. These steps can be performed in GBSI company.

To complete these steps, you must first complete the steps in the "Create a configuration provider and mark it as active" procedure. You must also have an Excel file which will be imported when creating the template. This file can be accessed from the [Template of Payment Report](#).

## Upload the Payments data model configuration

1. In the navigation pane, go to **Modules > Organization administration > Workspaces > Electronic reporting**.
2. In the list, mark the configuration provider for sample company, Litware, Inc. If you don't see this configuration provider, you must first complete the steps in [Create configuration providers and mark them as active](#).
3. Select **Set active**.
4. Select **Repositories**. Select a repository for the Operations Resources type, if available. If its available, skip the following steps about creating a new repository.
5. Select **Add** to open the drop dialog.
6. In the **Configuration repository type** field, enter `Operations resourcesdd`.
7. Select **Create repository**.
8. Select **OK**.
9. Select **Open**.
10. In the tree, select **Payment model**.
11. Select **Import**. Import this data model. It will be used as a data source in a new format configuration. Skip this step if this configuration has been already imported.
12. Select **Yes**.
13. Close the pages until you return to the Electronic reporting page.

## Create a new format configuration

1. Select **Reporting configurations**.
2. In the tree, select **Payment model**.
3. Select **Create configuration** to open the drop dialog.
4. In the **New** field, enter `Format based on data model PaymentModel`. Create a format that is based on the PaymentModel data model.
5. In the **Name** field, type `Sample worksheet report`. Sample worksheet report
6. In the **Description** field, type `Sample worksheet report for vendors' payments`. Sample worksheet report for vendors' payments.
7. In the **Data model definition** field, enter or select a value. Select the **CustomerCreditTransferInitiation**

definition.

8. Select **Create configuration**.

## Design a new document in OPENXML worksheet format

1. In the tree, select **Payment model\Sample worksheet report**.
2. Select **Designer**.
3. On the Action Pane, select **Import**.
4. Select **Import from Excel**.
5. Select **Attachments**. Attach the existing Excel document as a template.
6. Select **New**.
7. Select **File**. Point to the existing Excel file.
8. Close the page.
9. In the **Template** field, enter or select a value. Select the attached Excel file to be used as a template.
10. Select **OK**. Note that ER format components have been created in the designing format based on the structure of the referring MS Excel document (named ranges).

## Create a new data source to calculate totals by currency codes

1. Select the **Mapping** tab.
2. Select **Add root** to open the drop dialog.
3. In the tree, select **Functions\Group by**.
4. In the **Name** field, type `PaymentByCurrency`.
5. Select **Edit group by**.
6. In the tree, expand **model**, then select **model\Payments**.
7. Select **Add field to**.
8. Select **What to group**.
9. In the tree, expand **model\Payments**, then select **model\Payments\Currency**.
10. Select **Add field to**.
11. Select **Grouped fields**.
12. In the tree, select **model\Payments\Instructed Amount(InstructedAmount)**.
13. Select **Add field to**, then select **Aggregation fields**.
14. In the **Method** field, select an option. Select the **SUM aggregation** function.
15. In the **Name** field, type `TotalInstructedAmount`.
16. Select **Save**.
17. Close the page.
18. Select **OK**.

## Map format components to data sources

1. In the tree, select **model\Payments\Initiating Party(InitiatingParty)\Name** and **Excel\ReportHeader\CompanyName**.
2. Select **Bind**.
3. In the tree, select **model\Payments\Creditor\Identification\Source ID(SourceID)** and **Excel\PaymLines\VendAccountName**.
4. Select **Bind**.
5. In the tree, select **model\Payments\Creditor\Name** and **Excel\PaymLines\VendName**.
6. Select **Bind**.

7. In the tree, select `model\Payments\Creditor Agent(CreditorAgent)\Name` and `Excel\PaymLines\Bank`.
8. Select **Bind**.
9. In the tree, select `model\Payments\Creditor Agent(CreditorAgent)\Routing Number(RoutingNumber)` and `Excel\PaymLines\RoutingNumber`.
10. Select **Bind**.
11. In the tree, select `model\Payments\Creditor Account(CreditorAccount)\Identification\Number` and `Excel\PaymLines\AccountNumber`.
12. Select **Bind**.
13. In the tree, select `model\Payments\Instructed Amount(InstructedAmount)` and `Excel\PaymLines\Debit`.
14. Select **Bind**.
15. In the tree, select `model\Payments\Currency` and `Excel\PaymLines\Currency`.
16. Select **Bind**.
17. In the tree, select `PaymentByCurrency\grouped\Currency` and `Excel\SummaryLines\SummaryCurrency`.
18. Select **Bind**.
19. In the tree, select `PaymentByCurrency\aggregated\TotalInstructedAmount` and `Excel\SummaryLines\SummaryAmount`.
20. Select **Bind**.
21. In the tree, select `PaymentByCurrency` and `Excel\SummaryLines`.
22. Select **Bind**.
23. In the tree, select `model\Payments` and `Excel\PaymLines`.
24. Select **Bind**.
25. Select **Save**, then close the page.

## Use the created configuration for payments processing

1. Select **Change status**.
2. Select **Complete**.
3. Select **OK**.
4. In the navigation pane, go to **Modules > Accounts payable > Payment setup > Methods of payment**.
5. Use the Quick Filter to filter on the **Method of payment** field with a value of **Electronic**.
6. Select **Edit**.
7. Expand the **File formats** section.
8. Select **Yes** in the **Generic electronic reporting** field.
9. In the **Export format configuration** field, enter or select a value. Select the **Sample worksheet report** configuration.
10. Select **Save**.
11. Close the page.

## Use the created configuration for testing of payment journals processing

1. In the navigation pane, go to **Modules > Accounts payable > Payments > Payment journal**.
2. Select **New** to create a new payment journal.
3. In the **Name** field, type **VendPay**.
4. Select **Lines**.

5. In the **Account** field, specify the values `GB_SI_000001`.
6. Set **Debit** to `1000`.
7. Select **New**.
8. In the **Account** field, specify the values `GB_SI_000005`.
9. Set **Debit** to `2000`.
10. In the **Currency** field, type `EUR`.
11. In the **Offset account** field, specify the values `GBSI OPER`.
12. In the **Method of payment** field, type `Electronic`.
13. Select **Save**.
14. Select **Generate payments**.
15. In the **Method of payment** field, type `Electronic`.
16. In the **File name** field, type `Payments`.
17. In the **Bank account** field, type `GBSI OPER`.
18. Select **OK**, then select **OK** again. Review the created worksheet, including details of payment lines as well as totals for each currency code used in this payment message.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Reuse ER configurations with Excel templates to generate reports in Word format

2/18/2021 • 6 minutes to read • [Edit Online](#)

To generate reports as Microsoft Word documents, you can [configure](#) a new [Electronic reporting \(ER\) format](#). Alternatively, you can reuse an ER format that was originally designed to generate reports as Excel workbooks. In this case, you must replace the Excel template with a Word template.

The following procedures show how a user in either the System administrator role or the Electronic reporting developer role can configure an ER format to generate reports as Word files by reusing an ER format that was designed to generate reports as Excel files.

These procedures can be completed in the GBSI company.

## Prerequisites

To complete these procedures, you must first follow the steps in the [Design a configuration for generating reports in OPENXML format](#) task guide.

You must also download and locally save the following templates for the sample report:

- [Template of Payment Report \(SampleVendPaymDocReport.docx\)](#)
- [Bounded Template of Payment Report \(SampleVendPaymDocReportBounded.docx\)](#)

These procedures are for a feature that was added in Dynamics 365 for Operations version 1611 (November 2016).

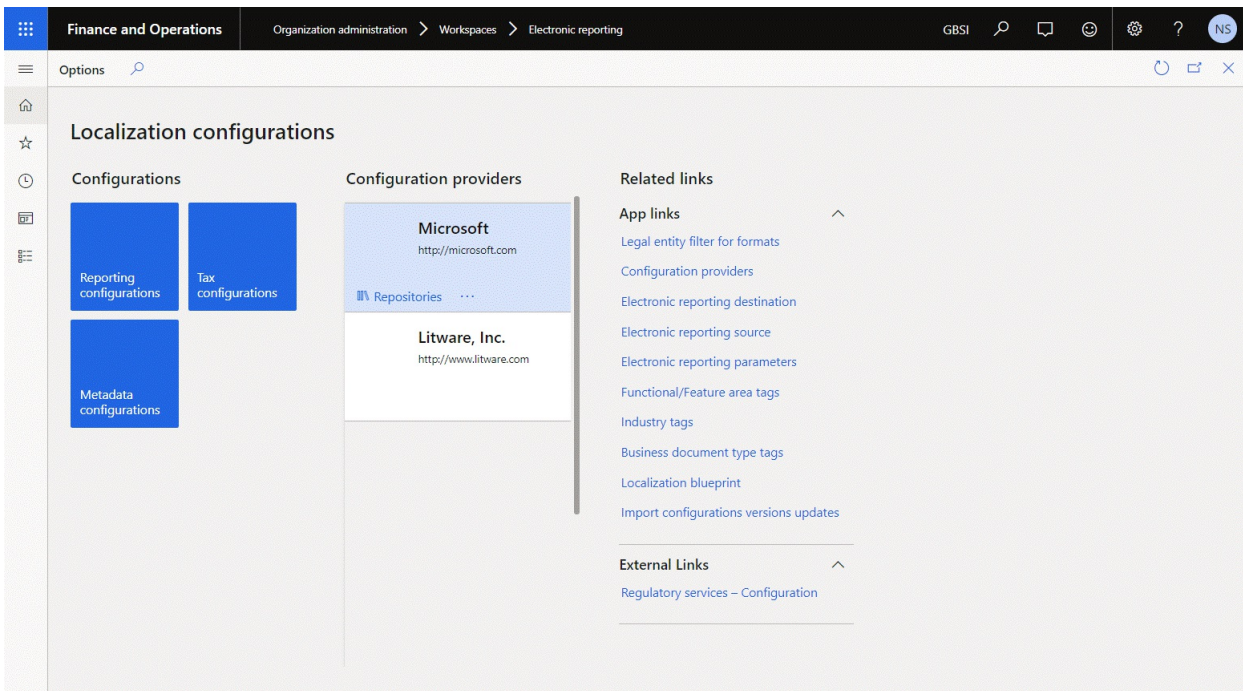
## Select the existing ER report configuration

1. In Dynamics 365 Finance, go to **Organization administration** > **Workspaces** > **Electronic reporting**.
2. Make sure that the **Litware, Inc.** configuration provider is selected as **Active**. If it isn't, follow the steps in the [Create configuration providers and mark them as active](#) task guide.
3. Select **Reporting configurations**. You will reuse the existing ER configuration that was designed to generate the report output in OPENXML format.
4. On the **Configurations** page, in the configuration tree in the left pane, expand **Payment model**, and select **Sample worksheet report**.

### NOTE

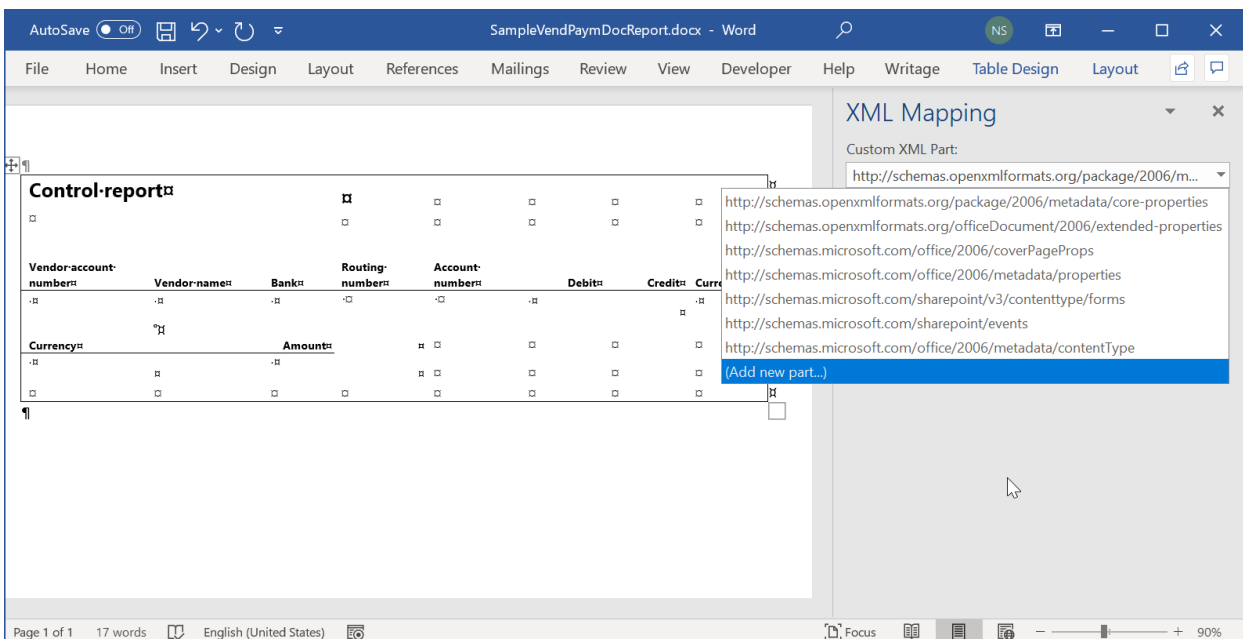
The draft version of the selected ER format can be edited on the **Versions** FastTab.

5. Select **Designer**.
6. On the **Format designer** page, notice that the title of the root format element indicates that an Excel template is currently used.



## Review the downloaded Word template

1. In the Word desktop application, open the **SampleVendPaymDocReport.docx** template file that you downloaded earlier.
2. Verify that the template contains only the layout of the document that you want to generate as ER output.



## Replace the Excel template with the Word template and add a custom XML part

Currently, the Excel document is used as a template to generate the output in OPENXML format. You will replace this template with the **SampleVendPaymDocReport.docx** Word template file that you downloaded earlier. You will also extend the Word template by adding a custom XML part.

1. In Finance, on the **Format designer** page, on the **Format** tab, select **Attachments**.
2. On the **Attachments** page, select **Delete** to remove the existing Excel template. Select **Yes** to confirm the change.

3. Select **New > File**.

**NOTE**

You must select a document type that has been [configured](#) in the ER parameters to store templates of ER formats.

4. Select **Browse**, and then browse to and select the **SampleVendPaymDocReport.docx** file that you downloaded earlier.

5. Select **OK**.

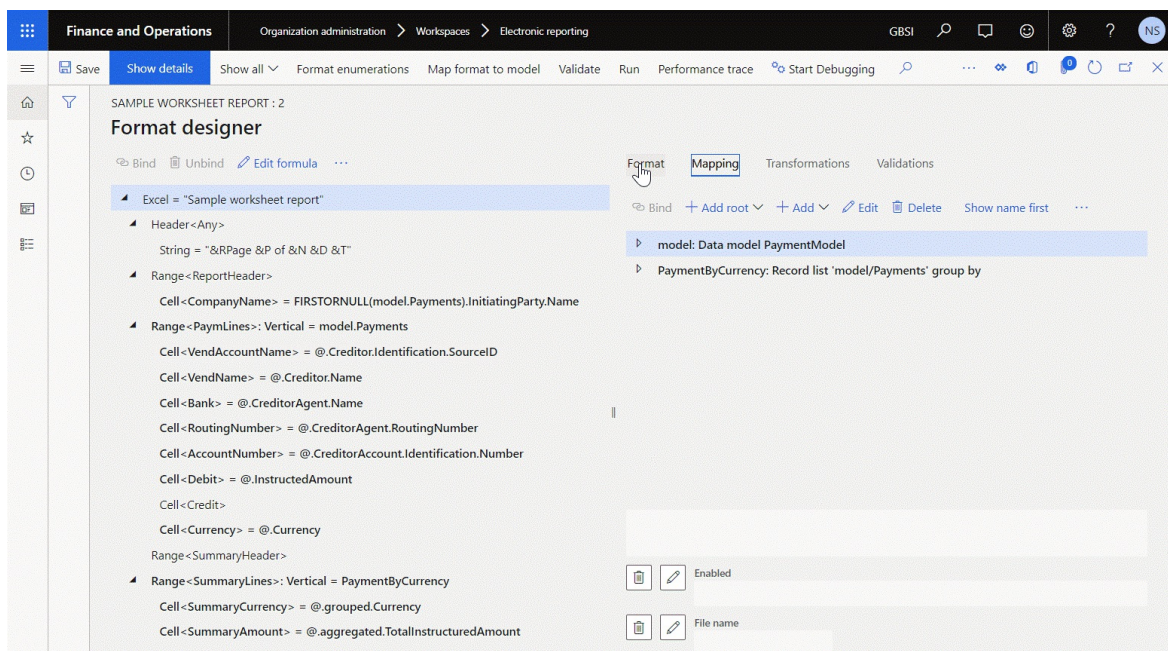
6. Close the **Attachments** page.

7. On the **Format designer** page, in the **Template** field, enter or select the **SampleVendPaymDocReport.docx** file to use that Word template instead of the Excel template that was previously used.

8. Select **Save**.

In addition to storing configuration changes, the **Save** action updates the attached Word template. The hierarchical structure of the designed format is added to the attached Word document as a new custom XML part that is named **Report**. The attached Word template contains the layout of the document that will be generated as ER output and the structure of data that ER will enter in that template at runtime.

9. Notice that the title of the root format element indicates that a Word template is currently used.



10. On the **Format** tab, select **Attachments**.

You can now map the elements of the **Report** custom XML part to the content controls of the Word document.

If you're familiar with the process of designing Word documents as forms that contain [content controls](#) that are mapped to elements of [custom XML parts](#), complete all steps in the next procedure to create the document. For more information, see [Create forms that users complete or print in Words](#). Otherwise, skip the next procedure.

## Get a Word document that has a custom XML part and do data mapping

1. In Finance, on the **Attachments** page, select **Open** to download the selected template from Finance and store it locally as a Word document.

2. In the Word desktop application, open the document that you just downloaded.
3. On the **Developer** tab, select **XML Mapping Pane**.

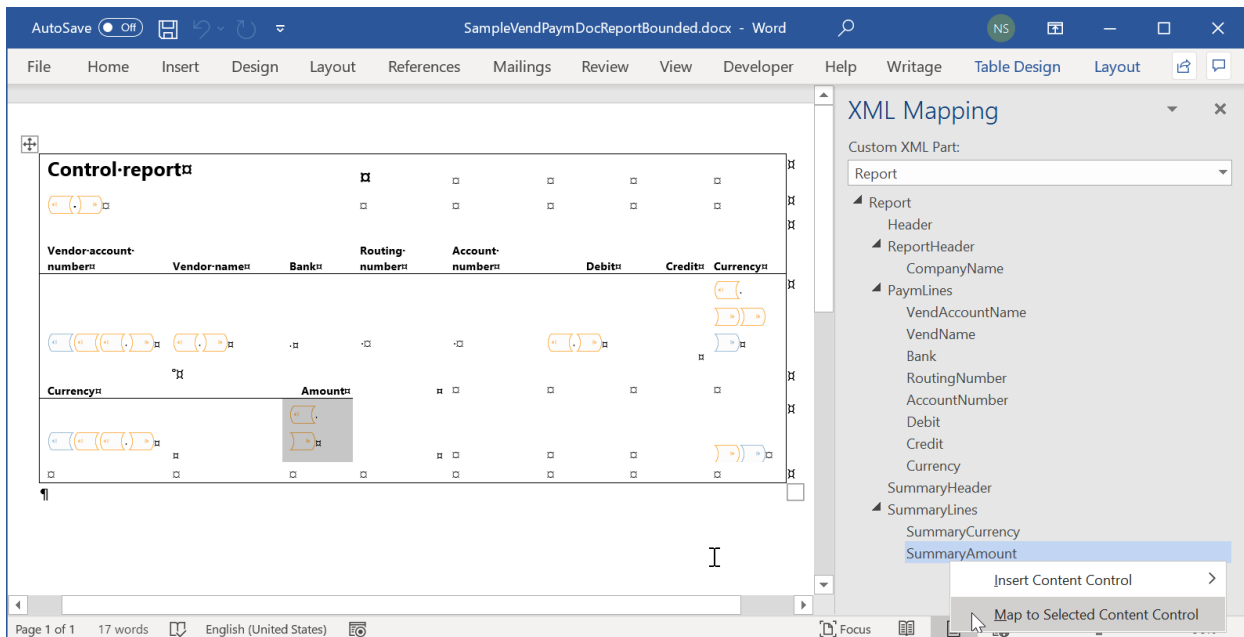
#### NOTE

If the **Developer** tab doesn't appear on the ribbon, customize the ribbon to add it.

4. In the **XML Mapping** pane, in the **Custom XML Part** field, select the **Report** custom XML part.
5. Map the elements of the selected **Report** custom XML part and the content controls of the Word document.
6. Save the updated Word document locally as **SampleVendPaymDocReportBounded.docx**.

## Review the Word template where the custom XML part is mapped to content controls

1. In the Word desktop application, open the **SampleVendPaymDocReportBounded.docx** template file.
2. Verify that the template contains the layout of the document that you want to generate as ER output. The content controls that are used as placeholders for data that ER will enter in this template at runtime are based on the mappings that are configured between elements of the **Report** custom XML part and the content controls of the Word document.



## Upload the Word template where the custom XML part is mapped to content controls

1. In Finance, on the **Attachments** page, select **Delete** to remove the Word template that has no mappings between elements of the **Report** custom XML part and content controls. Select **Yes** to confirm the change.
2. Select **New > File** to add a new template file that contains mappings between elements of the **Report** custom XML part and content controls.



## NOTE

You must select a document type that has been [configured](#) in the ER parameters to store templates of ER formats.

3. Select **Browse**, and then browse to and select the **SampleVendPaymDocReportBounded.docx** file that you downloaded or prepared by completing the procedure in the [Get a Word that has a custom XML part to do data mapping](#) section.
4. Select **OK**.
5. Close the **Attachments** page.
6. On the **Format designer** page, in the **Template** field, select the document that you just downloaded.
7. Select **Save**.
8. Close the **Format designer** page.

## Mark the configured format as runnable

To run the draft version of the editable format, you must make it [runnable](#).

1. In Finance, on the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
2. In the **User parameters** dialog box, set the **Run settings** option to **Yes**, and then select **OK**.
3. Select **Edit** to make the current page editable, as required.
4. For the currently selected **Sample worksheet report** configuration, set the **Run Draft** option to **Yes**.
5. Select **Save**.

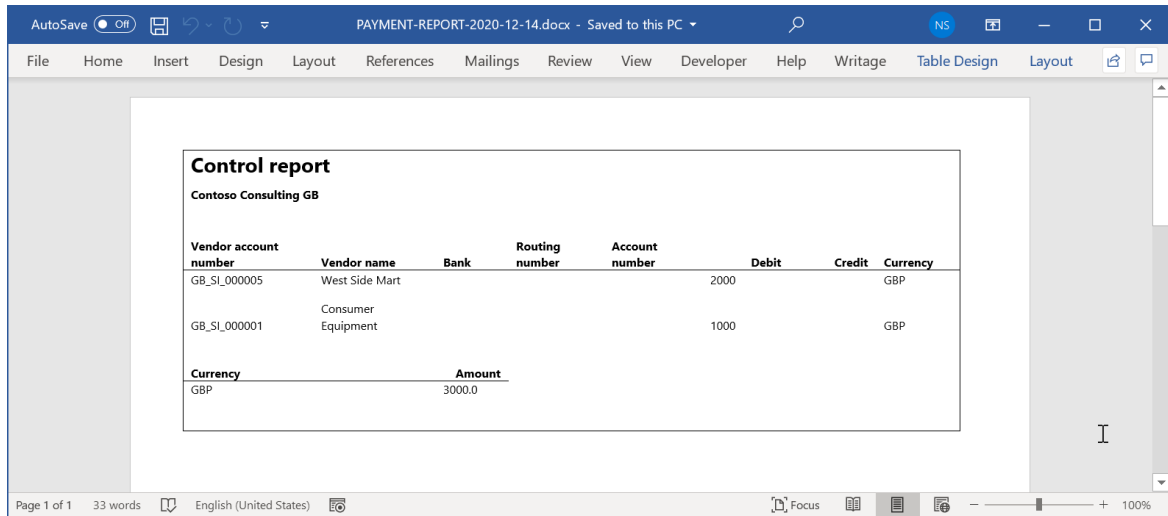
## Run the format to create output in Word format

1. In Finance, go to **Accounts payable > Payments > Payment journal**.
2. In a payment journal that you entered earlier, select **Lines**.
3. On the **Vendor payments** page, select all rows in the grid.
4. Select **Payment status > None**.

	DEBIT	CREDIT	BALANCE	DEBIT	CREDIT	BALANCE
VOUCHER	2,000.00	2,000.00	0.00	3,151.10	3,151.10	0.00
JOURNAL	3,000.00	3,000.00	0.00	4,726.60	4,726.60	0.00

5. On the Action Pane, select **Generate payments**.
6. In the dialog box that appears, follow these steps:
  - a. In the **Method of payment** field, select **Electronic**.

- b. In the **Bank account** field, select **GBSI OPER**.
  - c. Select **OK**.
7. In the **Electronic report parameters** dialog box, select **OK**.
  8. The generated output is presented in Word format and contains the details of the processed payments. Analyze the generated output.



**Control report**

Contoso Consulting GB

Vendor account number	Vendor name	Bank	Routing number	Account number	Debit	Credit	Currency
GB_SI_000005	West Side Mart				2000		GBP
GB_SI_000001	Consumer Equipment				1000		GBP

Currency	Amount
GBP	3000.0

## Additional resources

- [Design a new ER configuration to generate reports in Word format](#)
- [Embed images and shapes in documents that you generate by using ER](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Design ER configurations to fill in PDF templates

2/18/2021 • 14 minutes to read • [Edit Online](#)

The procedures in this topic are examples that show how a user in either the **System administrator** role or the **Electronic reporting developer** role can configure an Electronic reporting (ER) format that generates reports as PDF files by using fillable PDF documents as report templates. These steps can be performed in any company of Dynamics 365 Finance or Regulatory Configuration Services (RCS).

## Prerequisites

Before you begin, you must have one of the following types of access, depending on the service that you use to complete the procedures in this topic:

- Access to Finance for one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
- Access to RCS for one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator

You must also complete the [Create configuration providers and mark them as active](#) procedure.

Finally, you must download the following files from [CustomerSource](#).

CONTENT DESCRIPTION	FILE NAME
Template for the first page of the report	<a href="#">IntrastatReportTemplate1.pdf</a>
Template for other pages of the report	<a href="#">IntrastatReportTemplate2.pdf</a>
Sample ER format - PDF	<a href="#">Intrastat report (PDF).version.1.1.xml</a>
Sample ER format - Excel	<a href="#">Intrastat (import from Excel).version.1.1.xml</a>
Sample dataset	<a href="#">Intrastat sample data.xlsx</a>

## Design the format configuration

**Get access to the list of configurations provided by Microsoft**

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Make sure that the **Litware, Inc.** provider is available and marked as active.
3. On the tile for the **Microsoft** provider, select **Repositories**.

**NOTE**

If a repository of the LCS type already exists, skip the remaining steps of this procedure.

4. Select **Add**.
5. In the drop-down dialog box, in the **Configuration repository type** field group, select the **LCS** option.
6. Select **Create repository**.
7. Select **OK**.

**Get the model configurations provided by Microsoft**

1. On the left side of the **Configuration repositories** page, select the **Show filters** button (the funnel symbol).
2. Add a filter for a value of **LCS** in the **Type** field, and use the **begins with** operator.
3. Select **Apply**.
4. Select **Open**.
5. In the tree, select **Intrastat model**.
6. On the **Versions** FastTab, select version **1**.

**NOTE**

If the **Import** button on the **Versions** FastTab is unavailable, skip the remaining steps of this procedure.

7. Select **Import**.
8. Select **Yes** to confirm the import of the selected version of the **Intrastat model** model configuration.

**Create a new format configuration**

1. In the **Electronic reporting** workspace, select the **Reporting configurations** tile.
2. In the tree, select **Intrastat model**.
3. Select **Create configuration**.

**NOTE**

If the **Create configuration** button isn't available, you must turn on design mode on the **Electronic reporting parameters** page that can be opened from the **Electronic reporting** workspace.

4. In the drop-down dialog box, in the **New** field group, select the **Format based on data model Intrastat** option.
5. In the **Name** field, enter **Intrastat report (PDF)**.
6. In the **Description** field, enter **Intrastat report in PDF format**.

**NOTE**

The active configuration provider is automatically entered. This provider will be able to maintain this configuration. Although other providers can use this configuration, they won't be able to maintain it.

7. Optional: In the **Format type** field, you can select a specific format of electronic document. If you select **PDF**, at design time, the ER Operations designer will offer just the format elements that are applicable only to documents that are generated in PDF format (**PDF\File**, **PDF\PDF Merger**, etc.). If you leave this field blank, a format of electronic document will be specified at design time in the ER Operations designer when a first format element will be added. For example, if you add the **Excel\File** as the first format element, the ER Operations designer will offer you just the format elements that are applicable only to documents that are generated in Excel format (**Excel\Cell**, **Excel\Range**, etc.). format.

8. Select **Create configuration**.

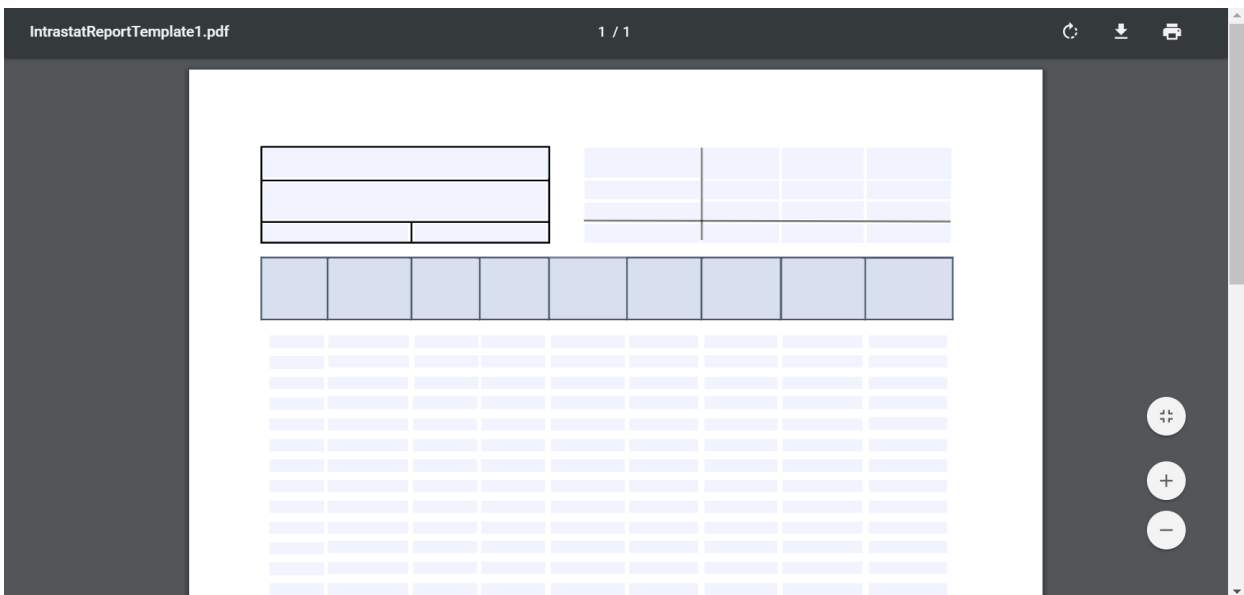
A new ER format configuration is created. You can use the draft version of this configuration to store the ER format component that is designed to generate electronic documents in PDF format.

### Design the format of an electronic document

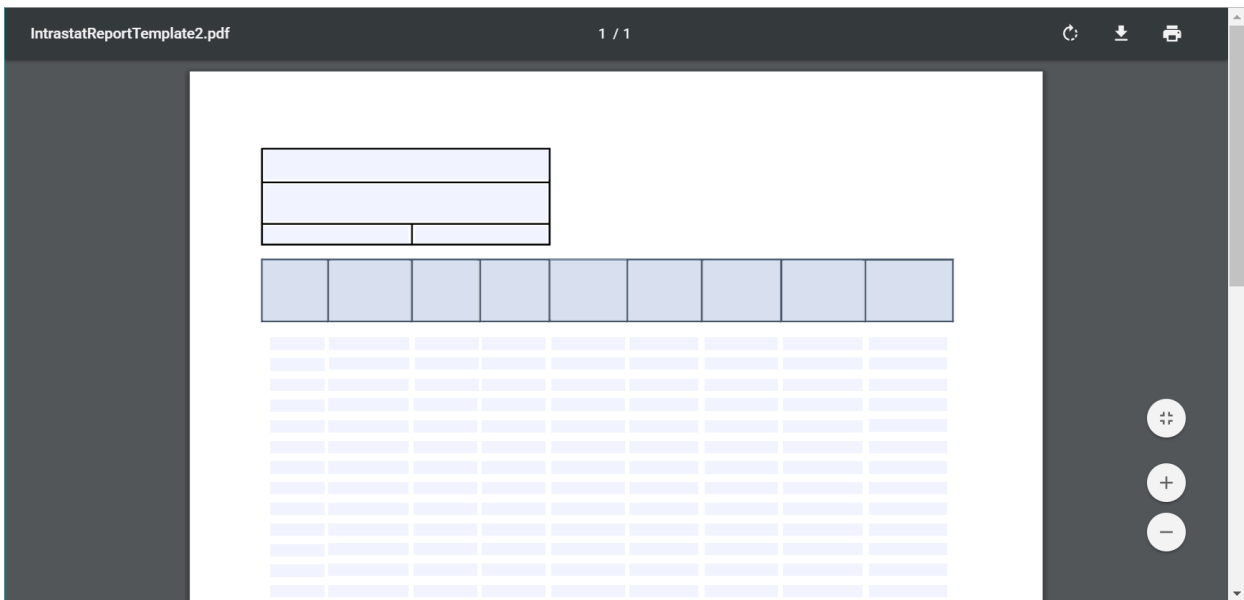
Next, in the ER format configuration that you created, you will design the ER format that generates the **Intrastat control** report in PDF format. The first page of this report must show a summary of the report and details of the foreign trade transactions that are reported on. The other pages must show only details of the foreign trade transactions that are reported on. Because the report pages that are generated must have different layouts, two different templates in PDF format will be used in the ER format.

In any PDF viewer, open the PDF templates that you downloaded. Notice that each template contains multiple fields that must be filled in. In each template, details of foreign trade transactions are presented as 42 rows, each of which has nine fields. The row number appears at the end of each field's name (for example, **Date 1...Date 42** and **Commodity 1...Commodity 42**).

The following illustration shows the PDF template for the first page of the report.



The following illustration shows the PDF template for other pages of the report.



1. On the **Configurations** page, select **Designer**.
2. Select **Add root**.
3. In the drop-down dialog box, in the tree, select **PDF > PDF Merger**.

When you select the **PDF Merger** element as the root element of the format, all PDF documents that are generated at run time will be merged into a single final PDF document. If you need only one PDF template to generate all the required documents by using the ER format that you design, you can select **PDF file** as the root element.

4. In the **Name** field, enter **Output**.
5. In the **Language preferences** field, select **User preference**. The report will be generated in the preferred language of the user who runs it.
6. In the **Culture preferences** field, select **User preference**. Values and dates that are presented on the pages of the report will be formatted based on the preferred locale of the user who runs the report.
7. Select **OK**.
8. On the Action Pane, on the **Import** tab, select **Import from PDF**.

When a fillable PDF document is imported as a template for this ER format, all the required ER format elements (**PDF file**, **Field group**, and **Field** elements) are automatically created in the format that is designed, based on the structure of the PDF document that is imported.

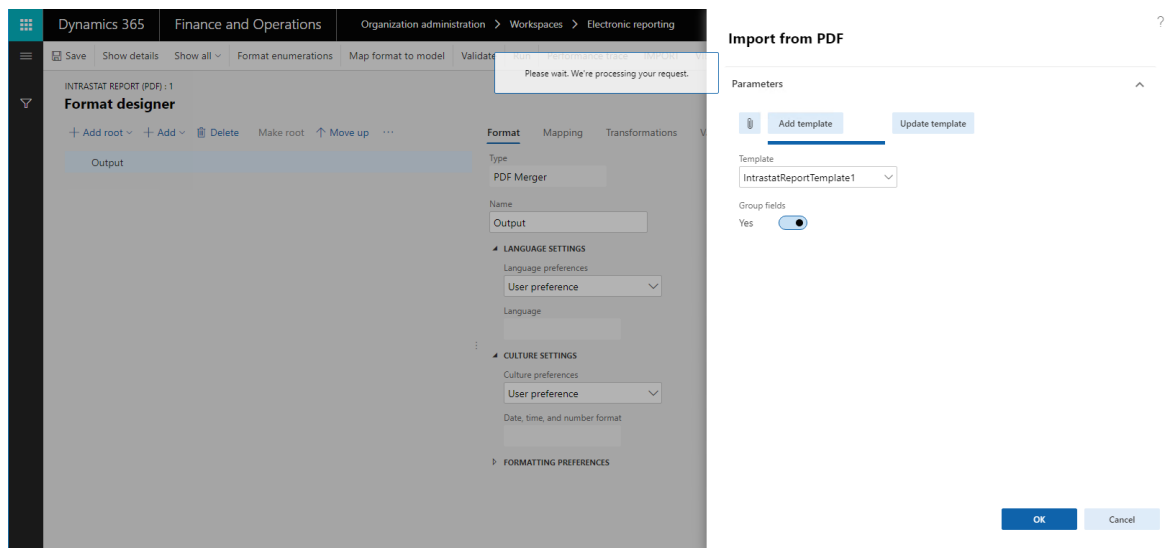
9. Select **Browse**. Navigate to and select the **IntrastatReportTemplate1.pdf** file that you downloaded earlier as a prerequisite.
10. Select **OK**.

The selected file is loaded, and the **Template** field in the **Import from PDF** dialog box is filled in.

11. Set the **Group fields** option to **Yes**. If the selected PDF document contains any field groups, they will be used to group the ER format elements that are created. A **Field group** format element will be created for this purpose.

If this option is set to **No**, the required ER format elements will be created as a flat list of elements that are nested under the **PDF File** format element that is created.

12. Select **OK**.



- In the tree, expand **Output**.

Notice that the **PDF File** component has been automatically created to manage the creation of the first page of the report that is generated at run time.

- In the tree, expand **Output > PDF File**.

Notice that the structured list of format elements has been automatically created in this ER format, based on the structure of the fillable PDF document that you imported earlier.

- In the tree, select **Output > PDF File**.

- In the **Name** field, enter **Page 1**.

This format element will be used to generate the first page of the **Intrastat control** report. That page will show a summary of the report and details of foreign trade transactions.

If you leave the **Language preferences** field blank, the **Language preferences** setting of the parent **PDF Merger** element will determine the language of the report that is generated by using this format element. You can select another value to override the setting of the parent element.

If you leave the **Culture preferences** field blank, the **Culture preferences** setting of the parent **PDF Merger** element will determine the locale of the report that is generated by using this format element. The locale determines the format of values and dates on the pages of the report. You can select another value to override the setting of the parent element.

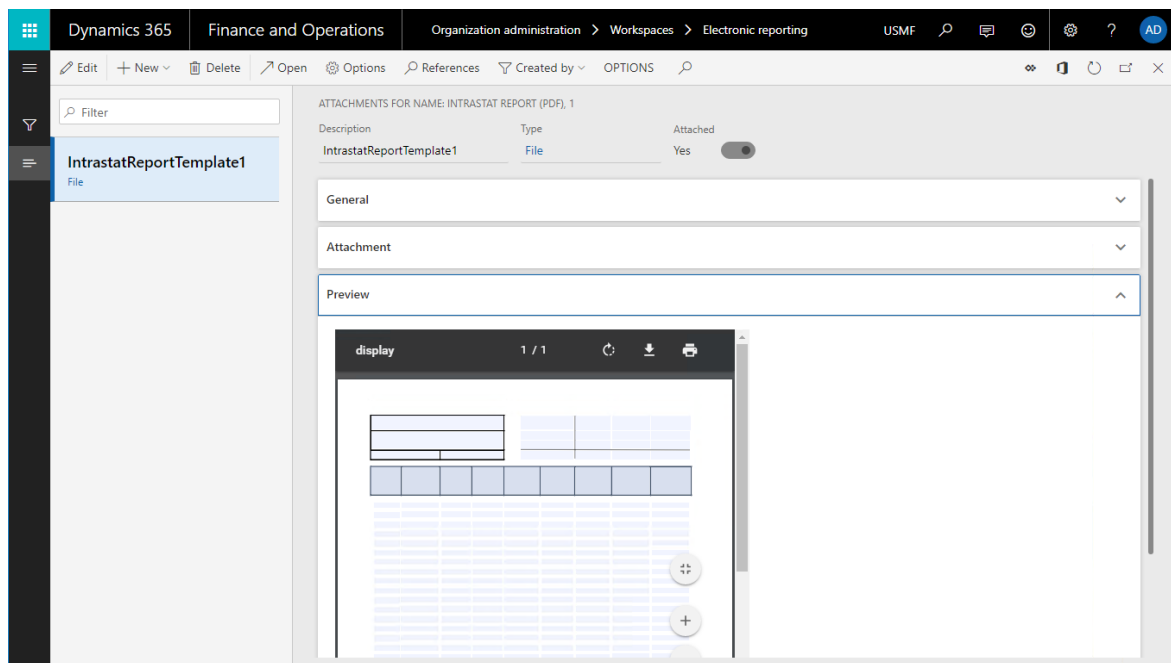
- On the Action Pane, select the **Import** tab. Notice that the **Update from PDF** button has become available for selected format element, **PDF File**.

You can use this button to import the updated PDF template to the edited format. When the updated PDF template is imported, the list of format elements will be changed accordingly:

- For any new fields in the updated PDF template, new format elements are created in the edited ER format.
- If the updated PDF template no longer includes fields that correspond to any existing format elements in the edited ER format, those format elements are deleted from the ER format.

- On the **Format** tab, select **Attachments**.

Notice that the imported PDF document is attached to the edited ER format.



19. Continue to design this format by importing the second PDF template, adding necessary bindings to data sources, and so on.
20. Select **Save**.
21. Close the page.
22. Select **Delete**.
23. Select **Yes**.

To learn how new **PDF Merger**, **PDF File**, **Field group** and **Field** format elements can be used to generate documents in PDF format, you can import and analyze the sample ER format.

### Import the format configuration

Next, you will import the sample ER format that you previously downloaded to generate the **Intrastat control** report in PDF format. The first page of the report must show a summary of the report and details of the foreign trade transactions that are reported on. The other pages must show only details of the foreign trade transactions that are reported on.

1. On the **Configurations** page, select **Exchange > Load from XML file**.
2. Select **Browse**. Navigate to and select the **Intrastat report (PDF).version.1.1.xml** file that you downloaded earlier as a prerequisite.
3. Select **OK**.

## Analyze the format configuration

### Format layout

1. On the **Configurations** page, in the tree, select **Intrastat model > Intrastat report (PDF)**.
2. Select **Designer**.
3. Select **Show details**.
4. In the tree, expand **Output: PDF Merger**.

Notice that this ER format contains two **PDF File** elements, each of which uses a different PDF template. One template is used to generate the first page of the report in PDF format, and the other template is used to generate the other pages.



5. In the tree, expand **Output: PDF Merger > Page 1: PDF File (IntrastatReportTemplate1)**.

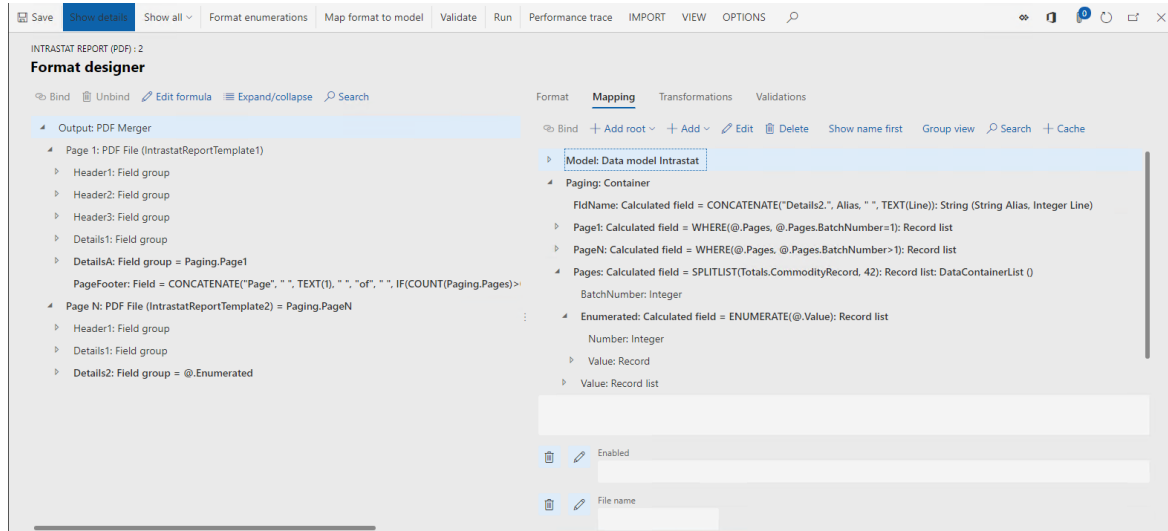
6. In the tree, expand **Output: PDF Merger > Page N: PDF File (IntrastatReportTemplate2)**.

Notice that, because the content of the two PDF templates differs, the structure of the nested format elements for the two **PDF File** elements also differs.

## Format mapping

1. On the **Format designer** page, select the **Mapping** tab.

2. In the tree, expand **Paging > Pages**.



Note the following details:

- The **Output > Page 1** format element of the **PDF File** type isn't bound to any data source, and the **Enabled** expression of this format element is empty. Therefore, at run time, the **IntrastatReportTemplate1** PDF template will be filled in only one time when an individual PDF document is generated.
- The **Output > Page N** format element of the **PDF File** type is bound to the **Paging.PageN** data source of the **Record list** type, and the **Enabled** expression of this format element is empty. Therefore, at run time, the **IntrastatReportTemplate2** PDF template will be filled in for each record from the bound record list when an individual PDF document is generated.
- Because the **Page 1: PDF File** and **Page N: PDF File** format elements are children of the **Output: PDF Merger** format element, all PDF documents that are filled in will be merged into a single final PDF document.
- The **Paging.Page1** and **Paging.PageN** data sources are configured as filters of records from the **Paging.Pages** data source. This data source is configured to split the whole set of foreign trade transactions into batches. Each batch contains up to 42 records. The following ER expression is used to split the transactions into batches:  
  
SPLITLIST(Totals.CommodityRecord,42)
- The **Paging.Pages** data source contains the **Paging.Pages.Enumerated** element that returns the details of each record that is included in a batch. These details include the record's sequence in the current batch (the **Paging.Pages.Enumerated.Number** field). The **Paging.Pages.Enumerated.Number** field is used in the **Name** expression of **PDF Field** format elements to dynamically generate a field name that is based on the transaction number in a batch. The field name that is generated is then used to fill in the correct PDF field in the PDF template that is used.

- The **Output > Page N > Details 2** format element of the **PDF Group** type is bound to the **Paging.PageN.Enumerated** data source (or **@.Enumerated** if the **Relative path** view mode is used) of the **Record list** type. Therefore, at run time, the nested elements of this PDF group will be filled in for each record from the bound record list. In this way, individual PDF lines are virtually generated when for each Nth of 42 records of the **Paging.PageN.Enumerated** list the following PDF fields are filled in: **Date N**, **Direction N**, **Commodity N**, etc. Therefore, in this respect, the behavior of this **Field group** format element resembles the behavior of the **XML > Sequence** and **Text > Sequence** format elements.

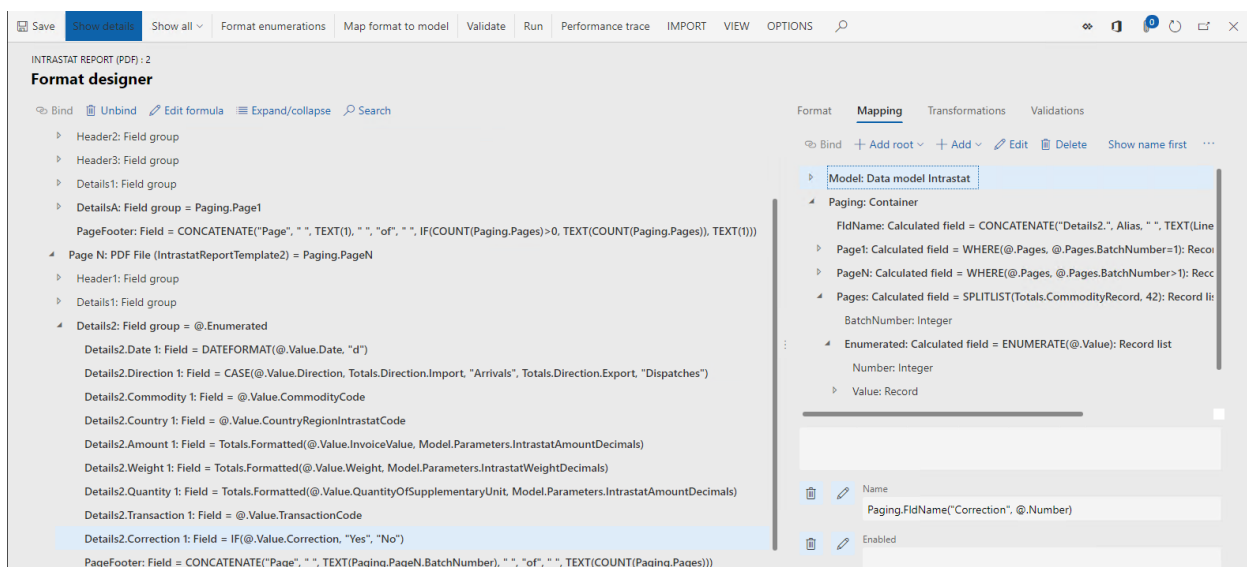
3. In the tree, expand **Output > Page N > Details2**.

4. In the tree, select **Output > Page N > Details2 > PageFooter**.

Notice that the **Name** attribute of this format element is defined as **PageFooter**. Also notice that the **Name** expression of the format element is empty.

5. In the tree, select **Output > Page N > Details2 > Correction 1**.

Notice that the **Name** attribute of this format element is defined as **Correction 1**. Also notice that the **Name** expression of the format element is defined as **Paging.FldName("Correction",@.Number)**.



Note that the **Field** format element is used to fill in an individual field of a fillable PDF document that is defined as a template of the parent **PDF File** format element. The binding of the **PDF File** format element or its nested elements, if it has any nested elements, specifies the value that is entered in corresponding PDF fields. Different properties of the **Field** format element can be used to specify which PDF field is filled in by an individual format element:

- On the **Format** tab, the **Name** attribute of the format element
- On the **Mapping** tab, the **Name** expression of the format element

Because both properties are optional for a **Field** format element, the following rules are applied to specify the target PDF field:

- If the **Name** attribute is blank, and the **Name** expression returns an empty string at run time, an exception is thrown at run time to notify the user that a PDF field can't be found in the PDF template that is being used to fill in the PDF document.
- If the **Name** attribute is defined, and the **Name** expression is blank, the PDF field that has the same name as the **Name** attribute of the format element is filled in.
- If the **Name** attribute is defined, and the **Name** expression is configured, the PDF field that has the same name as the value that is returned by the **Name** expression of the format element is filled in.

#### NOTE

A PDF check box can be filled in as selected in the following ways:

- When the corresponding **Field** format element is bound to a data source field of the **Boolean** data type that has the **True** value
- When the corresponding **Field** format element contains a nested **String** format element that is bound to a data source field that has a text value of **1**, **True**, or **Yes**

## Run the format configuration

### Import the format configuration

Next, you will load the **Intrastat (import from Excel)** sample ER format. This format is designed to parse a user-selected Microsoft Excel workbook that simulates foreign trade transactions.

1. On the **Configurations** page, select **Exchange > Load from XML file**.
2. Select **Browse**. Navigate to and select the **Intrastat (import from Excel).version.1.1.xml** file that you downloaded earlier as a prerequisite.
3. Select **OK**.
4. In the tree, select **Intrastat model > Intrastat (import from Excel)**.
5. Select **Edit**.
6. Set the **Default for model mapping** option to **Yes**.

#### NOTE

If you previously set the **Default for model mapping** option to **Yes** for the **Intrastat model** configuration or another configuration that is nested under the **Intrastat model** configuration, set this option to **No**.

When the **Default for model mapping** option is set to **Yes**, the imported **Intrastat (import from Excel)** ER format is assigned as the default data source for the **Intrastat report (PDF)** format configuration. Then, when the **Intrastat report (PDF)** format configuration is run, the content of the Excel workbook that is parsed by the **Intrastat (import from Excel)** ER format will simulate foreign trade transactions that must be reported. The following illustration shows an example of an Excel workbook.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	9202001	FALSE	1/31/2018	Import	IT	DE	100.11	GRM		1 Speaker	100.00	100.00	11	2		300
2	9202002	FALSE	2/1/2018	Import	FR	AT	102.34	GRM		2 Monitor	102.00	102.00	12	3		299
3	9202003	FALSE	2/2/2018	Import	ES	AT	104.57	GRM		3 Headset	105.00	105.00	11	4		298
4	9202004	FALSE	2/3/2018	Export	AT	IT	106.80	GRM		4 Notebook	107.00	107.00	23	5		297
5	9202005	FALSE	2/4/2018	Import	IT	DE	109.03	GRM		5 Speaker	109.00	109.00	11	2		296
6	9202006	FALSE	2/5/2018	Import	FR	AT	111.26	GRM		6 Monitor	111.00	111.00	12	3		295
7	9202007	FALSE	2/6/2018	Import	ES	AT	113.49	GRM		7 Headset	113.00	113.00	11	4		294
8	9202008	FALSE	2/7/2018	Export	AT	IT	115.72	GRM		8 Notebook	116.00	116.00	23	5		293
9	9202009	FALSE	2/8/2018	Import	IT	DE	117.95	GRM		9 Speaker	118.00	118.00	11	2		292
10	9202010	FALSE	2/9/2018	Import	FR	AT	120.18	GRM		10 Monitor	120.00	120.00	12	3		291
11	9202011	FALSE	2/10/2018	Import	ES	AT	122.41	GRM		11 Headset	122.00	122.00	11	4		290
12	9202012	FALSE	2/11/2018	Export	AT	IT	124.64	GRM		12 Notebook	125.00	125.00	23	5		289
13	9202013	FALSE	2/12/2018	Import	IT	DE	126.87	GRM		13 Speaker	127.00	127.00	11	2		288
14	9202014	FALSE	2/13/2018	Import	FR	AT	129.10	GRM		14 Monitor	129.00	129.00	12	3		287
15	9202015	FALSE	2/14/2018	Import	ES	AT	131.33	GRM		15 Headset	131.00	131.00	11	4		286
16	9202016	FALSE	2/15/2018	Export	AT	IT	133.56	GRM		16 Notebook	134.00	134.00	23	5		285
17	9202017	FALSE	2/16/2018	Import	IT	DE	135.79	GRM		17 Speaker	136.00	136.00	11	2		284
18	9202018	FALSE	2/17/2018	Import	FR	AT	138.02	GRM		18 Monitor	138.00	138.00	12	3		283
19	9202019	FALSE	2/18/2018	Import	ES	AT	140.25	GRM		19 Headset	140.00	140.00	11	4		282
20	9202020	FALSE	2/19/2018	Export	AT	IT	142.48	GRM		20 Notebook	142.00	142.00	23	5		281
21	9202021	FALSE	2/20/2018	Import	IT	DE	144.71	GRM		21 Speaker	145.00	145.00	11	2		280
22	9202022	FALSE	2/21/2018	Import	FR	AT	146.94	GRM		22 Monitor	147.00	147.00	12	3		279
23	9202023	FALSE	2/22/2018	Import	ES	AT	149.17	GRM		23 Headset	149.00	149.00	11	4		278
24	9202024	FALSE	2/23/2018	Export	AT	IT	151.40	GRM		24 Notebook	151.00	151.00	23	5		277
25	9202025	FALSE	2/24/2018	Import	IT	DE	153.63	GRM		25 Speaker	154.00	154.00	11	2		276
26	9202026	FALSE	2/25/2018	Import	FR	AT	155.86	GRM		26 Monitor	156.00	156.00	12	3		275
27	9202027	FALSE	2/26/2018	Import	ES	AT	158.09	GRM		27 Headset	158.00	158.00	11	4		274
28	9202028	FALSE	2/27/2018	Export	AT	IT	160.32	GRM		28 Notebook	160.00	160.00	23	5		273
29	9202029	FALSE	2/28/2018	Import	IT	DE	162.55	GRM		29 Speaker	163.00	163.00	11	2		272

### Run the format configuration

1. On the Configurations page, in the tree, select **Intrastat model > Intrastat report (PDF)**.
2. Select **Run**.
3. Select **Browse**. Navigate to and select the **Intrastat sample data.xlsx** file that you downloaded earlier as a prerequisite.
4. Select **OK**.
5. In the **Report direction** field, select **Both** to fill in all transactions from the imported Excel workbook in the PDF report that is generated.
6. Select **OK**.
7. Review the PDF document that is generated.

The follow illustration shows an example of the first page of the report that is generated.

out.Admin.pdf 1 / 3

Intrastat report				Number of lines	Amount	Weight	
Contoso Entertainment System Germany				Subtotal for arrivals	76	15946.33	19061.00
Reporting date				Subtotal for dispatches	24	5103.17	5989.00
2/4/2019				Total for report	100	21049.50	25050.00

Date	Direction	Commodity code	Country of dispatch/destination	Amount	Weight	Quantity of supplementary unit	Transaction code	Correction
1/31/2018	Arrivals	9202001	IT	100.11	300.00	1.00	11	No
2/1/2018	Arrivals	9202002	FR	102.34	299.00	2.00	12	No
2/2/2018	Arrivals	9202003	ES	104.57	298.00	3.00	11	No
2/3/2018	Dispatches	9202004	AT	106.80	297.00	4.00	23	No
2/4/2018	Arrivals	9202005	IT	109.03	296.00	5.00	11	No
2/5/2018	Arrivals	9202006	FR	111.26	295.00	6.00	12	No
2/6/2018	Arrivals	9202007	ES	113.49	294.00	7.00	11	No
2/7/2018	Dispatches	9202008	AT	115.72	293.00	8.00	23	No
2/8/2018	Arrivals	9202009	IT	117.95	292.00	9.00	11	No
2/9/2018	Arrivals	9202010	FR	120.18	291.00	10.00	12	No
2/10/2018	Arrivals	9202011	ES	122.41	290.00	11.00	11	No
2/11/2018	Dispatches	9202012	AT	124.64	289.00	12.00	23	No

The follow illustration shows an example of another page of the report that is generated.

out.Admin.pdf 2 / 3

3/9/2018	Arrivals	9202038	IT	182.62	263.00	38.00	11	No
3/10/2018	Arrivals	9202039	FR	184.85	262.00	39.00	12	No
3/11/2018	Arrivals	9202040	ES	187.08	261.00	40.00	11	No
3/12/2018	Dispatches	9202041	AT	189.31	260.00	41.00	23	No
3/13/2018	Arrivals	9202042	IT	191.54	259.00	42.00	11	No

Page 1 of 3

Intrastat report				Number of lines	Amount	Weight	
Contoso Entertainment System Germany				Subtotal for arrivals	76	15946.33	19061.00
Reporting date				Subtotal for dispatches	24	5103.17	5989.00
2/4/2019				Total for report	100	21049.50	25050.00

Date	Direction	Commodity code	Country of dispatch/destination	Amount	Weight	Quantity of supplementary unit	Transaction code	Correction
3/14/2018	Arrivals	9202043	FR	193.77	258.00	43.00	12	No
3/15/2018	Arrivals	9202044	ES	196.00	257.00	44.00	11	No
3/16/2018	Dispatches	9202045	AT	198.23	256.00	45.00	23	No

## Additional resources

- [ER Design a configuration for generating reports in OPENXML format \(November 2016\)](#)
- [Design ER configurations to generate reports in Word format](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Embed images and shapes in documents that you generate by using ER

2/18/2021 • 6 minutes to read • [Edit Online](#)

You can use the Electronic reporting (ER) tool to design reports that you can run to generate required electronic documents. You can use Microsoft Excel or Microsoft Word documents to specify the layout of a report. The ER Operations designer lets you attach the Excel or Word document as a template for the report. The named elements in the attached template are associated with the format elements of the ER report. Format elements of the report are bound to data sources. These elements specify the data that will be entered, at run time, in the documents that are generated.

This new functionality goes beyond existing ER capabilities for creating documents in Microsoft Office formats. For more information, play the following task guides. You can find these task guides under the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process.

- ER Design a configuration for generating reports in OPENXML format
- ER Design a configuration for generating reports in Microsoft WORD format

## Embed an image in an Excel document

First, you must add a placeholder for the image in an Excel document. Open an Excel workbook, and add a picture as a placeholder for the image that you will add later. Then use the ER tool to add a new ER format configuration to include the report that you're designing. Attach the Excel workbook as a template for the format of the report, and then import the content of the workbook into the ER format. The format definition will be created automatically. The image placeholder that you added will be included in the ER format definition as a **CELL** element.

### NOTE

You can manually specify the format definition instead of importing it. When you save your changes, the format will be validated.

Next, bind the **CELL** element of the ER format to the field from the format's data source that provides the picture's data in binary format at run time. When an ER data model is used as a format's data source, the data type of the field must be **CONTAINER**. Currently, an ER data model field that has the **CONTAINER** data type can be bound to several types of data sources that return images in binary format. You can access a field in a data table and a file that is attached to the data table's record by using the Document management framework.

### IMPORTANT

- If you want to fill the image placeholder in the document that you're creating by using the Excel template, the ER format must contain the **CELL** element that refers to the named picture element in the Excel template. Otherwise, no image placeholder will appear in the report's output. If the binding of a **CELL** element returns no data at run time, the document that is generated will show the image placeholder from the template. To hide an image in the document that you're generating, define a **CELL** element, and specify that the **Enabling** expression should return a value of **FALSE**.
- In the Excel template, use a unique name for every element. These elements include pictures and cells. If you duplicate an element name, the content of the report that is generated will be ambiguous and confusing.

## Embed a shape in an Excel document

First, you must add a placeholder for the shape in an Excel document. Open an Excel workbook, and select **Shape**, **Text box**, or **WordArt** as a placeholder for the shape. Then use the ER tool to add a new ER format configuration to include the report that you're designing. Attach the Excel workbook as a template for the format of the report, and then import the content of the workbook into the ER format. The format definition will be created automatically. The shape placeholder that you added will be included in the ER format definition as a **CELL** element that refers to the named Excel shape element.

### NOTE

You can manually specify the format definition instead of importing it. When you save your changes, the format will be validated.

Next, bind the **CELL** element in the ER format to the field from the format's data source that provides the data at run time. This data can be converted to a text string. When the **CELL** element in the ER format refers to a shape element in the Excel template that supports text, the text that is provided through this binding at run time will be shown in a shape in the document that is generated.

### IMPORTANT

- If you want to use the Excel template that includes the shape placeholder to generate a new document, the ER format must contain a **CELL** element that refers to the Excel shape element. Otherwise, no shape placeholder will appear in the report's output. If the binding of a **CELL** element that refers to the named Excel shape element returns no data at run time, the document that is generated will show the text of the shape placeholder from the Excel template. To hide a shape in the document that you're generating, define a **CELL** element that refers to the named Excel shape element, and specify that the **Enabling** expression should return a value of **FALSE**.
- In the Excel template, use a unique name for every element. These elements include shapes and cells. If you duplicate an element name, the content of the report that is generated will be ambiguous and confusing.

## Embed an image in a Word document

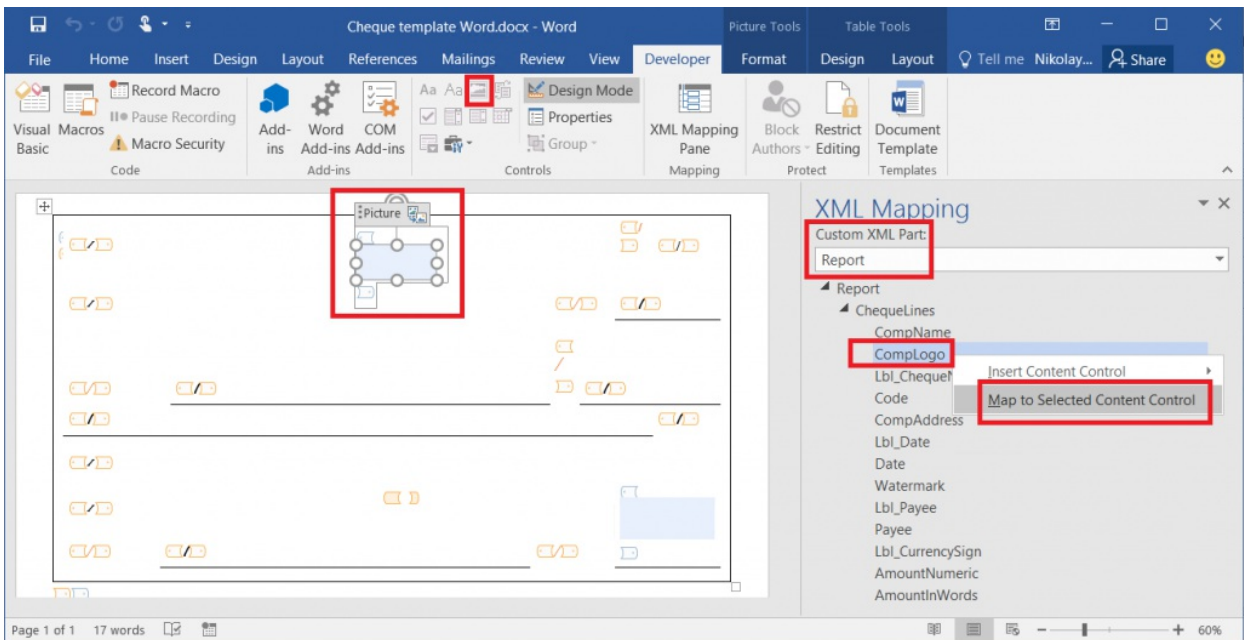
### IMPORTANT

You can reuse the ER format that uses an Excel template to create documents that include embedded images. In the ER format, make sure that a name is specified for the **CELL** element that refers to a named picture element in Excel, and that is bound to a data source that returns a picture at run time.

First, you must configure the Word document's layout. Use the **Picture Content** control to create a placeholder for the embedded image. To access this control, you must first make the **Developer** tab visible on the Word Ribbon.

Next, delete the Excel template from the ER format, and attach the Word template document. Update the reference to the template, and save your changes. The structure of the current ER format is saved to the Word template as a new custom XML part that is named **Report**.

Next, save the Word template for the current ER format to your local computer. Open the template, and open the **XML Mapping** pane. Find the custom XML part that is named **Report**, and then point to the **CELL** element in the ER report that is bound to a data source that returns an image in binary format. Map this XML part's item to the selected **Picture Content** control, and save your changes.



Finally, delete the Word template from the ER format, and attach the Word document that includes the mapped custom XML parts. Update the format reference to the template, and save the changes that you made to this ER format.

## More information

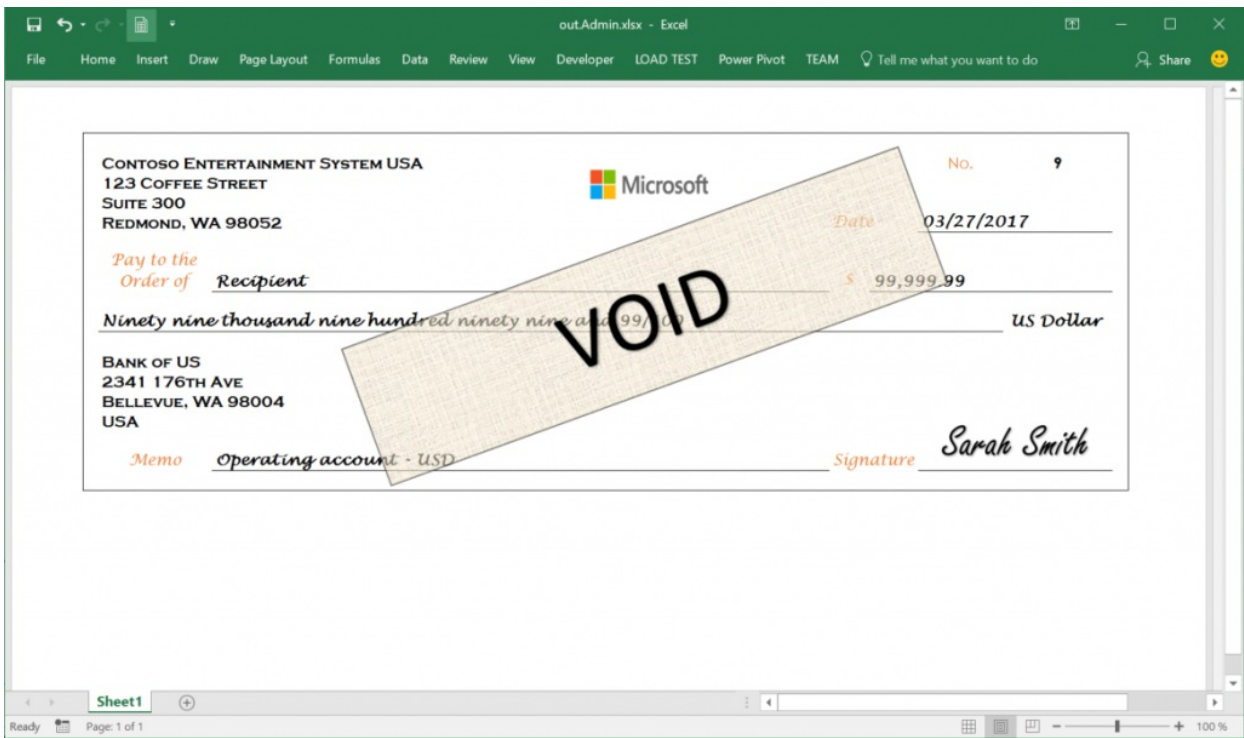
To become familiar with the details of this feature, play the set of task guides, **ER Make reports in MS Office formats with embedded images**. These task guides show how you can embed the images of a company logo and an authorized person's signature in the payment checks that are generated by using the ER tool in Excel and Word documents.

The following table lists the files that are required in order to complete the **ER Make reports in MS Office formats with embedded images** task guides. [Download](#) and save the files to your local computer.

DESCRIPTION	FILE NAME
ER data model configuration	Model for cheques.xml
ER format configuration	Cheques printing format.xml
Company logo image	Company logo.png
Signature image	Signature image.png
Alternative signature image	Signature image 2.png
Microsoft Word template for printing payment checks	Cheque template Word.docx
Microsoft Excel template for printing payment checks	Cheque template.xlsx

The following graphic provides an example of the test printout for a payment check that is generated from the Excel template.





#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Design configurations to generate reports in Office format that have embedded images

2/18/2021 • 3 minutes to read • [Edit Online](#)

To complete the steps in this procedure, first complete the procedure, "ER Create a configuration provider and mark it as active." This procedure explains how to design Electronic reporting (ER) configurations to generate a Microsoft Excel or Word document that contains embedded images. In this procedure, you will create the required ER configurations for the sample company, Litware, Inc. These steps can be completed using the USMF dataset. This procedure is created for users with the assigned role of system administrator or electronic reporting developer. Before you begin, download and save the files listed in the Help topic, [Embed images and shapes in documents that you generate by using ER](#). The files are: Model for cheques.xml, Cheques printing format.xml, Company logo.png, Signature image.png, Signature image 2.png, and Cheque template Word.docx.

## Verify prerequisites

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as Active. If you don't see this configuration provider, complete the steps in the procedure, "Create a configuration provider and mark it as active."
3. Click Reporting configurations.

## Add a new ER model configuration

1. Instead of creating a new model, you can load the ER model configuration file (Model for cheques.xml) that you saved earlier. This file contains the sample data model for payment cheques and the mapping of the data model to the data components of the Dynamics 365 for Operations application.
2. On the Versions FastTab, click Exchange.
3. Click Load from XML file.
4. Click Browse, and then select Model for cheques.xml.
5. Click OK.
6. The loaded model will be used as a data source of information to generate documents that contain images in Excel and Word.

## Add a new ER format configuration

1. Instead of creating a new format, you can load the ER format configuration file (Cheques printing format.xml) that you saved earlier. This file contains the sample layout of the format to print cheques using the pre-printed form and the mapping of this format to the 'Model for cheques' data model.
2. Click Exchange.
3. Click Load from XML file.
4. Click Browse and select the Cheques printing format.xml file.
5. Click OK.
6. In the tree, expand 'Model for cheques'.
7. In the tree, select 'Model for cheques\Cheques printing format'.
8. The loaded format will be used to generate documents that contain images in Excel and Word.

## Configure ER user parameters

1. On the Action Pane, click Configurations.
2. Click User parameters.
3. Select Yes in the Run settings field.  
Turn on the 'Run draft' flag to start the draft version of the selected format instead of the completed one.
4. Click OK.

## Configure Cash & bank management parameters

1. Go to Cash and bank management > Bank accounts > Bank accounts.
2. Use the Quick Filter to filter on the Bank account field with a value of 'USMF OPER'.
3. On the Action Pane, click Set up.
4. Click Check.
5. Expand the Setup section.
6. Click Edit.
7. Select Yes in the Company logo field.
8. Click Company logo.
9. Click Change.
10. Click Browse and select the file that you downloaded earlier, Company logo.png.
11. Click Save.
12. Close the page.
13. Expand the Signature section.
14. Select Yes in the Print first signature field.
15. Click Change.
16. Click Browse and select the file that you downloaded earlier, Signature image.png.
17. Expand the Copies section.
18. In the Watermark field, select an option.
19. Select Yes in the Generic electronic Export format field.
20. Select 'Cheques printing form' configuration.
21. Now the selected ER format will be used for printing cheques.
22. Click Attach.
23. Click New.
24. Click File.
25. Click Browse and select the file that you downloaded earlier, Signature image 2.png.
26. Close the page.
27. Close the page.
28. Close the page.
29. Go to Cash and bank management > Setup > Cash and bank management parameters.
30. Select Yes in the Allow prenote creation on inactive bank accounts: field.
31. Click Save.
32. Close the page.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Review configurations to generate reports in Office format that have embedded images

2/18/2021 • 3 minutes to read • [Edit Online](#)

To complete these steps, you must first complete the steps in the "ER Make reports in MS Office formats with embedded images (Part 1: Set up parameters)" task guide.

This procedure shows how to design Electronic reporting (ER) configurations to generate electronic documents that contain embedded images in Microsoft Excel and Microsoft Word. In this example, you will review ER configurations for the sample company Litware, Inc.

This procedure is intended for users who have the System administrator or Electronic reporting developer role assigned to them. The steps can be completed by using the USMF data set.

## Review the imported data model

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, select 'Model for cheques'.
3. Click Designer.
  - This model is designed to represent payment cheques from the business standpoint and the mapping of this model to the application's data sources. Review this model by the ER Operations designer. Note the attributes of the model elements that are presented: structure, name, description, data type, and so on.
4. In the tree, expand 'root'.
5. In the tree, select 'root\cheques'.
6. In the tree, expand 'root\cheques'.
7. In the tree, expand 'root\cheques\attributes'.
8. In the tree, expand 'root\payer'.
9. In the tree, select 'root\istestrun'.
10. In the tree, select 'root\layout'.
  - The layout element of this model represents the details of the printing cheque form layout for the selected bank account. It also includes two nodes of the Container data type to store images.
11. In the tree, expand 'root\layout'.
12. In the tree, select 'root\layout\company logo'.
13. In the tree, expand 'root\layout\company logo'.
14. In the tree, select 'root\layout\company logo\image'.
15. In the tree, select 'root\layout\company logo\isprinted'.
16. In the tree, select 'root\layout\signature'.
17. In the tree, expand 'root\layout\signature'.
18. In the tree, select 'root\layout\signature\image'.
19. In the tree, select 'root\layout\signature\isprinted'.
  - Note that two image data model elements are bound to the fields of the tables that contain images of the company logo and the authorized person's signature in binary format.
20. In the tree, expand 'root\layout\watermark'.
21. Click Map model to datasource.
22. Click Designer.

23. In the tree, expand 'chequeselected'.
24. In the tree, expand 'layout'.
25. In the tree, expand 'layout\company logo'.
26. In the tree, expand 'layout\signature'.
27. In the tree, expand 'layout\watermark'.
28. Toggle 'Show details' on.
  - Note that the cheques data model element is bound to the TmpChequePrintout table that, at runtime, will contain records for cheques that the user has selected for printing.
29. Close the page.
30. Close the page.
31. Close the page.

## Review the imported format

1. In the tree, expand 'Model for cheques'.
2. In the tree, select 'Model for cheques\Cheques printing format'.
3. Click Designer.
4. Click Attachments.
5. Click Open.
  - Open the attached report's template in Excel.
  - Review the attached report's Excel template that will be used to print cheques. The template contains two cheques per page and is designed to print cheques to the preprinted form. Note that two blank images are embedded. These blank images are for the company logo and the signature of the person who is authorizing a payment. Verify that the images are named CompLogo and SignatureImage, respectively, in Excel.
6. Close the page.
7. In the tree, expand 'Report'.
8. In the tree, expand 'Report\ChequeLines'.
9. In the tree, select 'Report\ChequeLines\CompLogo'.
10. Toggle 'Show details' on.
  - Note that the 'CompLogo' format's cell element represents the Excel item that is used to populate the company logo image in the report. This format element is bound to the image data model element that, at runtime, contains a company logo image in binary format.
11. Click the Mapping tab.
12. Click Edit enabled.
  - Note that you can make the 'CompLogo' format's cell element so that it's no longer enabled. In this case, the associated Excel image element will hide a company logo in the generated report. If the enabled expression returns TRUE and the defined binding brings no image, the associated Excel image element will show an image that has been saved in the Excel template.
13. Close the page.
14. In the tree, expand 'labels: Container'.
  - Some labels that are presented in the preprinted cheque form will be included in the report when it's created for testing purposes. However, those labels won't be printed during real printing, because the preprinted form already includes them.
15. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Generate reports in Office format that have embedded images

2/18/2021 • 3 minutes to read • [Edit Online](#)

The following steps explain how a user playing either 'System administrator' or 'Electronic reporting developer' role can design Electronic reporting (ER) configurations to generate electronic documents in MS office formats (Excel and Word) containing embedded images.

In this example, you will use created ER configurations for sample company, 'Litware, Inc.'. To complete these steps, you must first complete the steps in the "ER Make reports in MS Office formats with embedded images (Part 2: Review configurations)" task guide. These steps can be performed in 'USMF' company.

## Run format with initial model mapping

1. Go to Cash and bank management > Bank accounts > Bank accounts.
2. Use the Quick Filter to filter on the Bank account field with a value of 'USMF OPER'.
3. On the Action Pane, click Set up.
4. Click Check.
5. Click Print test.
  - Run the format for testing purposes.
6. Select Yes in the Negotiable check format field.
7. Click OK.
  - Review the created output. The company logo is presented in the report as well as the authorized person's signature. The signature image is taken from the field of the 'Container' data type of the cheque layout record that is associated with the selected bank account.
8. Expand the Copies section.
9. Click Edit.
10. In the Watermark field, enter 'Print watermark as Void'.
  - Change the watermark layout setting to show the watermark text in generating document in an Excel shape element.
11. Click Print test.
12. Click OK.
  - Review the created output. The watermark is shown in the created report in accordance to the selection option.
13. Close the page.
14. On the Action Pane, click Manage payments.
15. Click Checks.
16. Click Show filters.
17. Apply the following filters: Enter a filter value of "381","385","389" on the "Check number" field using the "is one of" filter operator.
18. In the list, mark all rows.
19. Click Print check copy.
  - Run the format to reprint the selected cheques.
  - Review the created output. The selected cheques have been reprinted. The company logo and labels are not printed out since they are presented on the pre-printed form.

## Modify the mapping of the imported data model

1. Close the page.
2. Close the page.
3. Go to Organization administration > Electronic reporting > Configurations.
4. In the tree, select 'Model for cheques'.
5. Click Designer.
6. Click Map model to datasource.
7. Click Designer.
  - We will change the binding of the data model's signature item to get the signature image from the file that has been attached to the cheque layout record that is associated with the selected bank account.
8. Turn off Show details.
9. In the tree, expand 'layout'.
10. In the tree, expand 'layout\signature'.
11. In the tree, select 'layout\signature\image = chequesaccount.<Relations'.BankChequeLayout.Signature1Bmp'.
12. In the tree, expand 'chequesaccount'.
13. In the tree, expand 'chequesaccount<Relations'.
14. In the tree, expand 'chequesaccount<Relations\BankChequeLayout'.
15. In the tree, expand 'chequesaccount<Relations\BankChequeLayout<Relations'.
16. In the tree, expand 'chequesaccount<Relations\BankChequeLayout<Relations<Documents'.
17. In the tree, select 'chequesaccount<Relations\BankChequeLayout<Relations<Documents\getFileContentAsContainer()'.
18. Click Bind.
19. Click Save.
20. Close the page.
21. Close the page.
22. Close the page.
23. Close the page.

## Run format using the adjusted model mapping

1. Go to Cash and bank management > Bank accounts > Bank accounts.
2. Use the Quick Filter to find records. For example, filter on the Bank account field with a value of 'USMF OPER'.
3. On the Action Pane, click Set up.
4. Click Check.
5. Click Print test.
6. Click OK.
  - Review the created output. The image from the Document Management attachment is presented as the signature of an authorized person.

## Use MS Word document as a template in the imported format

1. Close the page.
2. Close the page.
3. Go to Organization administration > Electronic reporting > Configurations.
4. In the tree, expand 'Model for cheques'.
5. In the tree, select 'Model for cheques\Cheques printing format'.



6. Click Designer.
7. Click Attachments.
8. Click Delete.
9. Click Yes.
10. Click New.
11. Click File.
  - Click Browse and select the downloaded in advance 'Cheque template Word.docx' file.
12. Close the page.
13. In the Template field, enter or select a value.
14. Click Save.
15. Close the page.
16. Click Edit.
17. Select Yes in the Run Draft field.
18. Close the page.
19. Go to Cash and bank management > Bank accounts > Bank accounts.
20. Use the Quick Filter to filter on the Bank account field with a value of 'USMF OPER'.
21. Click Check.
22. Click Print test.
23. Click OK.
  - Review the created output. The output has been generated as a Word document with embedded images presenting the company logo, the signature of an authorized person and the selected text of the watermark.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Modify Electronic reporting formats by reapplying Excel templates

2/18/2021 • 2 minutes to read • [Edit Online](#)

The Electronic reporting (ER) tool is used to generate business documents in an electronic format. To generate a business document, you must create an ER format, and then use the ER designer to define the layout of the business document and specify the data that should be included in it. You can then run the ER format to generate the business document.

The ER tool can be used to generate business documents as Microsoft Excel files. You can use an Excel document as a template for these documents. To define the document layout in the ER designer, you can import the contents of the Excel document that you want to use as a template into the defined ER format. For more details, and to practice this scenario, play the task guide **ER Design a configuration for generating reports in OPENXML format** (part of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process).

If you edit the Excel document that is used as a template for a business document, new ER functionality lets you reapply the updated template to the ER format. The ER format is then updated so that it adheres to the updated template. For more details about this functionality, play the task guide **ER Modify a format by reapplying an Excel template** (part of the 7.5.5.3 Acquire/Develop IT service/solution components (10683) business process). In the task guide step where you import an updated template, use the modified template of the Payment Report Excel file, SampleVendPaymWsReport2, as a template.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Define the dependency of ER configurations on other components

2/18/2021 • 6 minutes to read • [Edit Online](#)

To complete these steps, you must first complete the steps in the task guide, ER Manage model mapping configurations, and you must have access to Microsoft Dynamics Lifecycle Services (LCS).

This procedure shows how to design an Electronic reporting (ER) configuration and specify its dependency from other software components, so that you can help guarantee that the configuration is correctly downloaded to a specific version of Finance and Operations. In this example, you will create required ER configurations for the sample company Litware, Inc.

This procedure is intended for users who have the System administrator or Electronic reporting developer role assigned to them. The steps can be performed in any company, because ER configurations are shared among companies.

1. Go to Organization administration > Electronic reporting > Configurations.
  - Make sure that the configurations tree contains the 'Sample data model' configuration and subordinate items. Otherwise, complete the steps in the task guide, ER Manage model mapping configurations, and then start this guide again.

## Define the dependency of ER configurations from other components

1. In the tree, expand 'Sample data model'.
2. In the tree, select 'Sample data model\Sample mapping'.
  - We selected the draft version of the 'Sample mapping' model mapping configuration. We will now define its dependency from other software components. This step is considered a prerequisite for controlling the download of this configuration's version from an ER repository and any further use of this version.
3. Expand the Prerequisites section.
  - Note that the 'Implementations' prerequisites group has been added automatically at this stage. This group contains the prerequisite component that refers to the data model configuration and has the Implementation flag turned on. This flag indicates that the 'Sample mapping' mapping configuration is considered the implementation of the 'Sample data model' data model. This component will force ER to download the 'Sample mapping' mapping configuration from an ER repository whenever the 'Sample data model' model configuration is downloaded.
4. Click Edit.
  - A single dependency of the current version of a configuration from a software component can be specified by using the definition of the component's type, and either the component version or a range of component versions.
  - Desired dependencies can be grouped together. When the 'All of' grouping type is selected, the dependency condition of this group is considered satisfied when each dependency condition from this group and subordinate group is satisfied. When the 'One of' grouping type is selected, the dependency condition of this group is considered satisfied when at least one dependency condition from this group is satisfied.
5. Click New.
6. Select Product prerequisite component.
7. Select Microsoft Dynamics 365 for Operations (1611).

8. In the Version field, type '[7.1.1541.3036,8)'.
  - [7.1.1541.3036,8)
  - Dependencies that you enter will be evaluated when this configuration is downloaded from any ER repository. This configuration version will be downloaded from the ER repository when version 1 of the 'Sample data model' configuration is either already in place or downloaded in advance. If it's downloaded in advance, it must be completed in Finance and Operations version 7.1.1541.3036 or later, but must not exceed major version 8.
9. Click Save.
10. Close the page.
11. Click Change status.
12. Click Complete.
13. Click OK.
14. In the tree, select 'Sample data model\Sample mapping (alternative)'.
15. Click Edit.
16. Click New.
17. Select Product prerequisite component.
18. Select Microsoft Dynamics AX 7.0 RTW.
19. In the Version field, type '[7.0.1265.3015,7.1)'.
  - [7.0.1265.3015,7.1)
  - Dependencies will be evaluated when the configuration is downloaded from any ER repository. This configuration version will be downloaded from the ER repository when version 1 of the 'Sample data model' configuration is either already in place or downloaded in advance. If it's downloaded in advance, it must be completed in Microsoft Dynamics 365 for Finance and Operations, Enterprise edition, the version of which must be 7.0.1265.3015 or later, but must not exceed minor version 1.
20. Click Save.
21. Close the page.
22. Click Change status.
23. Click Complete.
24. Click OK.

## Configure the ER repository

1. Close the page.
2. Go to Organization administration > Workspaces > Electronic reporting.
  - Open the list of ER repositories for the current ER provider, Litware, Inc.
3. In the list, mark the selected row.
4. Click Repositories.
5. Click Show filters.
6. Enter a filter value of "LCS" on the "Type name" field using the "contains" filter operator.
  - If the LCS repository is already registered for the current ER provider, you can skip the remaining steps in this sub-task. If the LCS repository isn't already registered, complete the remaining steps.
7. Click Add to open the drop dialog.
8. In the Configuration repository type field, enter 'LCS'.
9. Click Create repository.
10. In the Project field, enter or select a value.
  - Select the desired LCS project from the lookup of the 'Project' field.
11. Click OK.
12. Close the page.

## Upload configurations to LCS

1. Click Reporting configurations.
2. In the tree, select 'Sample data model'.
3. Select the completed version of this configuration.
4. Click Change status.
5. Click Share.
6. Click OK.
  - Version 1 of this model configuration has been uploaded to LCS by using the LCS project for the ER repository that was previously configured.
7. In the tree, expand 'Sample data model'.
8. In the tree, select 'Sample data model\Sample mapping'.
9. Select the completed version of this configuration.
10. Click Change status.
11. Click Share.
12. Click OK.
  - Version 1.1 of this model mapping configuration has been uploaded to LCS by using the LCS project for the ER repository that was previously configured.
13. In the tree, select 'Sample data model\Sample mapping (alternative)'.
14. Select the completed version of this configuration.
15. Click Change status.
16. Click Share.
17. Click OK.
  - Version 1.1 of this model mapping configuration has been uploaded to LCS by using the LCS project for the ER repository that was previously configured.

## Evaluate ER configuration dependencies

We will delete created configurations from the system and download them back from the LCS repository.

1. In the tree, select 'Sample data model\Sample mapping'.
2. Click Delete.
3. Click Yes.
4. In the tree, select 'Sample data model\Sample mapping (alternative)'.
5. Click Delete.
6. Click Yes.
7. In the tree, select 'Sample data model\Sample format'.
8. Click Delete.
9. Click Yes.
10. In the tree, select 'Sample data model'.
11. Click Delete.
12. Click Yes.
13. Close the page.
  - Open the list of ER repositories for the current ER provider, Litware, Inc.
14. Click Repositories.
15. Click Show filters.
16. Enter a filter value of "LCS" on the "Type name" field using the "contains" filter operator.
17. Click Open.

18. In the tree, select 'Sample data model'.
  - Note that you can view an evaluation of whether prerequisite conditions have been satisfied for each version of the ER configurations for the current repository. To view this evaluation, click Check prerequisites.
19. Click Check prerequisites.
20. Click Import.
21. Click Yes.
22. Close the page.
23. Close the page.
24. Close the page.
25. Go to Organization administration > Electronic reporting > Configurations.
26. In the tree, expand 'Sample data model'.
  - Note that the model 'Sample mapping' mapping configuration has been downloaded together with the selected data model configuration. The two files are downloaded together because 'Sample mapping' has been defined as implementing the selected data model, and because it's applicable for the application. The 'Sample mapping (alternative)' configuration hasn't been downloaded because the condition for the required application version isn't satisfied.
  - If you sign in to Finance and Operations, register the same provider, access the same LCS project, and download the same data model configuration, the 'Sample mapping (alternative)' configuration will download, whereas the 'Sample mapping' configuration will be skipped.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use financial dimensions as a data source (Part 1 - Design data model)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how either a system administrator or electronic reporting developer can configure an Electronic reporting (ER) model to use financial dimensions as a data source for ER reports. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the procedure, "Create a configuration provider and mark it as active".

## Create a new data model

1. Go to Organization administration > Workspaces > Electronic reporting.
  - Make sure that the "Litware, Inc." provider is available and marked as active.
2. Click Reporting configurations.
3. Click Create configuration to open the drop dialog.
4. In the Name field, type 'Financial dimensions sample model'.
5. Click Create configuration.
6. Click Designer.
7. Click New to open the drop dialog.
8. In the Name field, type 'Entry'.
9. Click Add.
10. Click New to open the drop dialog.
11. In the Name field, type 'Company'.
12. Click Add.
  - We will add to our model a new record list. This list will expose (for any ER reports using this model as data source) the settings of selected financial dimensions. Each financial dimension will be presented in this list as a record with appropriate fields representing dimension's setting.
13. Click New to open the drop dialog.
14. In the Name field, type 'Dimensions setting'.
15. In the Item type field, select 'Record list'.
16. Click Add.
17. Click New to open the drop dialog.
18. In the Name field, type 'Code'.
19. In the Item type field, select 'String'.
20. Click Add.
21. Click New to open the drop dialog.
22. In the Name field, type 'Name'.
23. Click Add.
24. In the tree, select 'Entry'.
25. Click New to open the drop dialog.
26. In the Name field, type 'Journal'.
27. In the Item type field, select 'Record list'.
28. Click Add.

29. Click New to open the drop dialog.
30. In the Name field, type 'Batch'.
31. In the Item type field, select 'String'.
32. Click Add.
33. Click New to open the drop dialog.
34. In the Name field, type 'Transaction'.
35. In the Item type field, select 'Record list'.
36. Click Add.
37. Click New to open the drop dialog.
38. In the Name field, type 'Date'.
39. In the Item type field, select 'Date'.
40. Click Add.
41. Click New to open the drop dialog.
42. In the Name field, type 'Debit'.
43. In the Item type field, select 'Real'.
44. Click Add.
45. Click New to open the drop dialog.
46. In the Name field, type 'Credit'.
47. Click Add.
48. Click New to open the drop dialog.
49. In the Name field, type 'Currency'.
50. In the Item type field, select 'String'.
51. Click Add.
52. Click New to open the drop dialog.
53. In the Name field, type 'Voucher'.
54. Click Add.
55. Click New to open the drop dialog.
56. In the Name field, type 'Dimensions data'.
57. In the Item type field, select 'Record list'.
58. Click Add.
  - We added to our model a new record list. This list will expose (for any ER reports using this model as data source) the values of selected financial dimensions. Each financial dimension will be presented in this list as a record with appropriate fields representing dimension's values. Dimension name will be also presented in this record as a field to be used, if needed, for selection purposes.
59. Click New to open the drop dialog.
60. In the Name field, type 'Code'.
61. In the Item type field, select 'String'.
62. Click Add.
63. Click New to open the drop dialog.
64. In the Name field, type 'Description'.
65. Click Add.
66. Click New to open the drop dialog.
67. In the Name field, type 'Name'.
68. Click Add.
69. Click Save.
70. Close the page.



Finance and Operations Preview Search for a page USMF

Save Create model Compare Map model to datasource Translate Options

Data model | FINANCIAL DIMENSIONS SAMPLE MODEL : 1

### Financial dimensions sample model

+ New Delete

SEARCH

- Entry
  - Company
- Dimensions setting
  - Code
  - Name
- Journal
  - Batch
- Transaction
  - Credit
  - Currency
  - Date
  - Debit
  - Dimensions data**
  - Code
  - Description
  - Name
  - Voucher

**Data model**

Root reference

**GENERAL**

Name: Financial dimensions sample m...

Description:

**Node**

Go to referenced item Switch item reference

**GENERAL**

Name: Dimensions data

Type: Record list

Label (\*Recommended to use labels):

Description:

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use financial dimensions as a data source (Part 2 - Model mapping)

2/18/2021 • 4 minutes to read • [Edit Online](#)

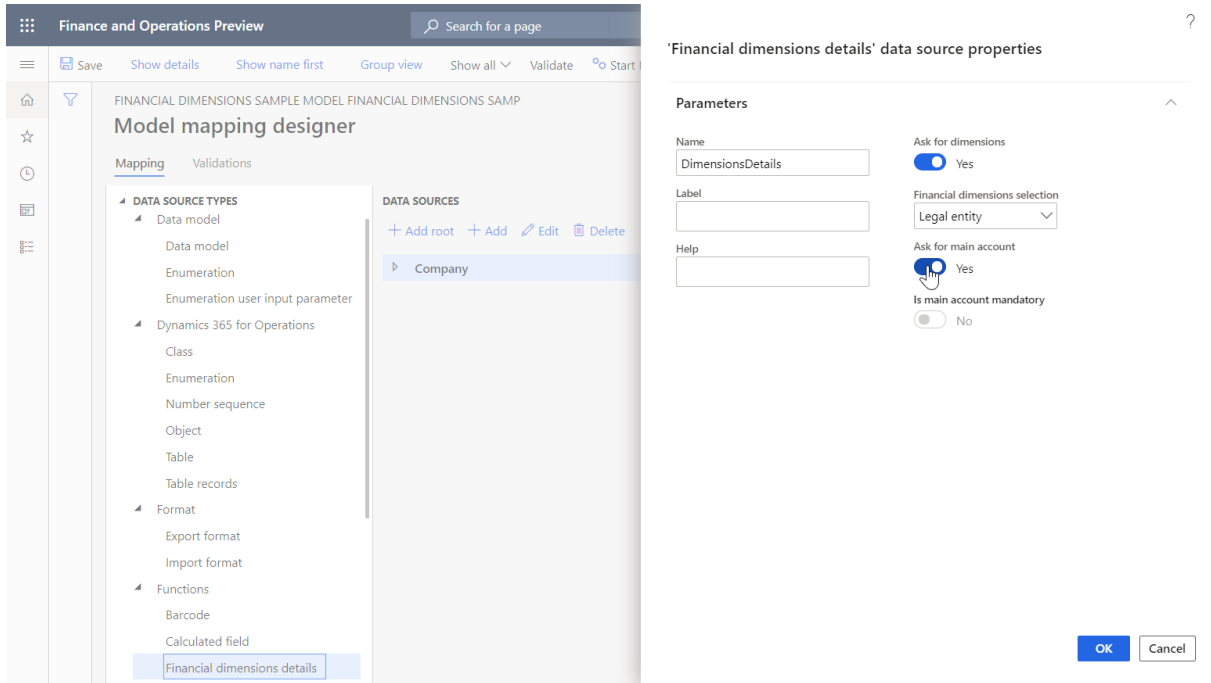
The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) model to use financial dimensions as a data source for ER reports. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the "ER Use financial dimensions as a data source (Part 1: Design data model)" procedure.

## Add required data sources to model mapping

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, select 'Financial dimensions sample model'.
3. Click Designer.
4. Click Map model to datasource.
5. Click New.
6. In the Definition field, select Entry.
7. In the Name field, type 'Dimensions data mapping'.
8. In the Description field, type 'Dimensions data mapping'.
9. Click Save.
10. Click Designer.
11. In the tree, select 'Dynamics 365 for Operations\Table'.
12. Click Add root.
13. In the Name field, type 'Company'.
14. In the Table field, type 'CompanyInfo'.
15. Click OK.
16. In the tree, select 'Functions\Financial dimensions details'.
17. Click Add root.
  - This data source specifies how the scope of financial dimensions will be defined for any report that will use this model as a data source.
18. In the Name field, type a value.
19. Select Yes in the Ask for dimensions field.
  - Select Yes to allow the user to select dimensions at run-time on the User dialog form. If set to No, all financial dimensions of the current instance will be used by default.
20. In the Financial dimensions selection field, select 'Legal entity'.
  - Select All to allow the user to select desired dimensions for the current instance in the Lookup field. Select Legal entity to allow the user to select dimensions for the company in the Lookup field. Select Dimension to allow the user to select dimensions using a single dimension set.
21. Select Yes in the Ask for main account field.
  - Set 'Ask for main account' to Yes to allow users to select the main account as part of the list of dimensions. If set to No, the main account will not be included to the list of dimensions and the 'Is main account mandatory' option is enabled. If 'Is main account mandatory' is set to Yes, include the main account in the list of dimensions regardless of the user's selection.

22. Click OK.



23. In the tree, select 'Dynamics 365 for Operations\Table records'.

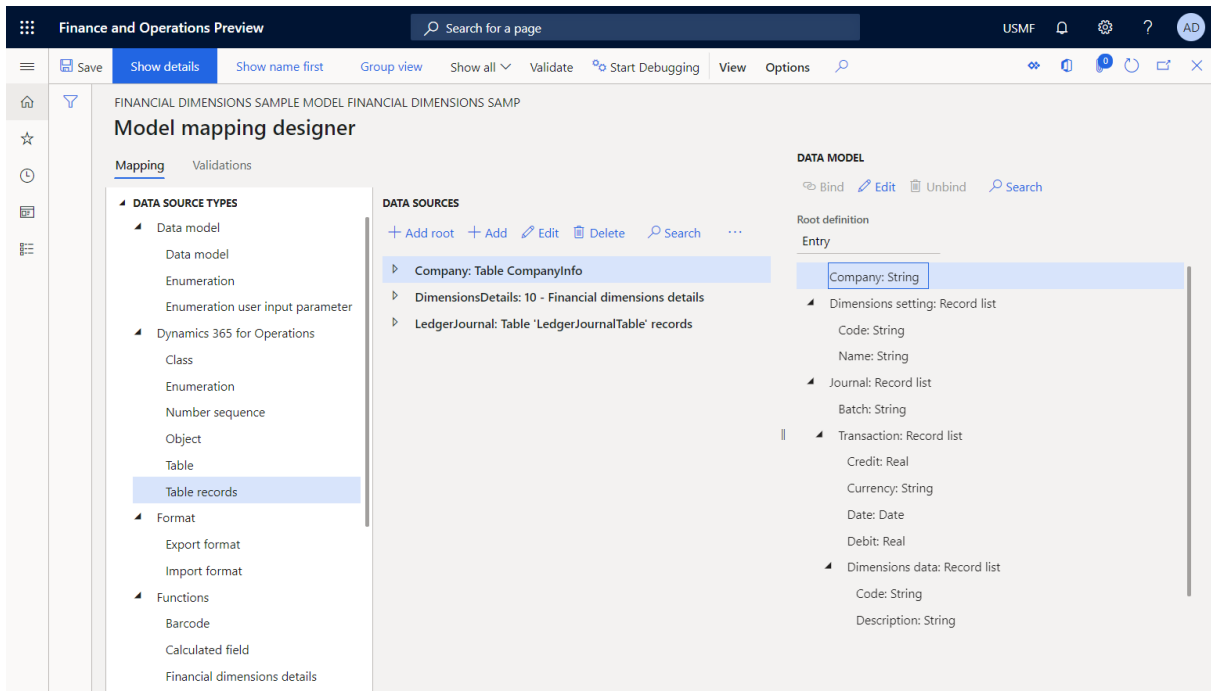
24. Click Add root.

25. In the Name field, type 'LedgerJournal'.

26. Select Yes in the Ask for query field.

27. In the Table field, type 'LedgerJournalTable'.

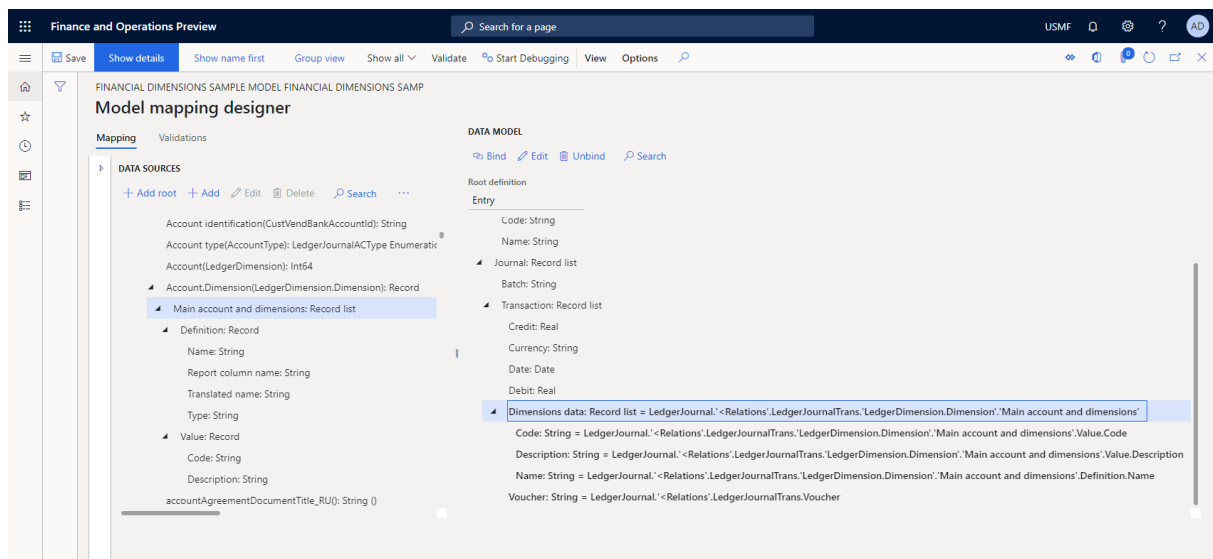
28. Click OK.



## Map data model elements to added data sources

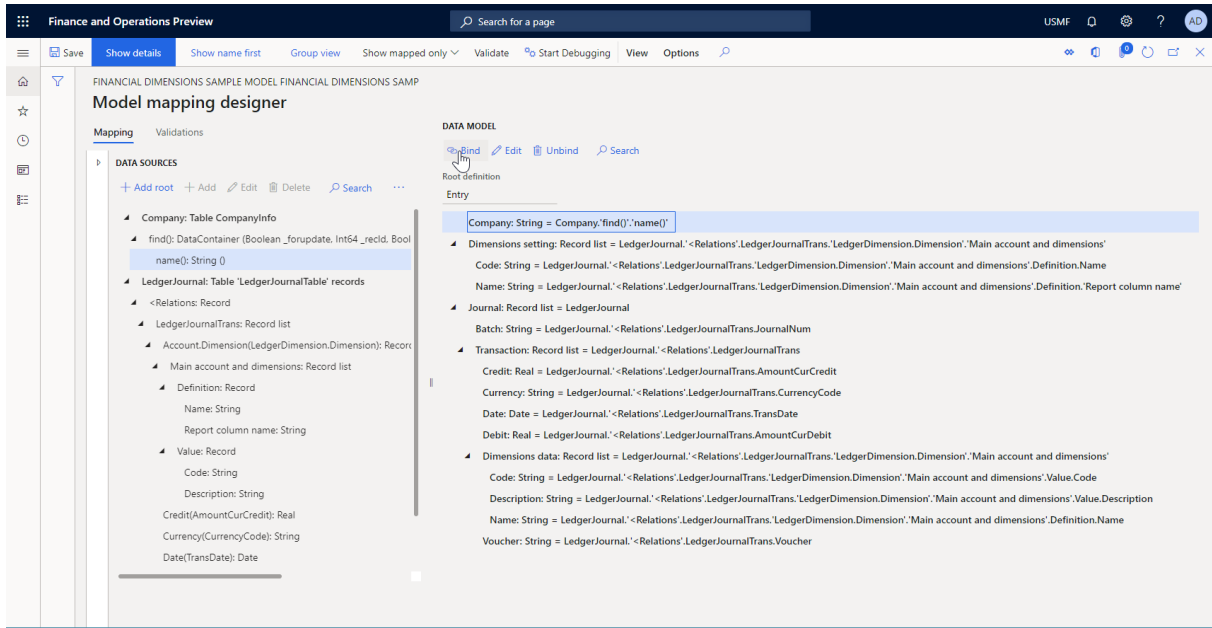
1. In the tree, expand 'Journal'.
2. In the tree, expand 'Journal\Transaction'.
3. In the tree, expand 'Journal\Transaction\Dimensions data'.
4. In the tree, expand 'Dimensions setting'.
5. In the tree, expand 'LedgerJournal'.
6. In the tree, expand 'LedgerJournal<Relations'.

7. In the tree, expand 'LedgerJournal<Relations\LedgerJournalTrans'.
8. In the tree, select 'LedgerJournal<Relations\LedgerJournalTrans\Voucher'.
9. In the tree, select 'Journal\Transaction\Voucher'.
10. Click Bind.
11. In the tree, select  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)'.
  - Note that for any reference to financial dimensions that is set to, for instance, LedgerDimension, a corresponding data source item is available (LedgerDimension.Dimension). This data source item offers the financial dimensions of that dimensions set as the record's list.
12. In the tree, expand  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)'.
13. In the tree, expand  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)\Main account and dimensions'.
14. In the tree, expand  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)\Main account and dimensions\Value'.
15. In the tree, expand  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)\Main account and dimensions\Definition'.
16. In the tree, select  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)\Main account and dimensions\Definition\Name'.
17. In the tree, select 'Journal\Transaction\Dimensions data\Name'.
18. Click Bind.
19. In the tree, select  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)\Main account and dimensions\Value\Description'.
20. In the tree, select 'Journal\Transaction\Dimensions data\Description'.
21. Click Bind.
22. In the tree, select  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)\Main account and dimensions\Value\Code'.
23. In the tree, select 'Journal\Transaction\Dimensions data\Code'.
24. Click Bind.
25. In the tree, select  
'LedgerJournal<Relations\LedgerJournalTrans\Account.Dimension(LedgerDimension.Dimension)\Main account and dimensions'.
26. In the tree, select 'Journal\Transaction\Dimensions data'.
27. Click Bind.



28. In the tree, select 'LedgerJournal<Relations>LedgerJournalTrans\Debit(AmountCurDebit)'.
29. In the tree, select 'Journal\Transaction\Debit'.
30. Click Bind.
31. In the tree, select 'LedgerJournal<Relations>LedgerJournalTrans\Date(TransDate)'.
32. In the tree, select 'Journal\Transaction\Date'.
33. Click Bind.
34. In the tree, select 'LedgerJournal<Relations>LedgerJournalTrans\Currency(CurrencyCode)'.
35. In the tree, select 'Journal\Transaction\Currency'.
36. Click Bind.
37. In the tree, select 'LedgerJournal<Relations>LedgerJournalTrans\Credit(AmountCurCredit)'.
38. In the tree, select 'Journal\Transaction\Credit'.
39. Click Bind.
40. In the tree, select 'LedgerJournal<Relations>LedgerJournalTrans'.
41. In the tree, select 'Journal\Transaction'.
42. Click Bind.
43. In the tree, select 'LedgerJournal\Journal batch number(JournalNum)'.
44. In the tree, select 'Journal\Batch'.
45. Click Bind.
46. In the tree, select 'LedgerJournal'.
47. In the tree, select 'Journal'.
48. Click Bind.
49. In the tree, expand 'Dimensions'.
50. In the tree, expand 'Dimensions\Main account and dimensions'.
51. In the tree, expand 'Dimensions\Main account and dimensions\Definition'.
52. In the tree, select 'Dimensions\Main account and dimensions\Definition\Name'.
53. In the tree, select 'Dimensions setting\Code'.
54. Click Bind.
55. In the tree, select 'Dimensions\Main account and dimensions\Definition\Report column name'.
56. In the tree, select 'Dimensions setting\Name'.
57. Click Bind.
58. In the tree, select 'Dimensions\Main account and dimensions'.
59. In the tree, select 'Dimensions setting'.
60. Click Bind.
61. In the tree, select 'Company'.

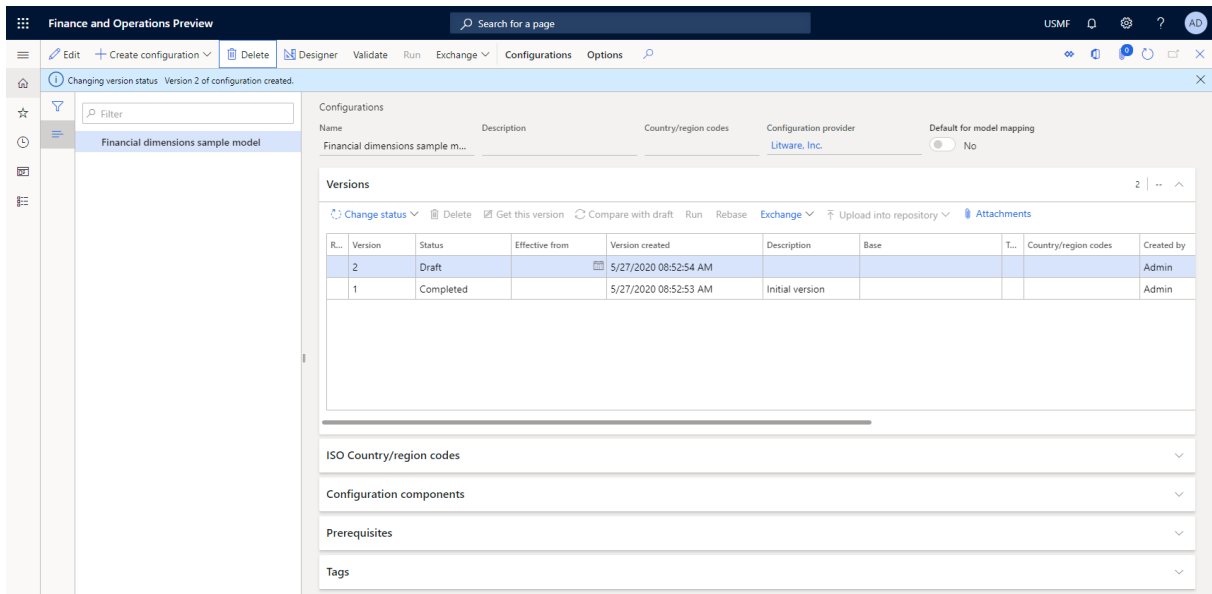
62. Click Edit.
63. In the expressionAsStringText field, enter 'Company.find().name()'.
  - Company.find().name()
64. Click Save.



65. Close the page.
66. Click Save.
67. Close the page.

## Complete this draft model's version

1. Close the page.
2. Close the page.
3. Click Change status.
4. Click Complete.
5. Click OK.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use financial dimensions as a data source (Part 3 - Design the report)

2/18/2021 • 4 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) model to use financial dimensions as a data source for ER reports. These steps can be performed in any company.

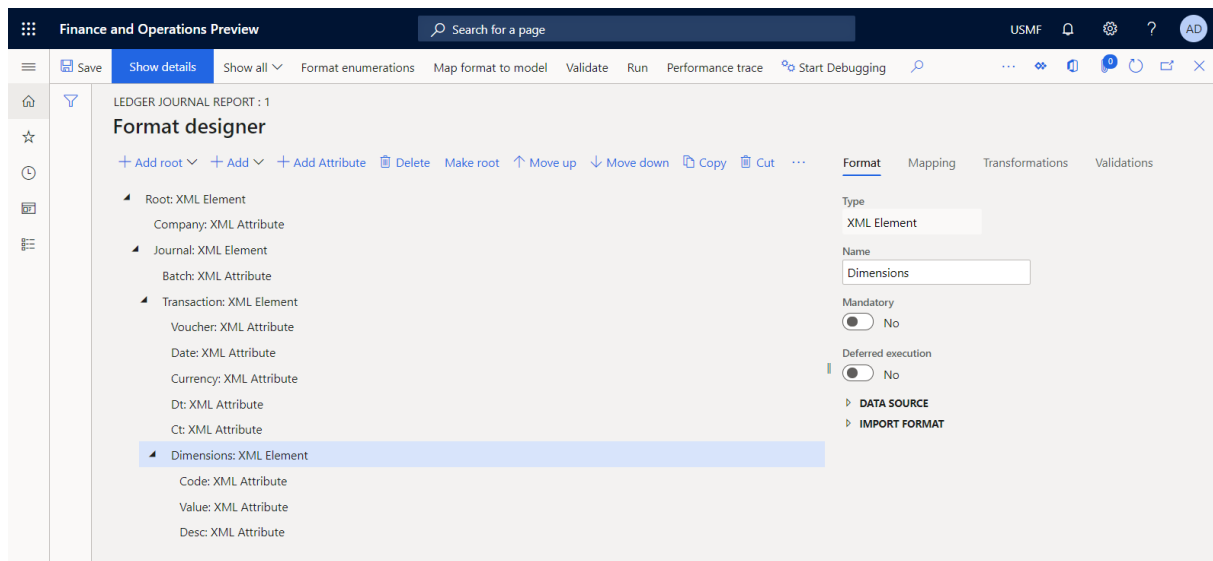
To complete these steps, you must first complete the steps in the "ER Use financial dimensions as a data source (Part 2: Model mapping)" procedure.

## Design a report to present financial dimensions

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, select 'Financial dimensions sample model'.
3. Click Create configuration to open the drop dialog.
4. In the New field, enter 'Format based on data model Financial dimensions sample model'.
  - Use the model that was created in advance as the data source for your new report.
5. In the Name field, type 'Ledger journal report'.
6. In the Data model definition field, select Entry.
7. Click Create configuration.
8. Click Designer.
9. Click Add root to open the drop dialog.
10. In the tree, select 'XML\Element'.
11. In the Name field, type 'Root'.
12. Click OK.
13. Click Add to open the drop dialog.
14. In the tree, select 'XML\Attribute'.
15. In the Name field, type 'Company'.
16. Click OK.
17. Click Add to open the drop dialog.
18. In the tree, select 'XML\Element'.
19. In the Name field, type 'Journal'.
20. Click OK.
21. In the tree, select 'Root: XML Element\Journal: XML Element'.
22. Click Add to open the drop dialog.
23. In the tree, select 'XML\Attribute'.
24. In the Name field, type 'Batch'.
25. Click OK.
26. Click Add to open the drop dialog.
27. In the tree, select 'XML\Element'.
28. In the Name field, type 'Transaction'.
29. Click OK.
30. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element'.
31. Click Add to open the drop dialog.



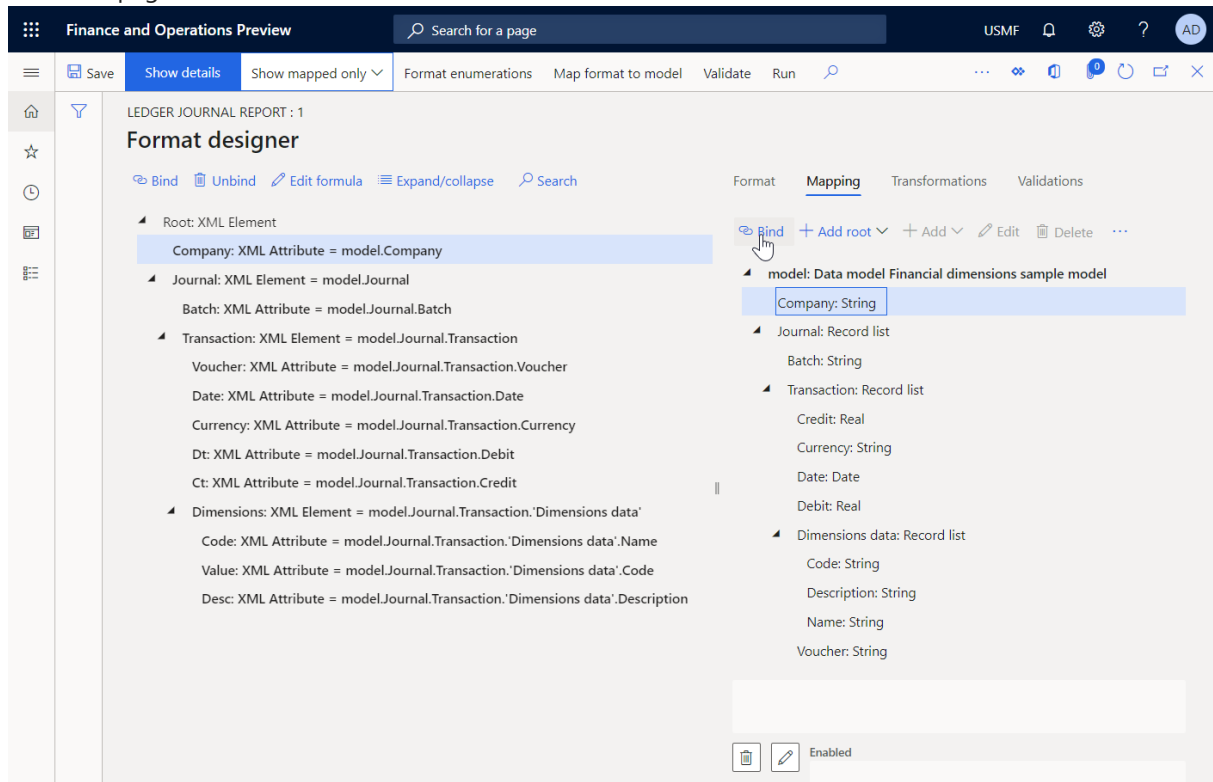
32. In the tree, select 'XML\Attribute'.
33. In the Name field, type 'Voucher'.
34. Click OK.
35. Click Add Attribute.
36. In the Name field, type 'Date'.
37. Click OK.
38. Click Add Attribute.
39. In the Name field, type 'Currency'.
40. Click OK.
41. Click Add Attribute.
42. In the Name field, type 'Dt'.
43. Click OK.
44. Click Add Attribute.
45. In the Name field, type 'Ct'.
46. Click OK.
47. Click Add to open the drop dialog.
48. In the tree, select 'XML\Element'.
49. In the Name field, type 'Dimensions'.
50. Click OK.
51. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Dimensions: XML Element'.
52. Click Add to open the drop dialog.
53. In the tree, select 'XML\Attribute'.
54. In the Name field, type 'Code'.
55. Click OK.
56. Click Add Attribute.
57. In the Name field, type 'Value'.
58. Click OK.
59. Click Add Attribute.
60. In the Name field, type 'Desc'.
61. Click OK.



Map report elements to data sources

1. Click the Mapping tab.
2. In the tree, expand 'model: Data model Financial dimensions sample model'.
3. In the tree, expand 'model: Data model Financial dimensions sample model\Journal: Record list'.
4. In the tree, expand 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list'.
5. In the tree, expand 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list'.
6. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Dimensions: XML Element\Desc: XML Attribute'.
7. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list\Description: String'.
8. Click Bind.
9. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Dimensions: XML Element\Value: XML Attribute'.
10. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list\Code: String'.
11. Click Bind.
12. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Dimensions: XML Element\Code: XML Attribute'.
13. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list\Name: String'.
14. Click Bind.
15. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list'.
16. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Dimensions: XML Element'.
17. Click Bind.
18. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Ct: XML Attribute'.
19. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Credit: Real'.
20. Click Bind.
21. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Dt: XML Attribute'.
22. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Debit: Real'.
23. Click Bind.
24. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Currency: XML Attribute'.
25. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Currency: String'.
26. Click Bind.
27. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Date: XML Attribute'.
28. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Date: Date'.
29. Click Bind.
30. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element\Voucher: XML Attribute'.
31. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Voucher: String'.

32. Click Bind.
33. In the tree, select 'Root: XML Element\Journal: XML Element\Transaction: XML Element'.
34. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list'.
35. Click Bind.
36. In the tree, select 'Root: XML Element\Journal: XML Element\Batch: XML Attribute'.
37. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Batch: String'.
38. Click Bind.
39. In the tree, select 'Root: XML Element\Journal: XML Element'.
40. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list'.
41. Click Bind.
42. In the tree, select 'Root: XML Element\Company: XML Attribute'.
43. In the tree, select 'model: Data model Financial dimensions sample model\Company: String'.
44. Click Bind.
45. Click Save.
46. Close the page.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use financial dimensions as a data source (Part 4 - Run the report)

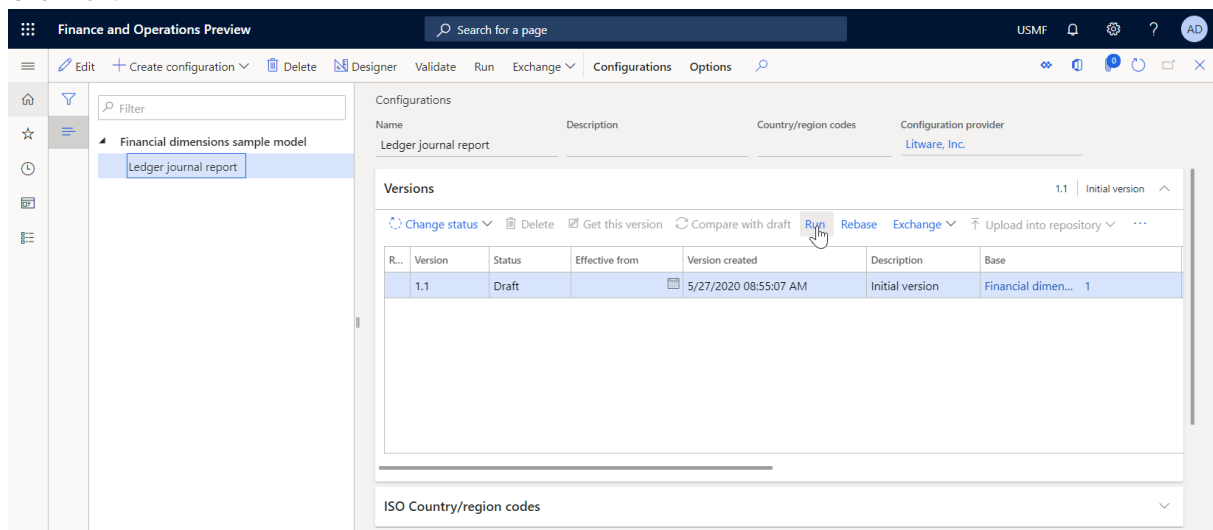
2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) model to use financial dimensions as a data source for ER reports. These steps can be performed in the DEMF company.

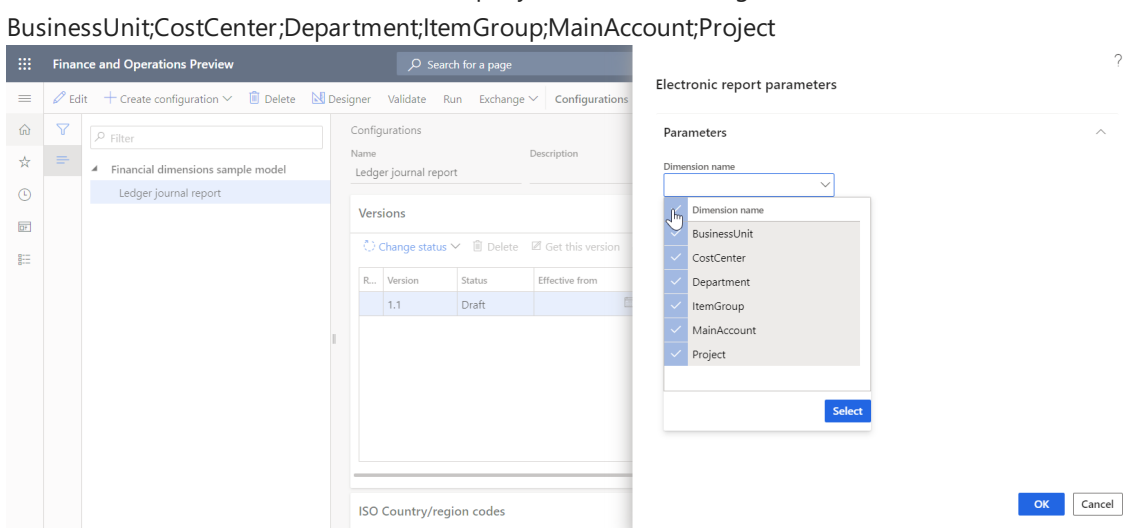
To complete these steps, you must first complete the steps in the "ER Use financial dimensions as a data source (Part 3: Design the report)" procedure. You must also configure default document types on the Electronic reporting parameters page. Default document types are also set when you download and import any ER configuration.

## Run report

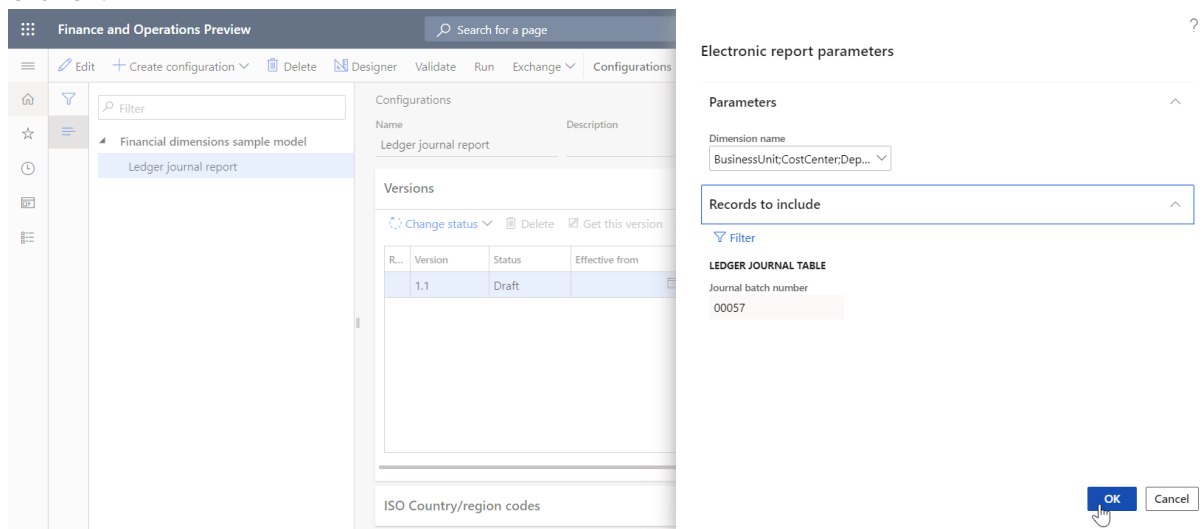
1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Financial dimensions sample model'.
3. In the tree, select 'Financial dimensions sample model\Ledger journal report'.
4. Click Run.



5. In the Dimension name field, enter or select a value.
- To select all dimensions in the current company, enter the following information:  
**BusinessUnit;CostCenter;Department;ItemGroup;MainAccount;Project**



6. Expand the Records to include section.
7. Click Filter.
8. Select the row for the Ledger journal table and the Journal batch number field.
9. In the Criteria field, type '00057'.
10. Click OK.
11. Click OK.



- Review the generated output. For each transaction of the selected batch, the financial dimensions from the corresponding dimensions set are presented. Run this report and select different dimensions to see that the report is not dependent on the number of selected dimensions or the number of dimensions configured for this instance.

```

<?xml version="1.0" encoding="UTF-8"?>
<Root Company="Contoso Entertainment System Germany">
  <Journal Batch="00057">
    <Transaction Ct="0" Dt="1800" Currency="EUR" Date="2015-01-20" Voucher="GNJL000005">
      <Dimensions Desc="Pension/Profit-Sharing Plan Expense" Value="602180" Code="MainAccount"/>
      <Dimensions Desc="Auto" Value="002" Code="BusinessUnit"/>
      <Dimensions Desc="" Value="" Code="CostCenter"/>
      <Dimensions Desc="Human Resources" Value="026" Code="Department"/>
      <Dimensions Desc="" Value="" Code="ItemGroup"/>
      <Dimensions Desc="" Value="" Code="Project"/>
    </Transaction>
    <Transaction Ct="0" Dt="900" Currency="EUR" Date="2015-01-20" Voucher="GNJL000005">
      <Dimensions Desc="Pension/Profit-Sharing Plan Expense" Value="602180" Code="MainAccount"/>
      <Dimensions Desc="Auto" Value="002" Code="BusinessUnit"/>
      <Dimensions Desc="" Value="" Code="CostCenter"/>
      <Dimensions Desc="IT Department" Value="025" Code="Department"/>
      <Dimensions Desc="" Value="" Code="ItemGroup"/>
      <Dimensions Desc="" Value="" Code="Project"/>
    </Transaction>
    <Transaction Ct="0" Dt="1800" Currency="EUR" Date="2015-01-20" Voucher="GNJL000005">
      <Dimensions Desc="Pension/Profit-Sharing Plan Expense" Value="602180" Code="MainAccount"/>
      <Dimensions Desc="Auto" Value="002" Code="BusinessUnit"/>
      <Dimensions Desc="" Value="" Code="CostCenter"/>
      <Dimensions Desc="Finance" Value="024" Code="Department"/>
      <Dimensions Desc="" Value="" Code="ItemGroup"/>
      <Dimensions Desc="" Value="" Code="Project"/>
    </Transaction>
    <Transaction Ct="0" Dt="4200" Currency="EUR" Date="2015-01-20" Voucher="GNJL000005">
      <Dimensions Desc="Pension/Profit-Sharing Plan Expense" Value="602180" Code="MainAccount"/>
      <Dimensions Desc="Auto" Value="002" Code="BusinessUnit"/>
      <Dimensions Desc="" Value="" Code="CostCenter"/>
      <Dimensions Desc="Operations" Value="023" Code="Department"/>
      <Dimensions Desc="" Value="" Code="ItemGroup"/>
      <Dimensions Desc="" Value="" Code="Project"/>
    </Transaction>
    <Transaction Ct="0" Dt="21300" Currency="EUR" Date="2015-01-20" Voucher="GNJL000005">
      <Dimensions Desc="Pension/Profit-Sharing Plan Expense" Value="602180" Code="MainAccount"/>
      <Dimensions Desc="Auto" Value="002" Code="BusinessUnit"/>
      <Dimensions Desc="" Value="" Code="CostCenter"/>
      <Dimensions Desc="Sales & Marketing" Value="022" Code="Department"/>
      <Dimensions Desc="" Value="" Code="ItemGroup"/>
      <Dimensions Desc="" Value="" Code="Project"/>
    </Transaction>
  </Journal Batch>
</Root>

```

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Configure format to do counting and summing (Part 1 - Create format)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to do counting and summing based on data of the already generated text output. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the "Create a configuration provider and mark it as active" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Get access to the list of configurations provided by Microsoft

1. Go to Organization administration > Workspaces > Electronic reporting.
  - Make sure that the "Litware, Inc." provider is available and marked as active.
2. Select the "Litware, Inc." provider.
3. Click Repositories.
  - If a repository of the "Operations resources" type already exists, skip the remaining steps of the current sub-task.
4. Click Add to open the drop dialog.
5. In the Configuration repository type field, enter 'Operations resources'.
6. Click Create repository.
7. Click OK.

## Get the Intrastat configurations provided by Microsoft

1. Click Open.
2. In the tree, select 'Intrastat model\Intrastat (DE)'.
3. Click Import.
  - Click Import for version 1.1 of the selected configuration.
4. Click Yes.
5. Close the page.
6. Close the page.
7. Click Reporting configurations.
8. In the tree, expand 'Intrastat model'.
9. In the tree, select 'Intrastat model\Intrastat (DE)'.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Configure format to do counting and summing (Part 2 - Configure computations)

2/18/2021 • 4 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to do counting and summing based on data of the already generated text output. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the "ER Configure format to do counting and summing (Part 1: Create format)" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Create a format configuration to add counting and summing details

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Reporting configurations.
3. In the tree, expand 'Intrastat model'.
4. In the tree, select 'Intrastat model\Intrastat (DE)'.
  - Assume that you need to customize the format provided by Microsoft by adding lines with summary details at the end of the Intrastat report. You need to do that by deriving our own instance of the Intrastat configuration from the Microsoft instance to make modifications.
5. Click Create configuration to open the drop dialog.
6. In the New field, enter 'Derive from Name: Intrastat (DE), Microsoft'.
7. In the Name field, type 'Intrastat (DE) with counting & summing'.
8. Click Create configuration.

## Configure this report to do counting and summation based on output details

1. Click Designer.
2. Select Yes in the Collect output details field.
  - This flag will activate at run-time the process of collecting output details for generating the Intrastat file.
  - You need to do counting for different Intrastat directions, so add a dedicated model enumeration to the data sources' list of this format configuration.
3. Click the Mapping tab.
4. Click Add root to open the drop dialog.
5. In the tree, select 'Data model\Enumeration '.
6. In the Name field, type 'Direction'.
7. In the Model enumeration field, enter or select a value.
  - Select the value Direction.
8. Click OK.
9. Click Add root to open the drop dialog.
10. In the tree, select 'Functions\Calculated field'.
11. In the Name field, type '\$BlockName'.

12. Click Edit formula.
13. In the Formula field, enter ""block"".
14. Click Save.
15. Close the page.
16. Click OK.
17. Click Add root to open the drop dialog.
18. In the tree, select 'Functions\Calculated field'.
19. In the Name field, type '\$RecName'.
20. Click Edit formula.
21. In the Formula field, enter ""record"".
22. Click Save.
23. Close the page.
24. Click OK.
25. Click Add root to open the drop dialog.
26. In the tree, select 'Functions\Calculated field'.
27. In the Name field, type '\$InvName'.
28. Click Edit formula.
29. In the Formula field, enter ""InvoicedAmountEUR"".
30. Click Save.
31. Close the page.
32. Click OK.
33. In the tree, select 'Intrastat\Data'.
34. Click Edit button for the 'Collected data key name' field
35. Click Add data source.
  - \$BlockName
36. Click Save.
37. Close the page.
38. Click the Edit button for the Collected data key value field.
39. In the Formula field, enter 'IF(Intrastat.CommodityRecord.Direction=Direction.Import, "Import", "Export")'.
  - IF(Intrastat.CommodityRecord.Direction=Direction.Import, "Import", "Export")
40. Click Save.
41. Close the page.
  - Count the lines of this sequence. The results will be used with the name "block" separately for different directions. Value "Import" will be used for any arrivals Intrastat transactions. The value "Export" will be used for any Intrastat dispatches transactions. Consider this to be a virtual Excel spreadsheet. For each transaction a row where the first column "block" is filled with the values "Import" and "Export" accordingly.
42. In the tree, expand 'Intrastat\Data: Sequence'.
43. In the tree, select 'Intrastat\Data: Sequence\Arrivals?'.
44. Click Edit button for the 'Collected data key name' field.
  - Count the lines of this sequence. The results will be memorized using the name "record".
45. In the tree, select '\$RecName'.
46. Click Add data source.
47. Click Save.
48. Close the page.
49. Click Edit button for the 'Collected data key value' field
50. In the Formula field, enter 'Intrastat.CommodityRecord.CommodityCode'.



51. Click Save.
52. Close the page.
  - Count the lines of this sequence. The results will be used with the name "record" separately for different commodity codes. Consider this to be a virtual Excel spreadsheet. For each transaction a row where the first column "block" is filled with the values "Import" and "Export" accordingly and the second block "record" is filled with the commodity code value.
53. In the tree, select 'Intrastat\Data: Sequence\Dispatches?'
54. Click Edit button for the 'Collected data key name' field
55. In the tree, select '\$RecName'.
56. Click Add data source.
57. Click Save.
58. Close the page.
59. Click the Edit button for the 'Collected data key value' field.
60. In the Formula field, enter 'Intrastat.CommodityRecord.CommodityCode'.
61. Click Save.
62. Close the page.
63. In the tree, expand 'Intrastat\Data: Sequence\Dispatches: Sequence?'
64. In the tree, expand 'Intrastat\Data: Sequence\Dispatches: Sequence?\Record = Intrastat.CommodityRecord'
65. Click the Format tab.
66. In the tree, select 'Intrastat\Data\Dispatches\Record\Invoice amount EUR'
67. Click the Mapping tab.
68. Click the Edit button for the 'Collected data key name' field.
69. In the tree, select '\$InvName'.
70. Click Add data source.
71. Click Save.
72. Close the page.
  - Summarize the invoiced amount values for lines of this sequence. The results will be used with the name "InvoicedAmountEUR" separately for different Intrastat directions and commodity codes. Consider this to be a virtual creation in Excel spreadsheet. For each transaction a row where the first column "block" is filled with the values "Import" and "Export" accordingly. The second block "record" is filled with the commodity code value, and the third column "InvoicedAmountEUR" is filled with the invoice amount value.
73. In the tree, expand 'Intrastat\Data\Arrivals?'
74. In the tree, expand 'Intrastat\Data\Arrivals?\Record = Intrastat.CommodityRecord'
75. Click the Format tab.
76. In the tree, select 'Intrastat\Data\Arrivals\Record\Invoice amount EUR'
77. Click the Mapping tab.
78. Click the Edit button for the 'Collected data key name' field.
79. In the tree, select '\$InvName'.
80. Click Add data source.
81. Click Save.
82. Close the page.
83. Click Save.
84. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Configure format to do counting and summing (Part 3 - Use computations to make the output)

2/18/2021 • 3 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to do counting and summing based on data of the already generated text output. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the "ER Configure format to do counting and summing (Part 2: Configure computations)" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Configure this report to use counting and summing info

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Reporting configurations.
3. In the tree, expand 'Intrastat model'.
4. In the tree, expand 'Intrastat model\Intrastat (DE)'.
5. In the tree, select 'Intrastat model\Intrastat (DE)\Intrastat (DE) with counting & summing'.
6. Click Designer.
7. Click the Mapping tab.
8. Click Add root to open the drop dialog.
  - Add a new data source to get the list of memorized blocks.
9. In the tree, select 'Functions\Calculated field'.
10. In the Name field, type '\$BlocksList'.
  - \$BlocksList
11. Click Edit formula.
12. In the tree, select 'Data collection functions\COLLECTEDLIST'.
13. Click Add function.
14. Click Add data source.
15. In the Formula field, enter 'COLLECTEDLIST('\$BlockName', '
  - COLLECTEDLIST('\$BlockName',
16. In the Formula field, enter 'COLLECTEDLIST('\$BlockName', "\*"')'.
  - COLLECTEDLIST('\$BlockName', "\*"')
17. Click Save.
  - The pattern "\*" means that all blocks will be included to the list for this record.
18. Close the page.
19. Click OK.
20. Click the Format tab.
21. In the tree, select 'Intrastat\Data'.
22. Click Add to open the drop dialog.
23. In the tree, select 'Text\Sequence'.
24. In the Name field, type 'Totals by blocks'.
  - Totals by blocks

25. In the Special characters field, select 'New line - Windows (CR LF)'.
26. Click OK.
27. In the tree, select 'Intrastat\Data\Totals by blocks'.
28. Click Add to open the drop dialog.
29. In the tree, select 'Text\String'.
30. In the Name field, type 'Block code'.
  - Block code
31. Click OK.
32. Click Add String.
33. In the Name field, type 'Lines counting'.
  - Lines counting
34. Click OK.
35. Click Add String.
36. In the Name field, type 'Total amount'.
  - Total amount
37. Click OK.
38. Click the Mapping tab.
39. In the tree, select '\$BlocksList'.
40. Click Bind.
  - Create a summary line for each memorized block.
41. Click the Format tab.
42. In the tree, select 'Intrastat\Data\Totals by blocks\Block code'.
43. Click the Mapping tab.
44. Click Edit formula.
45. In the Formula field, enter '"Block id: " & '.
  - "Block id: " &
46. In the tree, expand '\$BlocksList'.
47. In the tree, select '\$BlocksList\Value'.
48. Click Add data source.
49. Click Save.
50. Close the page.
51. Click the Format tab.
52. In the tree, select 'Intrastat\Data\Totals by blocks\Lines counting'.
53. Click the Mapping tab.
54. Click Edit formula.
  - Create output for the number of lines for each block presented in this report.
55. In the Formula field, enter '"Number of lines in this block: " & '.
  - "Number of lines in this block: " &
56. In the Formula field, enter '"Number of lines in this block: " & TEXT('.
  - "Number of lines in this block: " & TEXT(
57. In the tree, select 'Data collection functions\COUNTIFS'.
58. Click Add function.
59. Click Add data source.
60. In the Formula field, enter '"Number of lines in this block: " & TEXT(COUNTIFS('\$BlockName', '.
  - "Number of lines in this block: " & TEXT(COUNTIFS('\$BlockName',
61. In the tree, expand '\$BlocksList'.

62. In the tree, select '\$BlocksList\Value'.
63. Click Add data source.
64. In the Formula field, enter "'Number of lines in this block: " & TEXT(COUNTIFS('\$BlockName', '\$BlocksList'.Value, '  
• "Number of lines in this block: " & TEXT(COUNTIFS('\$BlockName', '\$BlocksList'.Value,
65. In the tree, select '\$RecName'.
66. Click Add data source.
67. In the Formula field, enter "'Number of lines in this block: " & TEXT(COUNTIFS('\$BlockName', '\$BlocksList'.Value, '\$RecName', "\*"'))'.  
• "Number of lines in this block: " & TEXT(COUNTIFS('\$BlockName', '\$BlocksList'.Value, '\$RecName', "\*"'))
68. Click Save.
69. Close the page.
70. Click the Format tab.
71. In the tree, select 'Intrastat\Data\Totals by blocks\Total amount'.
72. Click the Mapping tab.
73. Click Edit formula.  
• Create output that will be the total of the invoiced amount for each block presented in this report.
74. In the Formula field, enter "'Sum of invoiced amount: " & TEXT(SUMIFS('\$InvName', '\$BlockName', '\$BlocksList'.Value, '\$RecName', "\*"'))'.  
• "Sum of invoiced amount: " & TEXT(SUMIFS('\$InvName', '\$BlockName', '\$BlocksList'.Value, '\$RecName', "\*"'))
75. Click Save.
76. Close the page.
77. Click Save.
78. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Configure format to do counting and summing (Part 4 - Run format)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to do counting and summing based on data of the already generated text output. These steps can be performed in the DEMF company.

To complete these steps, you must first complete the steps in the "ER Configure format to do counting and summing (Part 3: Use computations to make the output)" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Test this configuration for generation of the Intrastat reports

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Reporting configurations.
3. In the tree, expand 'Intrastat model'.
4. In the tree, expand 'Intrastat model\Intrastat (DE)'.
5. In the tree, select 'Intrastat model\Intrastat (DE)\Intrastat (DE) with counting & summing'.
6. On the Action Pane, click Configurations.
7. Click User parameters.
8. Select Yes in the Run settings field.
9. Click OK.
10. Click Edit.
11. Select Yes in the Run Draft field.
12. Click Save.
13. Go to Tax > Setup > Foreign trade > Foreign trade parameters.
14. Expand the Electronic reporting section.
15. Select the "Intrastat (DE) with counting & summing" configuration.
16. Select the "Intrastat (DE) with counting & summing" configuration.
17. Click Save.
18. Close the page.
19. Go to Tax > Declarations > Foreign trade > Intrastat.
20. Click Output.
21. Click Report.
  - Run the Intrastat report generation process.
22. In the From date field, set the date to '2000-01-01'.
  - Define start and end dates for the reporting period that include the existing on the form transactions.
23. In the To date field, set the date to '2022-12-31'.
  - Define start and end dates for the reporting period that include the existing on the form transactions.
24. In the Direction field, select 'Arrivals'.
25. Select Yes in the Generate file field.
26. Click OK.
  - Review the created output with the summary lines in the end.

27. Click New.
28. In the list, mark the selected row.
29. In the Direction field, select 'Dispatches'.
30. In the Item number field, enter or select a value.
31. In the Commodity field, enter or select a value.
32. Set Weight to '10'.
33. Set Invoice amount to '10000'.
34. Set Statistical amount to '10000'.
35. Click Output.
36. Click Report.
37. In the Direction field, select 'Dispatches'.
38. Click OK.
  - Review the created output with the summary lines in the end. Note that it has been changed in comparison to the first run.

## Run this configuration in debug mode to review the collected counting & summing data

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Intrastat model'.
3. In the tree, expand 'Intrastat model\Intrastat (DE)'.
4. In the tree, select 'Intrastat model\Intrastat (DE)\Intrastat (DE) with counting & summing'.
5. On the Action Pane, click Configurations.
6. Click User parameters.
7. Select Yes in the Run in debug mode field.
8. Click OK.
9. Close the page.
10. Go to Tax > Declarations > Foreign trade > Intrastat.
11. Click Output.
12. Click Report.
13. Click OK.
14. Close the page.
15. Go to Organization administration > Electronic reporting > Configurations.
16. In the tree, expand 'Intrastat model'.
17. In the tree, expand 'Intrastat model\Intrastat (DE)'.
18. In the tree, select 'Intrastat model\Intrastat (DE)\Intrastat (DE) with counting & summing'.
19. Click Debug logs.
  - Note that a debug log record has been created for the execution process of the selected configuration.
20. Click Attach.
21. Click Open.
  - Review the created XML file that contains counting and summing details that were collected during the execution of the selected configuration.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use horizontally expandable ranges to dynamically add columns in Excel reports (Part 1 - Design format)

2/18/2021 • 5 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to generate reports as OPENXML worksheets (Excel) files in which the required columns can be created dynamically as horizontally expandable ranges. These steps can be performed in any company.

To complete these steps, you must first complete these three task guides:

"ER Create a configuration provider and mark it as active"

"ER Use financial dimensions as a data source (Part 1: Design data model)"

"ER Use financial dimensions as a data source (Part 2: Model mapping)"

You must also download and save a local copy of the template with a sample report found here, [Sample Financial Dimensions Web Service Report](#).

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Create a new report configuration

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, select 'Financial dimensions sample model'.
3. Click Create configuration to open the drop dialog.
4. In the New field, enter 'Format based on data model Financial dimensions sample model'.
  - Use the model created in advance as the data source for your new report.
5. In the Name field, type 'Sample report with horizontally expandable ranges'.
  - Sample report with horizontally expandable ranges
6. In the Description field, type 'To make Excel output with dynamically adding columns'.
  - To make Excel output with dynamically adding columns
7. In the Data model definition field, select Entry.
8. Click Create configuration.

## Design the report format

1. Click Designer.
2. Turn on the 'Show details' toggle button.
3. On the Action Pane, click Import.
4. Click Import from Excel.
5. Click Attachments.
  - Import the report's template. Use Excel file that you downloaded for that.
6. Click New.
7. Click File.
8. Close the page.



9. In the Template field, enter or select a value.
  - Select the downloaded template.
10. Click OK.
  - Add a new range to dynamically create Excel output with as many columns as you selected (in the user dialog form) for financial dimensions. Each cell for every column will represent a single financial dimension's name.
11. Click Add to open the drop dialog.
12. In the tree, select 'Excel\Range'.
13. In the Excel range field, type 'DimNames'.
  - DimNames
14. In the Replication direction field, select 'Horizontal'.
15. Click OK.
16. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Horizontal'.
17. Click Move up.
18. In the tree, select 'Excel = "SampleFinDimWsReport"\Cell'.
19. Click Cut.
20. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Horizontal'.
21. Click Paste.
22. In the tree, expand 'Excel = "SampleFinDimWsReport"\Range: Horizontal'.
23. In the tree, expand 'Excel = "SampleFinDimWsReport"\Range: Vertical'.
24. In the tree, expand 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical'.
25. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical'.
  - Add a new range to dynamically create Excel output with as many columns as you selected (in the user dialog form) for financial dimensions. Each cell for every column will represent a single financial dimension's value for each reporting transaction.
26. Click Add Range.
27. In the Excel range field, type 'DimValues'.
  - DimValues
28. In the Replication direction field, select 'Horizontal'.
29. Click OK.
30. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Cell'.
31. Click Cut.
32. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Range: Horizontal'.
33. Click Paste.
34. In the tree, expand 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Range: Horizontal'.

## Map format elements to data sources

1. Click the Mapping tab.
2. In the tree, expand 'model: Data model Financial dimensions sample model'.
3. In the tree, expand 'model: Data model Financial dimensions sample model\Journal: Record list'.
4. In the tree, expand 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list'.
5. In the tree, expand 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list'.
6. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Range: Horizontal\Cell'.
7. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list'.

Record list\Dimensions data: Record list\Code: String'.

8. Click Bind.
9. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Range: Horizontal'.
10. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Dimensions data: Record list'.
11. Click Bind.
12. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Cell'.
13. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Credit: Real'.
14. Click Bind.
15. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Cell'.
16. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Debit: Real'.
17. Click Bind.
18. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Cell'.
19. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Currency: String'.
20. Click Bind.
21. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Cell'.
22. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Date: Date'.
23. Click Bind.
24. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Cell'.
25. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list\Voucher: String'.
26. Click Bind.
27. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical\Cell'.
28. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Batch: String'.
29. Click Bind.
30. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Range: Vertical'.
31. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Transaction: Record list'.
32. Click Bind.
33. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical\Cell'.
34. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list\Batch: String'.
35. Click Bind.
36. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Vertical'.
37. In the tree, select 'model: Data model Financial dimensions sample model\Journal: Record list'.
38. Click Bind.
39. In the tree, expand 'model: Data model Financial dimensions sample model\Dimensions setting: Record list'.
40. In the tree, select 'model: Data model Financial dimensions sample model\Dimensions setting: Record list\Code: String'.
41. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Horizontal\Cell'.
42. Click Bind.
43. In the tree, select 'model: Data model Financial dimensions sample model\Dimensions setting: Record list'.
44. In the tree, select 'Excel = "SampleFinDimWsReport"\Range: Horizontal'.
45. Click Bind.

46. In the tree, select 'Excel = "SampleFinDimWsReport"\Cell'.
47. In the tree, select 'model: Data model Financial dimensions sample model\Company: String'.
48. Click Bind.
49. Click Save.
50. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use horizontally expandable ranges to dynamically add columns in Excel reports (Part 2 - Run format)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to generate reports as OPENXML worksheets (Excel) files in which the required columns can be created dynamically as horizontally expandable ranges. These steps can be performed in the DEMF company.

To complete these steps, you must first complete the steps in the "ER Use horizontally expandable ranges to dynamically add columns in Excel reports (Part 1: Design format)" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Find created format

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Financial dimensions sample model'.
3. In the tree, select 'Financial dimensions sample model\Sample report with horizontally expandable ranges'.

## Execute format to create Excel output

1. Click Run.
2. In the Dimension name field, type 'BusinessUnit;CostCenter;Department'.
  - In the Dimension name field, enter or select a value. To select all dimensions for the current company, enter the following: BusinessUnit;CostCenter;Department;ItemGroup;MainAccount;Project
3. Expand the Records to include section.
4. Click Filter.
5. Select the row for the Ledger journal table and the Journal batch number field.
6. In the Criteria field, type '00057..00058'.
  - 00057..00058
7. Click OK.
8. Click OK.
  - Review the generated output. Note that the newly created Excel file contains the same number of columns that were selected for financial dimensions. The report header in those columns represents financial dimensions' names. The transactions' lines in those columns represent financial dimensions. Run this report and select different dimensions to see that the report is not dependent on the number of selected dimensions or the number of dimensions configured for this instance.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use Document Management files in format outputs (Part 1 - Prepare data model)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to use Document Management files (attachments) in ER output. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the "Create a configuration provider and mark it as active" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Get access to the list of configurations provided by Microsoft

1. Go to Organization administration > Workspaces > Electronic reporting.

Make sure that the 'Litware, Inc.' provider is available and marked as active.

2. Select the 'Litware, Inc.' provider.
3. Click Repositories.

If a repository of the 'Operations resources' type already exists, skip the remaining steps of the current sub-task.

4. Click Add to open the drop dialog.
5. In the Configuration repository type field, enter 'Operations resources'.
6. Click Create repository.
7. Click OK.

## Get the Customer invoice model configurations provided by Microsoft

1. Click Show filters.
2. Apply the following filters: Enter a filter value of "Operations resources" on the "Name" field using the "begins with" filter operator; Enter a filter value of "" on the "Description" field using the "begins with" filter operator

3. Click Show filters.
4. Click Open.
5. In the tree, select 'Customer invoice model'.

Select the model configuration 'Customer invoice model' to import it.

6. Click Import.  
Click Import for version 1 of the selected configuration.
7. Click Yes.

8. Close the page.
9. Close the page.
10. Click Reporting configurations.
11. In the tree, select 'Customer invoice model'.

## Create the derived model to support access to the Document Management files.

You will create our own configuration of the Customer invoice model deriving it from the configuration provided by Microsoft. You will use this configuration to implement access to the Document Management files and make them available for electronic documents that you will create based on this model.

1. Click Create configuration to open the drop dialog.
2. In the New field, enter 'Derive from Name: Customer invoice model, Microsoft'.
3. In the Name field, type 'Customer invoice model (custom)'.
4. Click Create configuration.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use Document Management files in format outputs (Part 2 - Extend data model)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the System Administrator or Electronic Reporting Developer role can configure an Electronic reporting (ER) format to use Document Management files (attachments) in ER output. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the "ER Use Document Management files in format outputs (Part 1: Prepare data model)" task guide.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Extend data model to present the Document Management files in it

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Reporting configurations.
3. In the tree, expand 'Customer invoice model'.
4. In the tree, select 'Customer invoice model\Customer invoice model (custom)'.
5. Click Designer.
6. In the tree, select 'Customer invoice(InvoiceCustomer)'.
  - We will extend this data model to expose in it any files that have been attached to a sales order that is related to an electronically processing invoice.
7. Click New to open the drop dialog.
8. In the Name field, type 'Invoice attachments'.
  - Invoice attachments
9. In the Item type field, select 'Record list'.
10. Click Add.
11. Click New to open the drop dialog.
12. In the Name field, type 'File content'.
  - File content
13. In the Item type field, select 'Container'.
14. Click Add.
15. Click New to open the drop dialog.
16. In the Name field, type 'File name'.
  - File name
17. In the Item type field, select 'String'.
18. Click Add.

## Map new data model elements to data sources

1. Click Map model to datasource.
2. Use the Quick Filter to filter on the Definition field with a value of 'InvoiceCustomer'.
  - InvoiceCustomer
  - We will map new model elements to appropriate data sources.
3. Click Designer.

4. In the tree, select 'Invoice attachments'.
5. In the tree, expand 'Invoice attachments'.
6. In the tree, select 'Invoice attachments\File name'.
7. Click Edit.
8. In the Formula field, enter  
'CustInvoiceJour.'>Relations'.SalesTable.'<Relations'.'<Documents'.'originalFileName()'.
  - CustInvoiceJour.'>Relations'.SalesTable.'<Relations'.'<Documents'.'originalFileName()'
9. Click Save.
10. Close the page.
11. In the tree, select 'Invoice attachments\File content'.
12. Click Edit.
13. In the Formula field, enter  
'CustInvoiceJour.'>Relations'.SalesTable.'<Relations'.'<Documents'.'getFileContentAsContainer()'.
  - CustInvoiceJour.'>Relations'.SalesTable.'<Relations'.'<Documents'.'getFileContentAsContainer()'
14. Click Save.
15. Close the page.
16. In the tree, select 'Invoice attachments'.
17. Click Edit.
18. In the Formula field, enter 'CustInvoiceJour.'>Relations'.SalesTable.'<Relations'.'<Documents''.
  - CustInvoiceJour.'>Relations'.SalesTable.'<Relations'.'<Documents'
19. Click Save.
20. Close the page.
21. Click Save.
22. Close the page.
23. Close the page.
24. Close the page.
25. Click Change status.
26. Click Complete.
27. Click OK.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# ER Use Document Management files in format outputs (Part 3 - Create format)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to use Document Management files (attachments) in ER output. These steps can be performed in any company.

To complete these steps, you must first complete the steps in the "ER Use Document Management files in format outputs (Part 2: Extend data model)" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Create a format to process invoices

1. Go to Organization administration > Workspaces > Electronic reporting.
2. Click Reporting configurations.
3. In the tree, expand 'Customer invoice model'.
4. In the tree, select 'Customer invoice model\Customer invoice model (custom)'.
  - You will create a format to generate electronic messages with information about any files that have been attached to a sales order that is related to an electronically processing invoice.
5. Click Create configuration to open the drop dialog.
6. In the New field, enter 'Format based on data model Customer invoice model (custom)'.
7. In the Name field, type 'Electronic invoice sample message'.
  - Electronic invoice sample message
8. In the Data model definition field, enter or select a value.
  - InvoiceCustomer
9. Click Create configuration.

## Design a format to populate attachments into generating a message in MIME format

1. Click Designer.
2. Click Add root to open the drop dialog.
3. In the tree, select 'XML\Element'.
4. In the Name field, type 'Invoice'.
  - Invoice
5. Click OK.
6. Click Add to open the drop dialog.
7. In the tree, select 'XML\Attribute'.
8. In the Name field, type 'SalesOrder'.
  - SalesOrder
9. Click OK.
10. Click Add Attribute.
11. In the Name field, type 'InvoiceNumber'.
  - InvoiceNumber

12. Click OK.
13. Click Add Attribute.
14. In the Name field, type 'InvoiceAmount'.
  - InvoiceAmount
15. Click OK.
16. Click Add to open the drop dialog.
17. In the tree, select 'XML\Element'.
18. In the Name field, type 'EnclosedDocs'.
  - EnclosedDocs
19. Click OK.
20. In the tree, select 'Invoice\EnclosedDocs'.
21. Click Add Element.
22. In the Name field, type 'Document'.
  - Document
23. Click OK.
24. In the tree, select 'Invoice\EnclosedDocs\Document'.
25. Click Add to open the drop dialog.
26. In the tree, select 'XML\Attribute'.
27. In the Name field, type 'FileName'.
  - FileName
28. Click OK.
29. Click Add to open the drop dialog.
30. In the tree, select 'XML\Element'.
31. In the Name field, type 'FileContent'.
  - FileContent
32. Click OK.
33. In the tree, select 'Invoice\EnclosedDocs\Document\FileContent'.
34. Click Add to open the drop dialog.
35. In the tree, select 'Text\Base64'.
36. Click OK.

## Map format elements to data model as data source

1. In the tree, select 'Invoice\SalesOrder'.
2. Click the Mapping tab.
3. In the tree, expand 'model'.
4. In the tree, select 'model\Sales order number(SalesId)'.
5. Click Bind.
6. In the tree, select 'Invoice\InvoiceNumber'.
7. In the tree, expand 'model\Base invoice(InvoiceBase)'.
8. In the tree, select 'model\Base invoice(InvoiceBase)\Invoice number(Id)'.
9. Click Bind.
10. In the tree, select 'Invoice\InvoiceAmount'.
11. In the tree, select 'model\Base invoice(InvoiceBase)\Invoice amount(Amount)'.
12. Click Bind.
13. In the tree, expand 'model\Invoice attachments'.
14. In the tree, select 'model\Invoice attachments\File content'.

15. In the tree, select 'Invoice\EnclosedDocs\Document\FileContent\Base64'.
16. Click Bind.
17. In the tree, select 'model\Invoice attachments\File name'.
18. In the tree, select 'Invoice\EnclosedDocs\Document\FileName'.
19. Click Bind.
20. In the tree, select 'model\Invoice attachments'.
21. In the tree, select 'Invoice\EnclosedDocs\Document'.
22. Click Bind.
23. Click Save.
24. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use Document Management files in format outputs (Part 4 - Run format)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to use Document Management files (attachments) in ER output. These steps can be performed in the DEMF company.

To complete these steps, you must first complete the steps in the "ER Use Document Management files in format outputs (Part 3: Create format)" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Add necessary attachments for sales order of a single invoice

1. Go to Accounts receivable > Invoices > Open customer invoices.
2. Use the Quick Filter to find records. For example, filter on the Invoice field with a value of 'CIV-000148'.
  - CIV-000148
3. Click to follow the selected invoice's link.
  - CIV-000148
4. Click to follow the link in the Sales order field.
  - 000148
5. In the Lines or header field, select the option of Header.
  - Select Header to indicate that this will be the target for adding attachments.
6. Click Attach.
  - Add a few files as attachments for this sales order. Use the files of the document types that are supported by the Document Management (with file extensions DOCX, DPF, XML, JPG, etc.). Browse and select files to be attached and further processed with the related invoice in the ER electronic message.
7. Click New.
8. Click File.
9. Click New.
10. Click File.
11. Close the page.
12. Close the page.
13. Close the page.
14. Close the page.

## Run the designed report for the selected invoice

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Customer invoice model'.
3. In the tree, expand 'Customer invoice model\Customer invoice model (custom)'.
4. In the tree, select 'Customer invoice model\Customer invoice model (custom)\Electronic invoice sample message'.
5. Click Run.
6. Expand the Records to include () section.

7. Click Filter.
8. Select the row of the Customer invoice journal and the Sales order field.
9. In the Criteria field, type '000148'.
  - In the criteria "Sales order" field, type the order number 000148.
10. Click OK.
11. Click OK.
  - Review the generated output. Note that for each attachment a single XML node has been created. The attachment's content is populated to the XML output in MIME (base64) text format.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Use Document Management files in format outputs (Part 5 - Modify and run format)

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user assigned to the system administrator or electronic reporting developer role can configure an Electronic reporting (ER) format to use Document Management files (attachments) in ER output. These steps can be performed in the DEMF company.

To complete these steps, you must first complete the steps in the "ER Use Document Management files in format outputs (Part 4: Run format)" procedure.

This procedure is for a feature that was added in Dynamics 365 for Operations version 1611.

## Modify the format to populate attachments into generating messages in binary format

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Customer invoice model'.
3. In the tree, expand 'Customer invoice model\Customer invoice model (custom)'.
4. In the tree, select 'Customer invoice model\Customer invoice model (custom)\Electronic invoice sample message'.
5. Click Designer.
  - You will populate the invoice message in the generating output as an XML file using UNICODE encoding.
6. Click Add root to open the drop dialog.
7. In the tree, select 'Common\File'.
8. In the Name field, type 'Xml message'.
  - Xml message
9. In the Encoding field, type 'UTF-8'.
  - UTF-8
10. Click OK.
  - Configure the generating output as a zipped file.
11. Click Add root to open the drop dialog.
12. In the tree, select 'Common\Folder'.
13. In the Name field, type 'Zip output'.
  - Zip output
14. Click OK.
15. In the tree, select 'Zip output'.
  - Add attachments to the generating zipped file as files with original names and extensions.
16. Click Add to open the drop dialog.
17. In the tree, select 'Common\File'.
18. In the Name field, type 'Attached file'.
  - Attached file
19. Click OK.
20. In the tree, select 'Zip output\Attached file'.

21. Click Add to open the drop dialog.
22. In the tree, select 'Text\Base64'.
23. Click OK.

## Map new format elements to data model

1. Click the Mapping tab.
2. In the tree, expand 'model'.
3. In the tree, expand 'model\Invoice attachments'.
4. In the tree, select 'Zip output\Attached file\Base64'.
5. In the tree, select 'model\Invoice attachments\File content'.
6. Click Bind.
7. In the tree, select 'Zip output\Attached file'.
8. Click Edit filename.
9. In the tree, expand 'model'.
10. In the tree, expand 'model\Invoice attachments'.
11. In the tree, select 'model\Invoice attachments\File name'.
12. Click Add data source.
13. Click Save.
14. Close the page.
15. In the tree, select 'model\Invoice attachments'.
16. Click Bind.
17. Click Save.
18. Close the page.

## Run the designed report for the selected invoice

1. Click Run.
2. Expand the Records to include () section.
3. Click Filter.
4. Select the row of the Customer invoice journal and the Sales order field.
5. In the Criteria field, In the criteria "Sales order" field, type the order number 000148.
  - 000148
6. Click OK.
7. Click OK.
  - Review the generated output. Note,that in addition to the invoice message in XML format, a single file has been created for each attachment. The attachment files are populated with the zipped output in binary format.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Generate electronic documents and update application data by using ER

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can design Electronic reporting (ER) formats that can be used in the application to generate outgoing electronic documents. You can also design ER formats that parse incoming electronic documents and use the content in those documents to update application data.

With this functionality, a single ER format can be used to generate outgoing electronic documents and then update the application data. This feature can be used in the following scenarios:

- To prevent repeated usage of application data in subsequent processes you can mark an application's data immediately after it is used to generate electronic documents. For example, you can mark payment transactions as already processed immediately after they have been included in a generated payment message.
- To store the processing details of electronic documents that have been generated using ER logic. For example, a unique payment message identification that is generated using the ER expression. The expression is based on information entered in the ER dialog box when the ER format is run to generate documents.

To learn more about this feature, play the set of ER Generate documents with application data update Task guides (part of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process), which walk you through the details of Intrastat reporting and archiving. The following files are required to complete certain steps in these Task guides. Download and save these files to your local machine.

- [ER data model configuration: Intrastat \(model\)](#)
- [ER model mapping configuration: Intrastat \(mapping\)](#)
- [ER format configuration: Intrastat \(format\)](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Import configurations to generate documents that have application data

2/18/2021 • 4 minutes to read • [Edit Online](#)

To complete the steps in this procedure, you must first complete the procedure, "ER Create a configuration provider and mark it as active".

The steps in this procedure explain how to design Electronic reporting (ER) configurations to generate an electronic document. In this procedure, you will import the required ER configurations that have been created for the sample company, Litware, Inc. and use them to generate electronic documents. This procedure is created for users with the assigned role of system administrator or electronic reporting developer. These steps can be completed using the DEMF dataset. Before you begin, download and save the files listed in the Help topic, "Generate electronic documents and update application data with ER tool" (generate-electronic-documents-update-application-data/). The files are Intrastat (model).xml, Intrastat (mapping).xml, and Intrastat (format).xml.

1. Go to Organization administration > Workspaces > Electronic reporting.
  - Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as Active. If you don't see this configuration provider, complete the steps in the procedure, Create a configuration provider and mark it as active.
  - The steps in this procedure show how to use ER capabilities to complete an application data update and how to generate an Intrastat report. The details of the reporting process are archived in the application tables. Currently, when the Intrastat reporting process is activated from the Intrastat form, archiving is done based on the logic programmed in the existing source code. In this procedure, you will configure a similar yet simplified logic of application data using only the ER framework. No changes will be made to the source code.

## Import ER configurations

1. Click Reporting configurations.
2. Click Exchange.
3. Click Load from XML file.
  - Import the ER model configuration that contains the data model that is designed to be used as the data source for generating the Intrastat report. Later, you will extend this data model definition to use it for an application data update to archive details of the Intrastat reporting process.
  - Click Browse and select the Intrastat (model).xml file.
4. Click OK.
5. In the tree, select 'Intrastat (model)'.
6. Click Designer.
7. In the tree, expand 'For outgoing document'.
8. In the tree, expand 'For outgoing document\Transactions'.
  - Review the structure of the imported data model. Note that the root item 'For outgoing document' is defined to specify the data flow for getting data from the application and using it as data source to generate the Intrastat report. The 'Transactions (Record list)' is used to represent the list of Intrastat transactions that must be reported. Because you will archive reported commodity codes, the unique identifier of a single commodity code 'Commodity rec id (Int64)' is needed in this data flow.
9. Close the page.
10. Click Exchange.

11. Click Load from XML file.
  - Import the ER mapping configuration that specifies the data flow for getting data from the application and then using it to generate the Intrastat report. Later, you will extend this model mapping definition to get data from the Intrastat report and use it for the application data update to archive details of Intrastat reporting process.
  - Click Browse and select the Intrastat (mapping).xml file.
12. Click OK.
13. In the tree, expand 'Intrastat (model)'.
14. In the tree, select 'Intrastat (model)\Intrastat (mapping)'.
15. Click Designer.
  - Note that the current model mapping contains the value 'To model' in the Direction field. This means that this model mapping has been designed for getting data from the application and storing it in the data model.
16. Click Designer.
17. In the tree, expand 'List'.
18. In the tree, expand 'Transactions= List'.
  - Review the structure of the model mapping that uses the data model that is filtered based on the root item, 'For outgoing document.' Note that the added data source, 'List' provides access to the required application data, which is the list of records from the Intrastat table.
19. Close the page.
20. Close the page.
21. Click Exchange.
22. Click Load from XML file.
  - Import the ER format configuration that specifies the layout of the Intrastat report and the process of populating data to the report. Later, you will extend this format definition to put data from the Intrastat report in to the data model and then use it to update application data to archive the details of Intrastat reporting process.
  - Click Browse and select the Intrastat (format).xml file.
23. Click OK.
24. In the tree, select 'Intrastat (model)\Intrastat (format)'.
25. Click Designer.
26. Click Expand/collapse.
27. In the tree, select 'File\Declaration'.
28. Click the Mapping tab.
29. In the tree, select 'File'.
  - Review the structure of the format used to generate the Intrastat report. Note that it is designed to generate an XML file by populating data from the data model, which is based on the root item 'For outgoing document'. Verify that the name for generated file is defined on the user dialog form ('fn' data source is used for that).
30. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Design configurations to generate documents that have application data

2/18/2021 • 2 minutes to read • [Edit Online](#)

To complete the steps in this procedure, you must first complete the procedure, ER Generate documents with application data update (Part 1: Import configurations).

The steps in this procedure explain how to design Electronic reporting (ER) configurations to generate an electronic document. In this procedure, you run the ER imported format configuration that has been created for the sample company, Litware, Inc. to generate electronic documents.

This procedure is created for users with the assigned role of system administrator or electronic reporting developer. These steps can be completed using the DEMF dataset.

Before you begin, change the country context for the DEMF company from DEU (Germany) to BEL (Belgium). Click Organization administration > Organizations > Legal entities to update the country code in the primary address of the legal entity DEMF. Restart your application.

## Run imported ER format

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Intrastat (model)'.
3. In the tree, select 'Intrastat (model)\Intrastat (format)'.
4. Click Run.
  - Run the draft version of the ER format configuration to generate the Intrastat report.
5. In the Enter file name field, type 'intrastat.xml'.
  - Specify the name of the file.
6. Click OK.
  - Review the generated XML file.
7. Close the page.
8. Go to Tax > Declarations > Foreign trade > Intrastat.
  - Open this form to view the Intrastat transactions that are included in the generated electronic document.
9. Click Intrastat archive.
  - Because the executed ER format does not contain any settings for application data update, the details of the completed Intrastat report have not been archived.
10. Close the page.
11. Close the page.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Modify models and mappings to generate documents that have application data

2/18/2021 • 5 minutes to read • [Edit Online](#)

To complete the steps in this procedure, you must first complete the procedure, "ER Generate documents with application data update (Part 2: Generate documents)".

The steps in this procedure explain how to design Electronic reporting (ER) configurations to generate an electronic document and update application data. In this procedure, you will modify the ER configurations to start using them to generate electronic documents and update application data. This procedure is created for users with the assigned role of system administrator or electronic reporting developer. These steps can be completed using the DEMF dataset.

## Modify data model

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, select 'Intrastat (model)'.
  - You will extend how you use the data model. Besides using it as data source to generate the Intrastat report, the data model will be used to collect details about the Intrastat reporting process. The details will then be used to update application data.
3. Click Designer.
4. Click New to open the drop dialog.
5. In the New node as a field, enter 'Model root'.
6. In the Name field, type 'For application data update'.
  - For application data update
7. Click Add.
8. In the tree, select 'For application data update'.
  - This new root item is added to specify the data flow for moving data from the Intrastat report (used as a data source) to the application tables (the update destination). Note that different root items must be used for getting data that is posted to the outgoing document and for getting data from the document that is used to update application data.
9. Click New to open the drop dialog.
10. In the Name field, type 'Archive header'.
  - Archive header
11. In the Item type field, select 'Record list'.
12. Click Add.
  - Because you will create a record for each Intrastat report that is generated, you must create a new item for that.
13. Click New to open the drop dialog.
14. In the Name field, type 'File name'.
  - File name
15. In the Item type field, select 'String'.
16. Click Add.
17. Click New to open the drop dialog.
18. In the Name field, type 'Number of lines'.

- Number of lines
19. In the Item type field, select 'Integer'.
  20. Click Add.
    - Add this item to represent the number of Intrastat transactions that are reported during the current reporting process.
  21. Click New to open the drop dialog.
  22. In the Name field, type 'Archive lines'.
    - Archive lines
  23. In the Item type field, select 'Record list'.
  24. Click Add.
    - Add this item to represent the list of Intrastat transactions that are reported during the current reporting process.
  25. Click New to open the drop dialog.
  26. In the Name field, type 'Amount'.
    - Amount
  27. In the Item type field, select 'Real'.
  28. Click Add.
  29. Click New to open the drop dialog.
  30. In the Name field, type 'Commodity rec id'.
    - Commodity rec id
  31. In the Item type field, select 'Int64'.
  32. Click Add.
  33. Click New to open the drop dialog.
  34. In the Name field, type 'Item number'.
    - Item number
  35. In the Item type field, select 'String'.
  36. Click Add.
  37. Click Save.
  38. Close the page.

## Modify model mapping

1. In the tree, expand 'Intrastat (model)'.
2. In the tree, select 'Intrastat (model)\Intrastat (mapping)'.
  - Modify the existing model mapping to start using it for the application data update and to archive Intrastat reporting details.
3. Click Designer.
4. Click New.
5. In the Name field, type 'Update archive'.
  - Update archive
6. In the Direction field, select 'To destination'.
7. Click Save.
  - This new mapping specifies the data flow for moving data (Intrastat reporting details) from the data model to the application tables (the update destination). Note that different model's root items must be used to get data from the application for the reporting process and then use the data from data model for the application data update.
8. Click Designer.
9. In the tree, select 'Data model\Data model'.

- Add the required data source. This is the data model that contains details of the reported Intrastat transactions that must be archived.
10. Click Add root.
  11. In the Name field, type 'model'.
    - model
  12. In the Definition field, enter or select the value 'For application data update'.
    - For application data update
  13. Click OK.
  14. In the tree, expand 'model'.
  15. In the tree, select 'Functions\Calculated field'.
  16. In the tree, select 'model\Archive header'.
  17. Click Add.
    - Because you want to enumerate reported Intrastat transactions for archiving, the appropriate data source must be added.
  18. In the Name field, type 'Enumerated lines'.
    - Enumerated lines
  19. Click Edit formula.
  20. In the tree, select 'List\ENUMERATE'.
  21. Click Add function.
  22. In the tree, expand 'model'.
  23. In the tree, expand 'model\Archive header'.
  24. In the tree, select 'model\Archive header\Archive lines'.
  25. Click Add data source.
  26. In the Formula field, enter 'ENUMERATE(model.'Archive header'.'Archive lines')'.
    - ENUMERATE(model.'Archive header'.'Archive lines')
  27. Click Save.
  28. Close the page.
  29. Click OK.
  30. Click Add destination.
    - Add application tables as required destinations that require updates to archive details of reported Intrastat transactions.
  31. In the Name field, type 'Archive'.
    - Archive
  32. In the Table name field, type 'IntrastatArchiveGeneral'.
    - IntrastatArchiveGeneral
    - Keep the record action 'Insert' so you can add records during the detail archiving of each Intrastat reporting process.
  33. Select Yes in the Record infolog field.
    - Select Yes to get information about issues with the application data update.
  34. Select Yes in the Skip record action validation field.
    - Select Yes to suppress validation errors about the empty 'Intrastat archive ID' field. This will be done after records are added, based on the sequence number settings that are configured for this table in the Foreign trade parameters form.
  35. Click OK.
    - Bind elements of the added data source (the filtered model based on the selected root item) with elements from the added destination.
  36. In the tree, expand 'Archive'.

37. In the tree, expand 'Archive<Relations'.
38. In the tree, expand 'Archive<Relations\IntrastatArchiveDetail'.
39. In the tree, select 'Archive<Relations\IntrastatArchiveDetail\Amount(AmountMST)'.
40. In the tree, expand 'model\Archive header\Enumerated lines'.
41. In the tree, expand 'model\Archive header\Enumerated lines\Value'.
42. In the tree, select 'model\Archive header\Enumerated lines\Value\Amount'.
43. Click Bind.
44. In the tree, select 'Archive<Relations\IntrastatArchiveDetail\Commodity(IntrastatCommodity)'.
45. In the tree, select 'model\Archive header\Enumerated lines\Value\Commodity rec id'.
46. Click Bind.
47. In the tree, select 'Archive<Relations\IntrastatArchiveDetail\Item number(ItemId)'.
48. In the tree, select 'model\Archive header\Enumerated lines\Value\Item number'.
49. Click Bind.
50. In the tree, select 'Archive<Relations\IntrastatArchiveDetail\Line number(LineNumber)'.
51. In the tree, select 'model\Archive header\Enumerated lines\Number'.
52. Click Bind.
53. In the tree, select 'Archive<Relations\IntrastatArchiveDetail'.
54. In the tree, select 'model\Archive header\Enumerated lines'.
55. Click Bind.
56. In the tree, select 'Archive\File name(FileName)'.
57. In the tree, select 'model\Archive header\File name'.
58. Click Bind.
59. In the tree, select 'Archive\Number of lines(NumberOfLines)'.
60. In the tree, select 'model\Archive header\Number of lines'.
61. Click Bind.
62. In the tree, select 'Archive'.
63. In the tree, select 'model\Archive header'.
64. Click Bind.
65. Click Save.
66. Close the page.
67. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Modify formats to generate documents that have application data

2/18/2021 • 5 minutes to read • [Edit Online](#)

To complete the steps in this procedure, you must first complete the procedure, "ER Generate documents with application data update (Part 3: Modify model and mapping)".

The steps in this procedure explain how to design Electronic reporting (ER) configurations to generate an electronic document and update application data. In this procedure, you will modify the ER configurations to not just use them to generate electronic documents, but also to update application data. This procedure is created for users with the assigned role of system administrator or electronic reporting developer. These steps can be completed using the DEMF dataset.

## Modify format to collect details of reporting

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Intrastat (model)'.
3. In the tree, select 'Intrastat (model)\Intrastat (format)'.
4. Click Designer.
5. In the tree, expand 'File'.
6. In the tree, expand 'File\Declaration'.
7. In the tree, select 'File\Declaration\Data'.
8. In the Multiplicity field, select 'One many'.
  - Configure this format element to archive details of the Intrastat reporting process. This item represents the archive's header record.
9. In the tree, expand 'File\Declaration\Data'.
10. In the tree, select 'File\Declaration\Data\Item'.
11. In the Multiplicity field, select 'Zero many'.
  - Configure this format element to archive details of the Intrastat reporting process. This item will represent the list of archived lines.
12. In the tree, expand 'File\Declaration\Data\Item'.
13. In the tree, select 'File\Declaration\Data\Item\Dim1'.
14. Select Yes in the Excluded field.
  - You will not archive this data, so you can exclude this format element from the data source of Intrastat reporting details.
15. In the tree, expand 'File\Declaration\Data\Item\Dim1'.
16. In the tree, select 'File\Declaration\Data\Item\Dim1\property'.
17. Select Yes in the Excluded field.
18. In the tree, select 'File\Declaration\Data\Item\Dim1\date'.
19. Select Yes in the Excluded field.
20. In the tree, select 'File\Declaration\Data\Item\Dim2'.
21. Select Yes in the Excluded field.
22. In the tree, expand 'File\Declaration\Data\Item\Dim2'.
23. In the tree, select 'File\Declaration\Data\Item\Dim2\property'.
24. Select Yes in the Excluded field.



25. In the tree, select 'File\Declaration\Data\Item\Dim2\code'.
26. Select Yes in the Excluded field.
27. In the tree, select 'File\Declaration\Data\Item\Dim3'.
  - Several format elements can have the same name. For example, Dim. You cannot explicitly recognize them when you use this format as a data source for archiving Intrastat reporting details, so you need to define the alternative names for these format elements.
28. In the Name field, type 'Amount'.
  - Amount
29. In the Multiplicity field, select 'Exactly one'.
30. In the tree, select 'File\Declaration\Data\Item\Dim4'.
31. In the Name field, type 'Item'.
  - Item
32. In the Multiplicity field, select 'Exactly one'.
  - In addition to the design format elements, the following Intrastat reporting details must be archived: unique record identification of each reported commodity item and name of the generated file. Because this data will not be populated in the Intrastat report, you need to add the format that is related to these detail elements as data source items.
33. In the tree, select 'File\Declaration\Data'.
34. Click Add to open the drop dialog.
35. In the tree, select 'Data source\Item'.
36. In the Name field, type 'File name'.
  - File name
37. In the Data type field, select 'String'.
38. Click OK.
39. In the tree, select 'File\Declaration\Data\Item'.
40. Click Add Item.
41. In the Name field, type 'Commodity rec ID'.
  - Commodity rec ID
42. In the Data type field, select 'Int64'.
43. Click OK.
44. Click the Mapping tab.
45. In the tree, select 'File\Declaration\Data\File name'.
46. Click Bind.
47. In the tree, expand 'model'.
48. In the tree, expand 'model\Transactions'.
49. In the tree, select 'File\Declaration\Data\Item = model.Transactions\Commodity rec ID'.
50. In the tree, select 'model\Transactions\Commodity rec ID'.
51. Click Bind.
52. Click Save.

## Modify format to memorize details of reporting

1. Click Map format to model.
2. Click New.
3. In the Definition field, enter or select the 'For application data update' root item.
  - For application data update.
4. In the Name field, type 'Mapping to update data'.
  - Mapping to update data

5. Click Save.

- This mapping defines how the details of the Intrastat report are collected in the data model, the structure of which is specified by the selected root item 'For application data update'. These details, the model mapping with same root item 'For application data update', and the direction 'To destination' will be used for the application data update. The application data update starts immediately after the outgoing Intrastat report is generated. The application data update can be skipped at run-time, but the data model must be empty (containing empty record list).

6. Click Designer.

- The outgoing Intrastat report format is added by default as a data source for this model mapping.
- Bind elements of the designed report (presented as data source) to elements of the data model, which is filtered based on the selected model's root item.

7. In the tree, expand 'Archive header'.

8. In the tree, expand 'Archive header\Archive lines'.

9. In the tree, expand 'format'.

10. In the tree, expand 'format\Declaration: XML Element(Declaration)'.

11. In the tree, expand 'format\Declaration: XML Element(Declaration)\Data: XML Element 1..\* (Data)'.

12. In the tree, expand 'format\Declaration: XML Element(Declaration)\Data: XML Element 1..\* (Data)\Item: XML Element 0..\* (Item)'.

13. In the tree, expand 'format\Declaration: XML Element(Declaration)\Data: XML Element 1..\* (Data)\Item: XML Element 0..\* (Item)\Dim3: XML Element 1..1 (Amount)'.

14. In the tree, expand 'format\Declaration: XML Element(Declaration)\Data: XML Element 1..\* (Data)\Item: XML Element 0..\* (Item)\Dim4: XML Element 1..1 (Item)'.

15. In the tree, select 'Archive header\Number of lines'.

16. Click Edit.

17. In the tree, select 'List\COUNT'.

18. Click Add function.

19. In the tree, expand 'format'.

20. In the tree, expand 'format\Declaration: XML Element(Declaration)'.

21. In the tree, expand `format\Declaration: XML Element(Declaration)\Data: XML Element 1..* (Data)`.

22. In the tree, select

`format\Declaration: XML Element(Declaration)\Data: XML Element 1..* (Data)\Item: XML Element 0..* (Item)`.

23. Click Add data source.

24. In the Formula field, enter 'COUNT(format.Declaration.Data.Item)'.

- COUNT(format.Declaration.Data.Item)

25. Click Save.

26. Close the page.

27. In the tree, select 'Archive header\File name'.

28. In the tree, select 'format\Declaration: XML Element(Declaration)\Data: XML Element 1..\* (Data)\File name: Item String(File name)'.

29. Click Bind.

30. In the tree, select

`format\Declaration: XML Element(Declaration)\Data: XML Element 1..* (Data)\Item: XML Element 0..* (Item)\Dim4: XML Element 1..1 (Item)\number: String(number)`

31. In the tree, select 'Archive header\Archive lines\Item number'.

32. Click Bind.

33. In the tree, select

`format\Declaration: XML Element(Declaration)\Data: XML Element 1..* (Data)\Item: XML Element 0..* (Item)\Dim3: XML Element 1..1 (Amount)\value: Numeric Real(value)`

34. In the tree, select 'Archive header\Archive lines\Amount'.

35. Click Bind.

36. In the tree, select

```
format\Declaration: XML Element(Declaration)\Data: XML Element 1..* (Data)\Item: XML Element 0..* (Item)\Commodity rec ID: Item Int64(Commodity rec ID)
```

37. In the tree, select 'Archive header\Archive lines\Commodity rec ID'.

38. Click Bind.

39. In the tree, select 'Archive header\Archive lines'.

40. In the tree, select

```
format\Declaration: XML Element(Declaration)\Data: XML Element 1..* (Data)\Item: XML Element 0..* (Item)
```

41. Click Bind.

42. In the tree, select 'Archive header'.

43. In the tree, select `format\Declaration: XML Element(Declaration)\Data: XML Element 1..* (Data)`.

44. Click Bind.

45. Click Save.

46. Close the page.

47. Close the page.

48. Close the page.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Generate documents that have application data

2/18/2021 • 2 minutes to read • [Edit Online](#)

To complete the steps in this procedure, you must first complete the procedure, "ER Generate documents with application data update (Part 4: Modify format)".

The steps in this procedure explain how to design Electronic reporting (ER) configurations to generate an electronic document and update application data. In this procedure, you execute the ER format configuration to generate the Intrastat report and update application data for archiving details of the reporting process.

This procedure is created for users with the assigned role of system administrator or electronic reporting developer. These steps can be completed using the DEMF dataset. Before you begin, make sure that the country context for the DEMF company is BEL (Belgium).

## Set up foreign trade parameters

1. Go to Tax > Setup > Foreign trade > Foreign trade parameters.
2. Click the Number sequences tab.

Archiving details of Intrastat reporting process, we need to identify records of each archive we created. A special number sequence must be configured for that.

3. Select the 'Intrastat archive ID' reference.
4. In the Number sequence code field, type a value.

In the 'Number sequence code' field, enter or select the value 'Fore\_2'.

5. ResolveChanges the Number sequence code.
6. Click Save.
7. Close the page.

## Run modified ER format

1. Go to Organization administration > Electronic reporting > Configurations.
2. In the tree, expand 'Intrastat (model)'.
3. In the tree, select 'Intrastat (model)\Intrastat (format)'.
4. Click Run.
5. In the Enter file name field, type 'intrastat2.xml'.
6. Click OK.

## Review ER format execution's results

Review the generated XML file.

1. Close the page.
2. Go to Tax > Declarations > Foreign trade > Intrastat.

Open this form containing Intrastat transactions that have been included to the generated electronic document.

3. Click Intrastat archive.

Since the executed ER format contains now settings for application data update, the details of the completed Intrastat reporting have been archived. In this form, you can see the header record of the created archive.

4. Click Details.

In this form, you can see the details for the created archive.

5. Close the page.

6. Close the page.

7. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Parse incoming documents

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic covers the following three tasks:

- [Parse incoming documents to update application data](#)
- [Parse incoming documents in Excel format](#)
- [Parse incoming documents in CSV format](#)

## Parse incoming documents to update application data

You can design Electronic reporting (ER) formats and run them in the application to parse incoming electronic documents and then use their content to update application data.

The following new ER functionality that has been introduced improves the parsing of incoming electronic documents in XML format:

- The **CASE** format element can be used as a root element of the ER format that is configured to parse incoming electronic documents in XML format. The **FILE** format element is supported as a nested element of the **CASE** element. Therefore, you can configure a single ER format to parse incoming electronic documents that might contain different root XML elements.
- A **Parsing order of nested elements** attribute has been introduced for XML format elements in ER formats. You can use this attribute to define a single XML element that is expected in the incoming file. There are two valid sequences of the nested elements:
  - **As in format** – The incoming file is valid when the sequence of nested elements in the file is the same as the order that is described in the ER format.
  - **Any** – The incoming file is valid when all nested elements in the ER format are present in the parsing file, regardless of their sequence in that file.

To become more familiar with the details of this feature, play the task guide, ER - Parse incoming documents to update application data (part of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process). This task guide shows how the responses from a web service can be parsed by using an ER format.

To complete some steps of the task guide, you must download the following files:

CONTENT DESCRIPTION	FILE
ER data model configuration	<a href="#">EFSTAmodel.xml</a>
ER format configuration	<a href="#">EFSTAformat.xml</a>
Web service response sample 1	<a href="#">Response1.xml</a>
Web service response sample 2	<a href="#">Response2.xml</a>
Web service response sample 3	<a href="#">Response3.xml</a>
Web service response sample 4	<a href="#">Response4.xml</a>

## Parse incoming documents in Excel format

You can design Electronic reporting (ER) formats to parse incoming Microsoft Excel files that represent data in Microsoft Excel workbooks (files in XLSX format). You can then use the content from these files to update application data. This is useful if you:

- Design a new model and format and want to test them at run-time. In this case, Excel will simulate the actual application data.
- Manage data beyond your application in Excel and want to import this data to submit a specific report.

To learn more about this feature, play the task guides **ER Import data from a Microsoft Excel file (Part 1: Design format)** and **ER Import data from a Microsoft Excel file (Part 2: Import data)** (parts of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process). These task guides walk through how the incoming Excel file can be parsed by using the ER format to import information from incoming documents and update application data. You can download the task guide files from the [Microsoft Download Center](#).

Download the following files to complete the task guides mentioned above.

CONTENT DESCRIPTION	FILE
Incoming file in .XLSX format - template	<a href="#">1099import-template.xlsx</a>
Incoming file in .XLSX format - sample data	<a href="#">1099import-data.xlsx</a>

If you have not yet played the following task guide, [ER Create required configurations to import data from an external file](#) in the current Finance and Operations application, download the following file.

CONTENT DESCRIPTION	FILE
ER model configuration	<a href="#">1099model.xml</a>

## Parse incoming documents in CSV format

You can design Electronic reporting (ER) formats to parse incoming electronic documents that represent tabular data in plain text (files in CSV format) and then use the content from these documents to update application data. The following approach can be used:

- Begin your format's design by adding a new root sequence element to specify that each line in the parsing file is considered a separate record.
  - In the added sequence element, select the appropriate value, for example **New line - Windows (CR LF)**, in the **Special characters** field in the **Sequence element delimiter** field group.
- Continue your format's design by adding a nested sequence element of the added root sequence element to specify that each line in the parsing file is considered as a set of fields.
  - You can specify the character in the **Custom delimiter** field that will be recognized in the parsing line as a fields separator.

**NOTE**

- You can define different field separators for different sequence elements to parse specific file lines in which fields are separated by different characters.
- The **Custom delimiter** field can be left blank for certain sequence elements. An empty field means that any file line that is parsed by using this sequence will be parsed like a .txt (fixed length text) file line.

- In the **Quotation application** field, select the value when you expect that some fields of any line that is parsed by this sequence element will be enclosed by certain characters. The following options are available:
  - **All** – Include quotation characters in the parsing line for any field of any data type. If you select this, define the desired characters in the **Quotation mark** field that will be used for fields quotation. For example, the double quotes character.
  - **Text only** – Include quotation characters in the parsing line for any field of the **String** data type. If you select this, define the desired characters in the **Quotation mark** field that will be used for fields quotation.
  - **Derive from parent only** – Use the same **Quotation application** field settings that are defined for the parent sequence element. Note that the **Quotation mark** field setting will be taken from the settings of the parent sequence element as well.
  - **None** – Exclude quotation characters in the parsing line for any field of any data type.
- Complete your format's design by adding nested elements for the added sequence element that represents the set of fields of the parsing line. Add the required elements of the **Text** group, such as **String**, **DateTime**, and **Numeric**, to describe the structure of the parsing line as a set of individual fields of different data types.
  - For each format element that represents an individual field of the parsing line, by default, nothing is selected in the **Multiplicity** field. This means that the value of the field in the parsing line is considered required. In the **Multiplicity** field, select **Zero one** to consider the value of this field in the parsing line as optional.

**NOTE**

The data source item **isMatch** is available when you map this format to ER data model for each **String**, **DateTime**, or **Numeric** format element with the option **Zero one** selected in the **Multiplicity** field. When this field contains a value, **isMatch** will return **True**. If there is no value in the field, it will return **False**.

To learn more about this feature, play the task guide, **ER Create a format configuration to import data from an external CSV file** (part of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process), which walks through how the incoming CSV file can be parsed by using the ER format to update application data.

Download the following files to complete the task guide mentioned above.

TITLE	FILE NAME
ER format configuration	<a href="#">1099formatcsv.xml</a>
Sample of incoming file in .csv format	<a href="#">1099entriescsv.csv</a>

Download the following file that is required to complete the task guide mentioned above if you have not played the task guide, **ER Create required configurations to import data from an external file for electronic**



reporting in the current Finance and Operations application.

TITLE	FILE NAME
ER model configuration	<a href="#">1099model.xml</a>

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Create required configurations to import data from an external file

2/18/2021 • 12 minutes to read • [Edit Online](#)

The following steps explain how a user in the System administrator or Electronic reporting developer role can design Electronic reporting (ER) configurations to import data in to the application from an external file. In this example, you will create the required ER configurations for the sample company, Litware, Inc. To complete these steps, you must first complete the steps in the Task guide, "ER Create a configuration provider and mark it as active." These steps can be completed using the USMF data set. You must also download and save the following files locally using links from the Electronic reporting overview topic (<https://go.microsoft.com/fwlink/?linkid=852550>): 1099model.xml, 1099format.xml, 1099entries.xml, 1099entries.xlsx.

ER offers business users the ability to configure the process of importing external data files to tables in either .XML or .TXT format. First, an abstract data model and an ER data model configuration must be designed to represent the data that you are importing. Next, you need to define the structure of the file that you are importing and the method that you will use to port the data from the file to the abstract data model. The ER format configuration that maps to the designed data model must be created for that abstract data model. Then, the data model configuration must be extended with a mapping that describes how the imported data is persisted as abstract data model data and how it is used to update tables. The ER data model configuration must be appended with a new model mapping that describes the binding of the data model to the application's destinations.

The following scenario shows the ER data import capabilities. This includes vendor transactions that are tracked externally and then imported to be reported later in Vendor's settlement for 1099's.

## Add a new ER model configuration

1. Go to Organization administration > Workspaces > Electronic reporting.

Verify that the configuration provider for sample company 'Litware, Inc.' is available and marked as active. If you don't see this configuration provider, you must first complete the steps in the procedure, "Create a configuration provider and mark it as active."

2. Click Reporting configurations.

Instead of creating of a new model to support data import, load the file, 1099model.xml, that you previously downloaded. This file contains the custom data model of vendors' transactions. This data model is mapped to the data components that are in the AOT data entity.

3. Click Exchange.

4. Click Load from XML file.

Click Browse and navigate to the 1099model.xml file that you previously downloaded.

5. Click OK.

6. In the tree, select '1099 Payments model'.

## Review data model settings

1. Click Designer.

This model is designed to represent vendors' transactions from the business standpoint and are separate from the implementation.

2. In the tree, expand '1099-MISC'.
3. In the tree, select '1099-MISC\Transactions'.
4. In the tree, expand '1099-MISC\Transactions'.

The Transactions element of this model represents individual transactions. The child elements are used to specify required details, such as vendor account and transaction date, for each transaction.

5. Close the page.

## Add a new ER format configuration that supports data import

The steps in this subtask show you how a new format configuration can be created to manage data import from external files.

1. Click Create configuration to open the drop dialog.
2. In the New field, enter 'Format based on data model 1099 Payments model'.
3. Select Yes in the Supports data import field.
4. Press ESC key to close this page.

Instead of creating a new format to support data import, load the 1099format.xml file that you previously downloaded. This file contains the defined structure of the file you are importing and the mapping of the structure to the custom data model of vendors' transactions.

5. Click Exchange.
6. Click Load from XML file. Click Browse and navigate to the 1099format.xml file that you previously downloaded.
7. Click OK.
8. In the tree, expand '1099 Payments model'.
9. In the tree, select '1099 Payments model\Format for importing vendors' transactions'.

## Review format settings

1. Click Designer.
2. Toggle 'Show details' on.
3. Click Expand/collapse.
4. Click Expand/collapse.

The designed format represents the expected structure of the external file. This file must be in XML format and have the settlement root element. Each vendor's transaction is represented by the transaction element that is defined as having zero-to-many multiplicity. This means that the incoming file may contain anywhere from zero to multiple transactions. Nested elements of the 'transaction' element represent a single transaction's attributes. Note that all attributes, except country, are marked as mandatory, meaning that it is required to have them in the importing file.

## Review the settings of the format mapping to the data model

1. Click Map format to model.

The mapping 'For importing vendors' transactions' contains the data transfer rules from the incoming XML file to the selected part of the custom data model, which is defined by selecting the1099-MISC definition.

2. Click Designer.
3. Toggle 'Show details' on.
4. In the tree, expand 'format: Record'.
5. In the tree, select 'format: Record'.

Note that the designed format is presented here as a data source component.

6. In the tree, expand `format: Record\*settlement: XML Element 1..1 (settlement): Record`.

7. In the tree, expand

```
format: Record\*settlement: XML Element 1..1 (settlement): Record\transaction: XML Element 0..*
(transaction): Record list
```

8. In the tree, expand

```
format: Record\*settlement: XML Element 1..1 (settlement): Record\transaction: XML Element 0..*
(transaction): Record list\*vendor: XML Element 1..1 (vendor): Record
```

9. In the tree, expand

```
format: Record\*settlement: XML Element 1..1 (settlement): Record\transaction: XML Element 0..*
(transaction): Record list\country: XML Element 0..1 (country): Record
```

10. In the tree, select

```
format: Record\*settlement: XML Element 1..1 (settlement): Record\transaction: XML Element 0..*
(transaction): Record list\*vendor: XML Element 1..1 (vendor): Record
```

Note that the presentation of mandatory and optional format elements is different in the predefined 'format' data source component.

11. In the tree, expand 'Transactions: Record list= format.settlement.'\$enumerated''.

Note that the elements of the format that defines the structure of the imported file are bound to the elements of the custom data model. Based on these bindings, the content of the imported XML file will be stored at run-time in the existing data model. Pay attention to the binding of the country element. For any transaction element in the incoming file that has no such element, the default country code 'USA' will be populated in the data model.

12. Click the Validations tab.

This format mapping may contain user-defined logic to validate the accuracy of the imported data from a business standpoint. For example, based on the setting, for any transaction in the importing file without a defined country code, a warning message will be generated in the Infolog informing the user about the case and indicating the transaction's sequence number.

13. Close the page.

## Run the format mapping to the data model

Execute this format mapping for testing purposes. Use the file 1099entries.xml that you previously downloaded.

You can export this file from the 1099entries.xlsx workbook that is used to manage vendor transactions. The generated output will be imported from the selected XML file and populate the custom data model at real import.

1. Click Run.

Click Browse and navigate to the 1099entries.xml file that you previously downloaded.

2. Click OK.

Note the warning message about a missing country code for a transaction in the imported file.

Review the output in XML format, which represents the data that has been imported from the selected file and ported to the data model.

3. Close the page.
4. Close the page.

## Review the settings for the model mapping to the destinations

1. In the tree, select '1099 Payments model'.
2. Click Designer.
3. Click Map model to datasource.

The mapping For 1099 manual transactions import has been defined with the To destination direction type. This means that it has been entered to support data import and contains the setting of rules defining how the imported external file and persisted as abstract data model data is used to update tables in the application.

4. Click Designer.
5. In the tree, expand 'model: Data model 1099 Payments model'.
6. In the tree, expand 'model: Data model 1099 Payments model\Transactions: Record list'.

Note that the designed model is presented here as a data source element. At runtime, it will contain the data that is imported from the external file. Several tables were added as data source elements to ensure that the imported data is compliant with the data of the current application, including whether the importing transaction vendor account is available in the system, whether the combination of the importing country and state codes exists, etc.

7. In the tree, select 'model: Data model 1099 Payments model\Transactions: Record list\$failed: Calculated field = IF(OR(ISEMPTY(model.Transactions.'\$refs'.vendor), ISEMPTY(model.Transactions.'\$refs'.vendor1099), ISEMPTY(model.Transactions.'\$refs'.box1099), ISEMPTY(model.Transactions.'\$refs'.country), ISEMPTY(model.Transactions.'\$refs'.state), ISEMPTY(model.Transactions.'\$refs'.location)), true, false): Boolean'.
8. Click Edit.
9. Click Edit formula.

When at least one validation fails for a single imported transaction, this transaction will be marked as failed by the data source attribute '\$failed'.

10. Close the page.
11. Click Cancel.

12. In the tree, select 'tax1099trans: Table 'VendSettlementTax1099' records= model.Validated'.

13. Click Edit destination.

This ER destination was added to specify how the imported data will update the application tables. In this case, the data table VendSettlementTax1099 has been selected. Because the record action Insert has been selected, the imported transactions will be inserted in the table VendSettlementTax1099. Note that a single model mapping may contain several destinations. This means that the imported data can be used to update multiple application's tables at once. Tables, views, and data entities can be used as ER destinations.

If the mapping will be called from a point in the application (such as button or menu item) that was specifically designed for this action, the ER destination should be marked as the integration point. In this example this is the ERTableDestination#VendSettlementTax1099 point.

14. Click Cancel.

15. Click Show all.

16. Click Show mapped only.

17. In the tree, expand 'tax1099trans: Table 'VendSettlementTax1099' records= model.Validated'.

Note that the data source element that contains the only validated transactions is bound to the created destination. You can filter the imported transactions to skip the ones that are incompatible with the applications' data.

18. In the tree, select 'failed: Table 'VendSettlementTax1099Entity' records= model.Failed'.

19. Click the Validations tab.

This model mapping may contain user-defined logic to validate the correctness of the imported data from the existing application data. For example, based on the present setting, for any transaction in the imported file with a vendor account that is not in the system, a warning message will be generated informing the user and indicating the incorrect vendor account code.

Note that the Post validation action option can be used for each validation, to specify whether the import process must be continued or stopped, as well as if the already performed inserts/updates can be kept or rolled back.

20. Click Show mapped only.

21. Click Show all.

22. Close the page.

Execute this model mapping to test the designed format and model mappings. Use the file 1099entries.xml. The data from the selected file will be imported in to the system.

23. Click Run.

Note that the dialog box contains no additional questions about the format mapping that must be used to parse the imported file and then port the data to the data model. This is because there is currently only one format that uses this model, which is marked as designed to support data import.

Define the voucher ID to differentiate the imported transactions from other transactions that may already have been entered manually or imported.

24. In the Enter voucher id field, type 'IMPORT-001'.

Browse to get the '1099entries.xml' file.

25. Click OK.

The list of generated warnings provides information about incorrect vendor accounts, an incorrect tax 1099 box code, missing country codes, etc. Compare this list of warnings to the content that is included in the execution XML file.

26. Close the page.

27. Close the page.

28. Close the page.

29. Close the page.

30. Go to Accounts payable > Periodic tasks > Tax 1099 > Vendor settlement for 1099s.

This form shows the cumulative transactions in the Tax1099Summary table that have been created based on imported transactions.

31. In the From date field, set the date to '2000-01-01'.

32. Click Manual 1099 transactions.

This form contains the list of transactions that were added manually and those that we just imported.

33. Open Voucher column filter.

34. Enter a filter value of "IMPORT-001" on the "Voucher" field using the "begins with" filter operator.

## Review the relationship between model and format mappings

1. Close the page.

2. Close the page.

3. Go to Organization administration > Workspaces > Electronic reporting.

4. Click Reporting configurations.

5. In the tree, select '1099 Payments model'.

Assume that you want to support importing the same data but from a .TXT file format.

6. Click Create configuration to open the dialog box.

7. In the New field, enter 'Format based on data model 1099 Payments model'.

8. In the Name field, type 'Import data from TXT file'.

9. Select Yes in the Supports data import field.

10. Click Create configuration.

11. Click Designer.

12. Click Map format to model.

13. Click New.

14. In the Definition field, enter or select a value.

Select '1099-MISC' option.

15. In the Name field, type 'Import data from TXT file'.

16. In the Description field, type 'Import data from TXT file'.

17. Click Save.

18. Close the page.

19. Close the page.

20. Click Edit.

If you installed the hotfix "KB 4012871 Support of GER model mappings in separated configurations with an ability to specify different kinds of prerequisites for deploying them on different versions of Dynamics 365 Finance" ([KB 4012871](#)), execute the next step "Turn the flag 'Default for model mapping' on" for the entered format configuration. Skip the next step otherwise.

21. Select Yes in the Default for model-mapping field.

22. In the tree, select '1099 Payments model'.

23. Click Designer.

24. Click Map model to datasource.

25. Click Run.

If you installed the hotfix, KB 4012871 Support of GER model mappings in separated configurations with an ability to specify different kinds of prerequisites for deploying them on different versions ([KB 4012871](#)), select the preferred model mapping in the lookup field. If you haven't installed the hotfix yet, skip to the next step as the mapping has already been selected by the definition of the default format configuration.

If you have not installed the hotfix, KB 4012871, notice that the dialog box contains an additional model-mapping question that is used to parse the file that you are importing. The data is then ported from the dialog box to the data model. Currently, you can choose which format mapping must be used depending on the type of file that you plan to import.

If you plan to call this model mapping from a point in the application that is specifically designed for the action, the ER destination and the format mapping must be marked as part of the integration.

26. Click Cancel.

27. Close the page.

28. Close the page.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure data import from SharePoint

2/18/2021 • 9 minutes to read • [Edit Online](#)

To import data from an incoming file by using the Electronic reporting (ER) framework, you must configure an ER format that supports the import and then run a model mapping of the **To destination** type that uses that format as a data source. To import data, you must navigate to the file that you want to import. The incoming file can be manually selected by user. With the new ER capability to support importing data from Microsoft SharePoint, this process can be configured as unattended. You can use ER configurations to perform data import from files that are stored in Microsoft SharePoint folders. This topic explains how to complete the import from SharePoint. The examples use vendor transactions as business data.

## Prerequisites

To complete the examples in this topic, you must have the following access:

- Access one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
- Access to the instance of Microsoft SharePoint Server that is configured for use with the application.
- ER format and model configurations for 1099 payments.

### Create required ER configurations

Play the **ER Import data from a Microsoft Excel file** task guides, which are part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process. These task guides walk you through the process of designing and using ER configurations to interactively import vendor transactions from Microsoft Excel files. For more information, see [Parse incoming documents in Excel format](#). After you have completed the task guides, you will have the following setup.

#### ER configurations

- ER model configuration, **1099 Payments model**
- ER format configuration, **Format for importing vendors' transactions from Excel**

The screenshot shows the Dynamics 365 Finance and Operations interface. The main window displays the configuration details for an ER format configuration named "Format for importing vendors' t...". The configuration is set to "Country/region codes" and "Run Draft" is checked. The configuration provider is "Litware, Inc." and "Default for model mapping" is checked. The "Versions" table shows two versions: 6.2 (Draft) and 6.1 (Completed).

R...	Version	Status	Effective from	Version created	Description	Base
	6.2	Draft		4/2/2018 08:25:58 AM		1099 Payment... 6
	6.1	Completed		4/2/2018 08:21:29 AM	Initial version	1099 Payment... 6

#### Sample of the incoming file for data import

- Excel file **1099import-data.xlsx**, with vendor transactions that should be imported.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	US-101	3/25/2018	MISC-01	3,000.00	USA	WA								
2	US-103	4/12/2018	MISC-02	4,000.00	USA	IL								
3	US-105	5/1/2018	MISC-03	450.00	USA	CO								
4														
5														
6														
7														

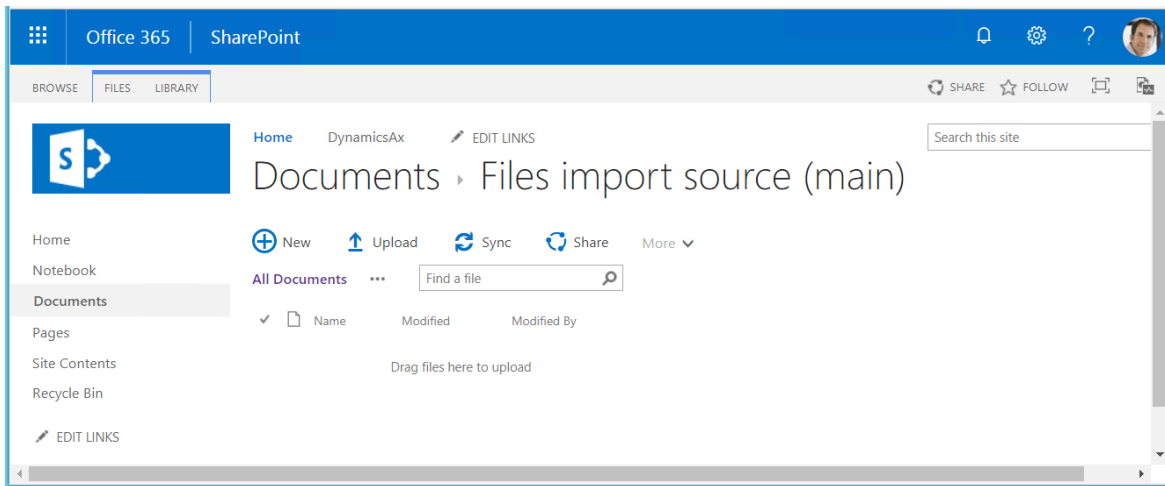
**NOTE**

The format for importing vendor transactions is selected as the default model mapping. Therefore, if you run a model mapping of the **1099 Payments model**, and that model mapping is of the **To destination** type, the model mapping runs this format to import data from external files. It then uses that data to update application tables.

## Configure access to SharePoint for file storage

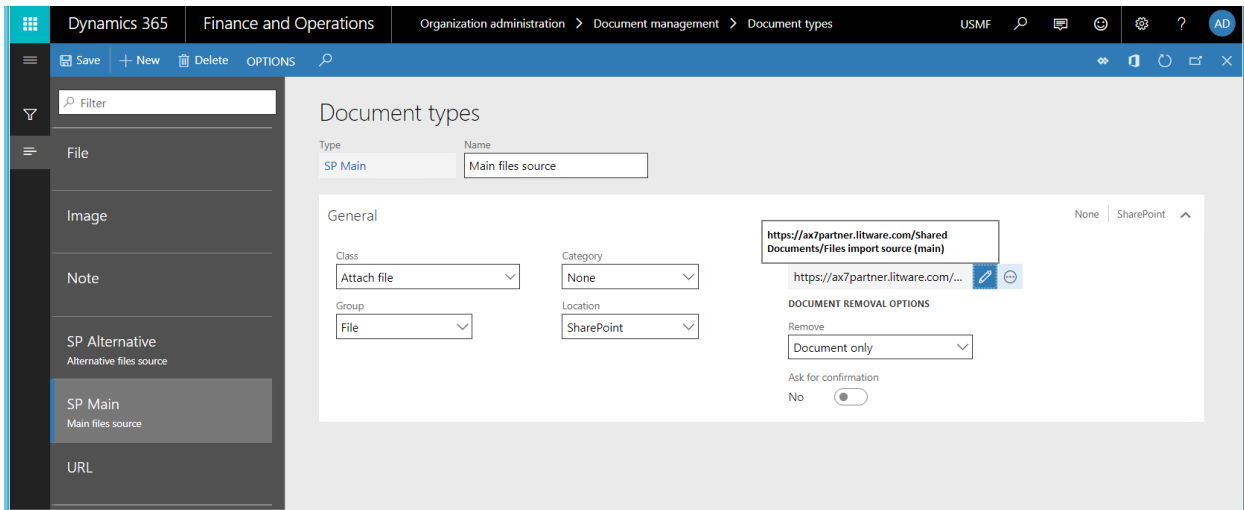
To store electronic report files in a SharePoint location, you must configure access to the SharePoint Server instance that will be used by the current company. In this example, the company is USMF. For instructions, see [Configure SharePoint storage](#).

1. Complete the steps in [Configure SharePoint storage](#).
2. Open the configured SharePoint site.
3. Create the following folders where incoming electronic reporting files can be stored:
  - Files import source (main) (Example shown in screenshot below)
  - Files import source (alternative)



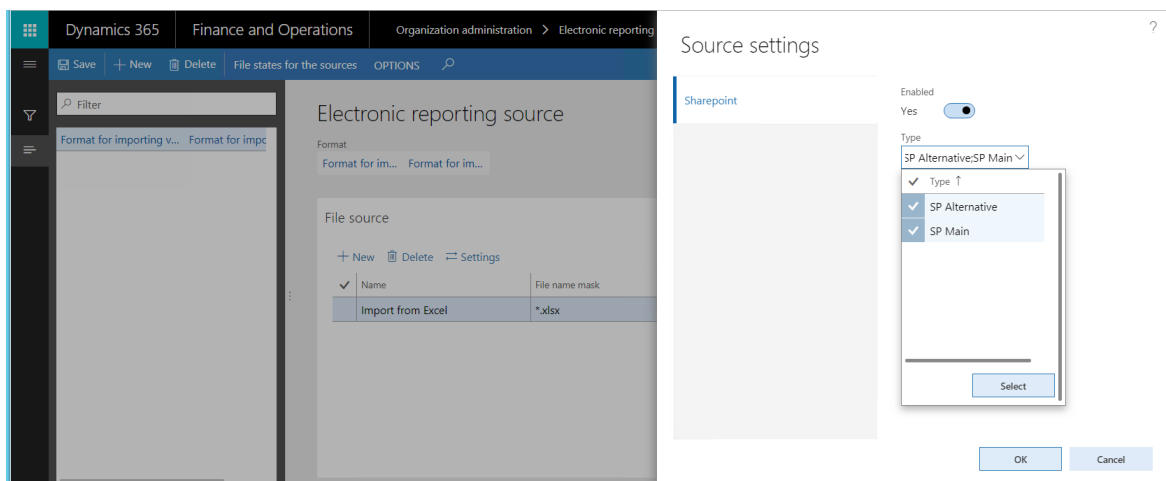
4. (Optional) Create the following folders where the files can be stored after import.
  - Files archive folder - This folder would be for successfully imported files.
  - Files warning folder - This folder would be for files that were imported with a warning.
  - Files error folder - This folder would be for files that failed to import.
5. Go to **Organization administration > Document management > Document types**.
6. Create the following document types that will be used to access the SharePoint folders that you created. For instructions, see [Configure document types](#).

DOCUMENT TYPE	GROUP	LOCATION	SHAREPOINT FOLDER
SP Main	File	SharePoint	Files import source (main)
SP Alternative	File	SharePoint	Files import source (alternative)
SP Archive	File	SharePoint	Files archive folder
SP Warning	File	SharePoint	Files warning folder
SP Error	File	SharePoint	Files error folder



## Configure ER sources for the ER format

1. Click **Organization administration > Electronic reporting > Electronic reporting source**.
2. On the **Electronic reporting source** page, configure the source files for data import by using the configured ER format.
3. Define a file name mask, so that only files with the .xlsx extension are imported. The file name mask is optional and is used only when it has been defined. You can define only one mask for each ER format.
4. Change **Sort files before import** to **Do not sort**, if there are several files for import and the import order is not important
5. Select all SharePoint folders that you created earlier.



## NOTE

- The ER *source* is defined for each application company individually. By contrast, ER *configurations* are shared across companies.
- When you delete an ER source setting for an ER format, all connected file states (see below) are also deleted by confirmation.

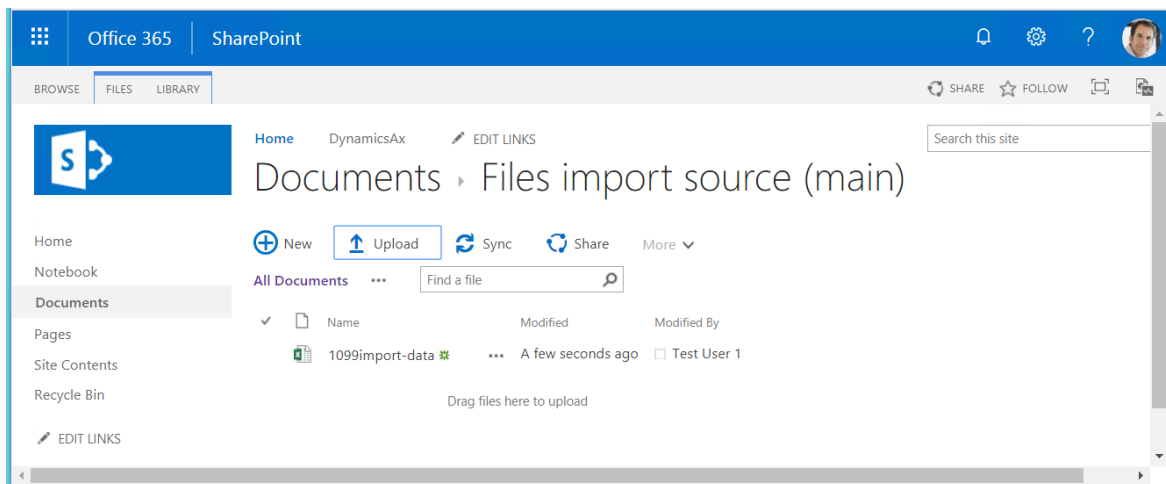
## Review the files states for the ER format

1. On the **Electronic reporting source** page, select **File states for the sources** to review the content of the configured file sources for the current ER format.
2. In the **Files** section, review the list of files. This list presents the following:
  - Source files that are applicable, based on the file name mask (if a file name mask is defined), and that are ready for data import. For these files, the **Sources log for the import format** section is blank.
  - Previously imported files. For each of these files, in the **Sources log for the import format** section, you can review the history of import of this file.

You can also open the **File states for the sources** page by selecting **Organization administration** > **Electronic reporting** > **File states for the sources**. In this case, the page provides information about file sources for all ER formats that file sources have been configured for in the company that you're currently signed in to.

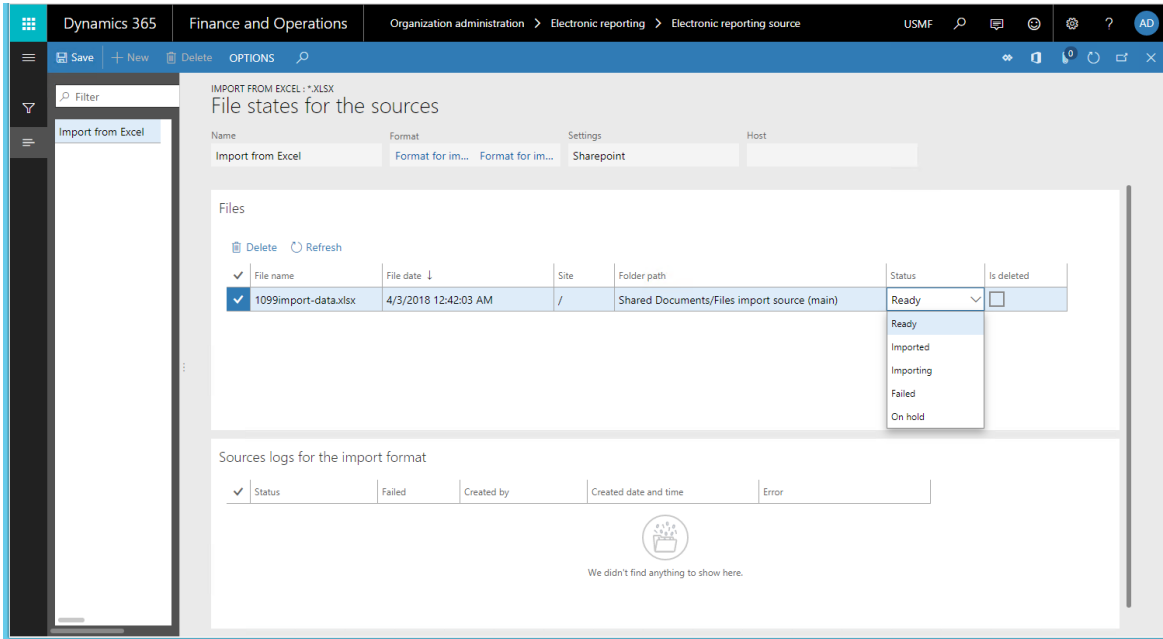
## Import data from Excel files that are in a SharePoint folder

1. In SharePoint, upload the Microsoft Excel file **1099import-data.xlsx** that contains vendor transactions to the **Files import source (main)** SharePoint folder that you created earlier.



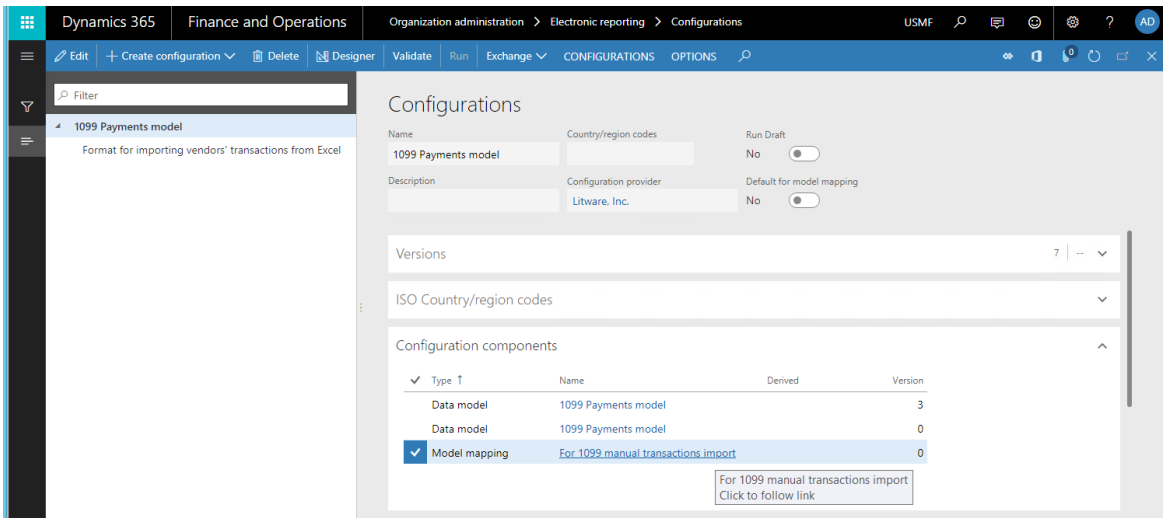
2. On the **File states for the sources** page, select **Refresh** to refresh the page. The Excel file that was uploaded to SharePoint appeared on this page with the status **Ready**. The following statuses are currently supported:
  - **Ready** – Assigned automatically for each new file in a SharePoint folder. This status means that the file is ready for import.
  - **Importing** – Assigned automatically by an ER report when the file will be locked by the import process to prevent its usage by other processes (if many of them are running simultaneously).
  - **Imported** – Assigned automatically by an ER report when the file import is successfully completed. This status means that the imported file has been deleted from the configured files source (SharePoint folder).

- **Failed** – Assigned automatically by an ER report when the file import completed with errors or exceptions.
- **On hold** – Assigned manually by user on this page. This status means that the file will not be imported for now. This status can be used to postpone the import of some files.



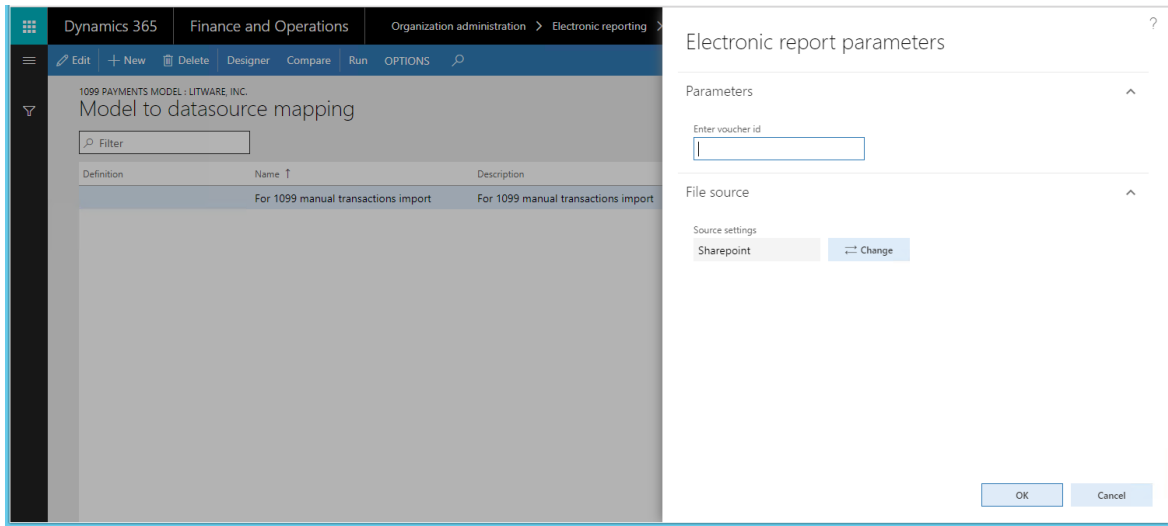
## Import data from SharePoint files

1. Open the ER configurations tree, select the **1099 Payment model**, and expand the list of ER model components.
2. Select the name of the model mapping to open the list of model mappings of the selected ER model configuration.



3. Select **Run** to run the selected model mapping. Because you configured file sources for the ER format, you can change the setting of the **File source** option, if needed. If you keep the setting of this option, the .xlsx files are imported from the configured sources (the SharePoint folders, in this example).

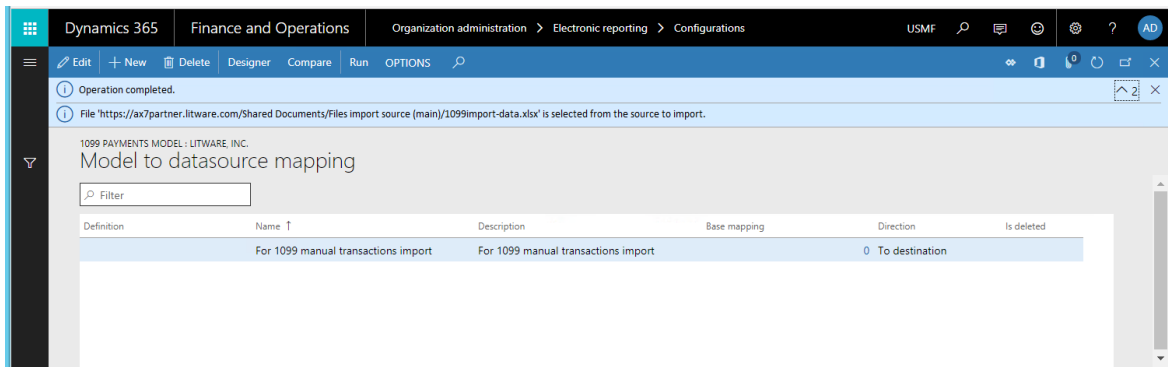
In this example, you're importing only one file. However, if there are multiple files, they are selected for importing in the order in which they were added to the SharePoint folder. Every run of an ER format imports a single selected file.



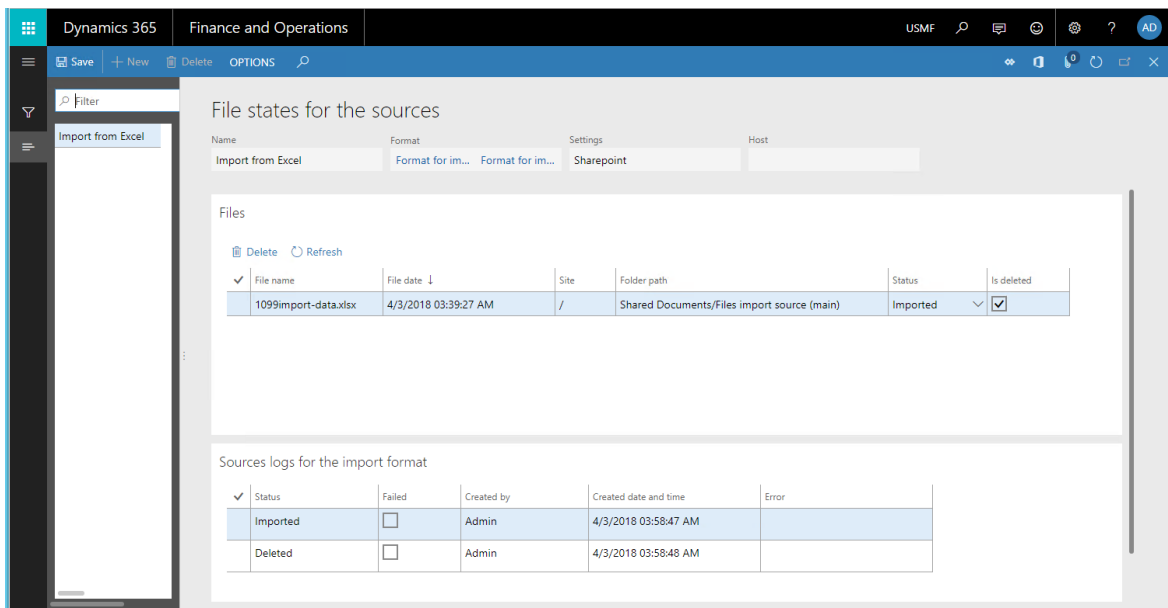
4. The model mapping can run **unattended** in batch mode. In this case, every time that a batch runs this ER format, a single file is imported from the configured file sources.

When a file is successfully imported from the SharePoint folder, it's deleted from that folder and moved to folder for successful imported files or to the folder to imported files with warnings. Otherwise it's moved to folder for failed files or stays in this folder if the folder for failed files isn't set up.

5. Enter the voucher ID, such as **V-00001**, and then select **OK**.



6. On the **File states for the sources** page, select **Refresh** to refresh the page.



7. In the **Files** section, review the list of files. The **Sources log for the import format** section provides the history of the Excel file import. Because this file was successfully imported, it's marked as **Deleted** in the SharePoint folder.

8. Review the **Files import source (main)** SharePoint folder. The Excel files that were successfully imported have been deleted from this folder.
9. Select **Accounts payable > Periodic tasks > Tax 1099 > Vendor settlement for 1099s**.
10. In the **From date** and **To date** fields, enter appropriate values. Then select **Manual 1099 transactions**.

The vendor transactions that were imported from the Excel files on SharePoint for voucher **V-00001**, are presented on the page.

The screenshot shows the Dynamics 365 interface for 'Tax 1099 transactions'. The breadcrumb navigation is 'Accounts payable > Periodic task > Tax 1099 > Vendor settlement for 1099s'. The table below displays the imported transactions.

Vendor account	Date	Voucher	Invoice	Lot ID	Internal invoice	1099 box	Settled federal ...	State
US-101	3/25/2018	V-00001				MISC-01	3,000.00	WA
US-103	4/12/2018	V-00001				MISC-02	4,000.00	IL
US-105	5/1/2018	V-00001				MISC-03	450.00	CO

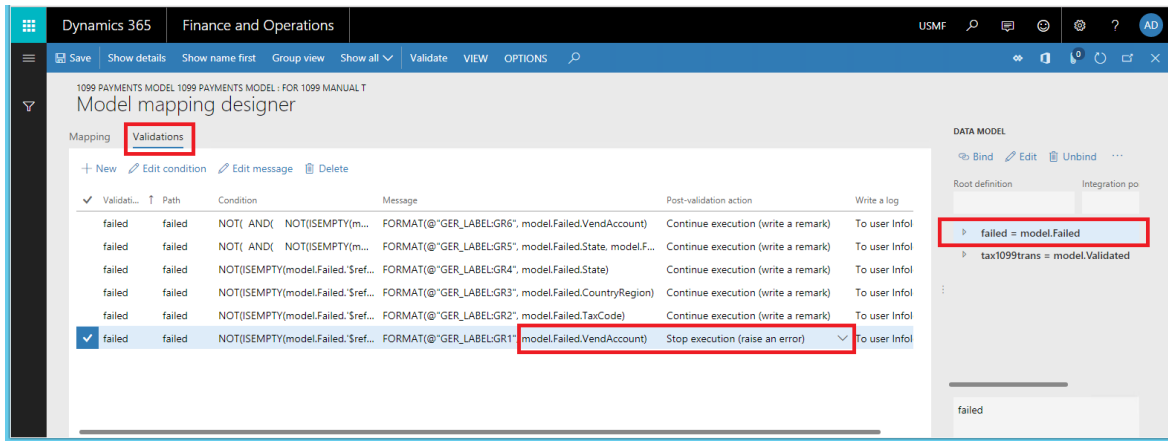
## Prepare an Excel file for import

1. Open the Excel file that you previously used. In row 3 column 1, add a vendor code that doesn't exist in the application. Add additional false vendor information to the row.

The screenshot shows an Excel spreadsheet titled '1099import-data 2.xlsx'. The table contains the following data:

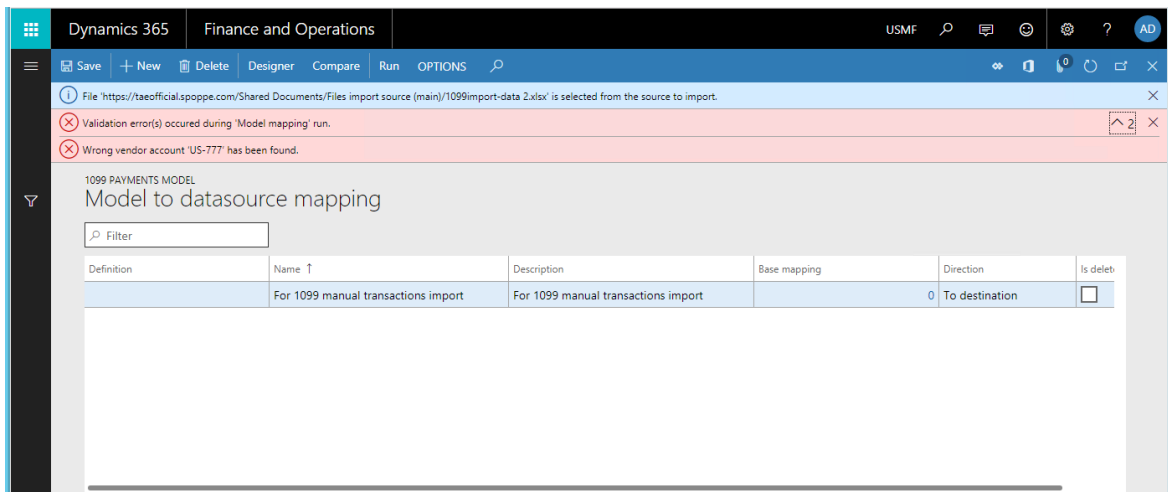
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	US-101	3/25/2018	MISC-01	3,000.00	USA	WA								
2	US-103	4/12/2018	MISC-02	4,000.00	USA	IL								
3	US-777	5/1/2018	MISC-03	450.00	USA	CO								

2. Upload the updated Excel file that contains vendors transactions to the **Files import source (main)** SharePoint folder.
3. Open the ER configurations tree, select the **1099 Payment model**, and expand the list of ER model components.
4. Select the name of the model mapping to update the model mapping so that the incorrect vendor code is considered an error during the data import process.
5. Select **Designer**.
6. On the **Validations** tab, you must change the post-validation action for the validation rule that was configured to evaluate whether the vendor account that is imported exists in the application. Update the value of the **Post-validation action** field to **Stop execution**, save your changes, and close the page.

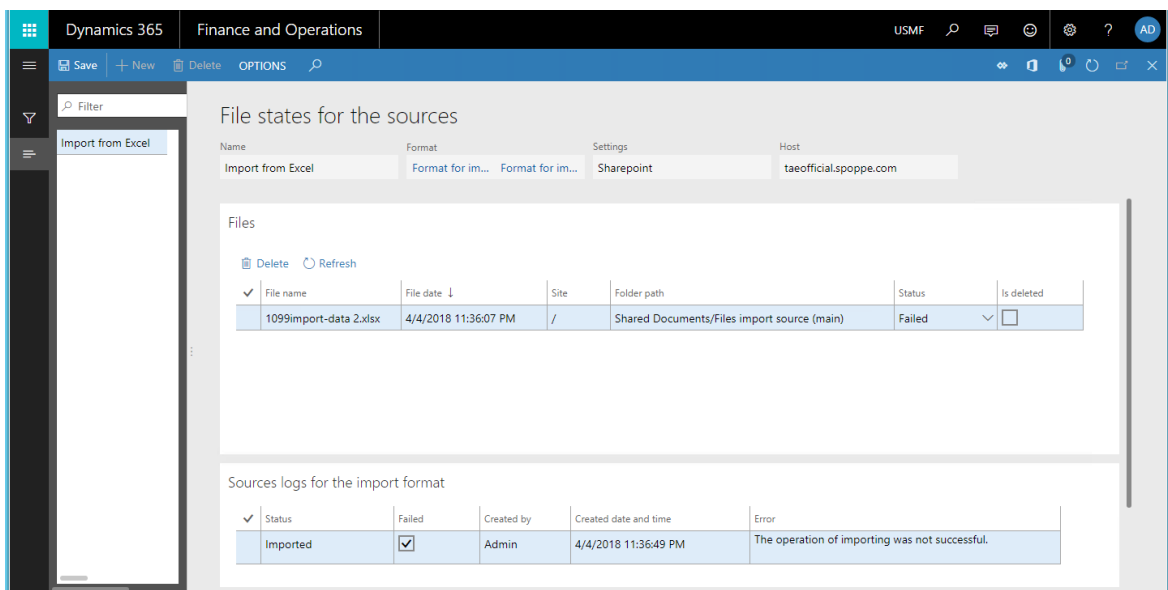


7. Save your changes, and close the ER model mapping designer.
8. Select **Run** to run the modified ER model mapping.
9. Enter the voucher ID, such as **V-00002**, and then select **OK**.

The Infolog contains a notification that there's a file in the SharePoint folder that contains incorrect vendor account and can't be imported.



10. On the **File states for the sources** page, select **Refresh**, and then, in the **Files** section, review the list of files.



The **Sources log** for the **import format** section indicates that the import process failed and that the file is in the Files error SharePoint folder (the **Is deleted** check box is not selected). If you fix this file on SharePoint by



adding the proper vendor code and then move it to the Files import source (main) SharePoint folder, you can import the file again.

11. Select **Accounts payable > Periodic tasks > Tax 1099 > Vendor settlement for 1099s**, enter appropriate values in the **From date** and **To date** fields, and then select **Manual 1099 transactions**.

Only transactions for voucher V-00001 are available. No transactions for voucher V-00002 are available even though the error for the last imported transaction has been found in the Excel file.

## Limitations

The ER framework doesn't offer the capability to initiate a new batch job that will execute a model mapping in unattended mode for data import. To do this, you must develop new logic so that the configured ER model mapping can be called from the application user interface (UI) to import data from inbound files. Therefore, some engineering work is required.

To learn more about the relevant ER API, see the [Code to run a format mapping for data import](#) section in the [ER framework API changes for Application update 7.3](#) topic.

Review the code in the `BankImport_RU` class of the `Application Suite` model to see how your custom logic can be implemented. This class extends the `RunBaseBatch` class. In particular, review the `runER()` method where the `ERIModelMappingDestinationRun` object is created as the runner of an ER model mapping.

## Additional resources

[Electronic Reporting overview](#)

[ER framework API changes for Application update 7.3](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Import files in XML format with optional attributes

2/18/2021 • 3 minutes to read • [Edit Online](#)

You can design Electronic reporting (ER) formats to parse incoming electronic documents in XML format. Certain attributes of XML elements can be specified in designed ER format as optional. It will allow you to handle incoming files with and without such XML attributes properly. You can then use the content from these files to update application data.

To learn more about this feature, complete the steps in the topic, [\(RCS\) Import files in XML format with optional attributes](#), which is part of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process. You can download this task guide and associated sample files from the [Microsoft Download Center](#).

CONTENT DESCRIPTION	FILE
Sample file in XML format	IncomingDocumentToLearnHowToHandleOptionalAttributes.xml
Task guide	RCS Import files in XML format with optional attributes.axtr

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can design ER format configuration to import files in XML format containing optional attributes. To complete these steps, you must first complete the steps in the procedure, [Create configuration providers and mark them as active](#). Before you begin, download and save locally the IncomingDocumentToLearnHowToHandleOptionalAttributes.xml file from Microsoft Download Center (<https://go.microsoft.com/fwlink/?linkid=874684>).

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as **Active**. If you don't see this configuration provider, complete the steps in the topic, [Create configuration providers and mark them as active](#).
3. Click **Reporting configurations**.

## Create a new data model configuration

1. Click **Create configuration** to open the drop dialog.
2. In the **Name** field, type 'Model to import xml file'.
3. Click **Create configuration**.
4. Click **Designer**.
5. Click **New** to open the drop dialog.
6. In the **Name** field, type 'Root'.
7. Click **Add**.
8. Click **New** to open the drop dialog.
9. In the **Name** field, type 'List'.
10. In the **Item type** field, select **Record list**.
11. Click **Add**.
12. Click **New** to open the drop dialog.
13. In the **Name** field, type 'Code'.
14. In the **Item type** field, select **String**.

15. Click **Add**.
16. Click **Save**.
17. Close the page.
18. Click **Change status**.
19. Click **Complete**.
20. Click **OK**.

## Create a format for data import

1. Click **Create configuration** to open the drop dialog.
2. In the **New** field, enter 'Format based on data model Model to import xml file'.
3. In the **Name** field, type 'Format to import xml file'.
4. Select **Yes** in the **Supports data import** field.
5. Click **Create configuration**.

## Design a format to parse incoming file in xml format

1. Click **Designer**.
2. Click **Add root** to open the drop dialog.
3. In the tree, select **XML\Element**.
4. In the **Name** field, type 'root'.
5. Click **OK**.
6. Click **Add** to open the drop dialog.
7. In the tree, select **XML\Element**.
8. In the **Name** field, type 'document'.
9. In the **Multiplicity** field, select **One many**.
10. Click **OK**.
11. In the tree, select **root\document**.
12. Click **Add** to open the drop dialog.
13. In the tree, select **XML\Attribute**.
14. In the **Name** field, type 'id'.
15. Click **OK**.
16. Click **Save**.

## Design a format mapping to save parsed information to data model

1. Click **Map format to model**.
2. Click **New**.
3. In the **Definition** field, enter or select a value.
4. In the **Name** field, type 'Mapping'.
5. Click **Save**.
6. Click **Designer**.
7. In the tree, expand **format**.
8. In the tree, expand **format\root: XML Element(root)**.
9. In the tree, select *\*format\root: XML Element(root)\document: XML Element 1.. (document)\*\**.
10. Click **Bind**.
11. In the tree, expand *\*format\root: XML Element(root)\document: XML Element 1.. (document)\*\**.
12. In the tree, select *format\root: XML Element(root)\document: XML Element 1.. (document)\id\**.

13. In the tree, expand **List = format.root.document**.
14. In the tree, select **List = format.root.document\Code**.
15. Click **Bind**.
16. Click **Save**.
17. Close the page.

## Run format mapping

1. Click **Run**.
2. Click **Browse** and select the file, **IncomingDocumentToLearnHowToHandleOptionalAttributes.xml**.
3. Click **OK**.

### NOTE

The selected file has not been imported as the format design assumes the existence of 'id' attribute for the 'document' element, but the imported file contains no such attribute.

## Modify format structure to handle xml attribute as optional

1. Close the page.
2. In the tree, expand **root\document**.
3. In the tree, select **root\document\id**.
4. In the **Empty string for missing attribute** field, select **Yes**.
5. Click **Save**.

## Run format mapping to test changes

1. Click **Map format to model**.
2. Click **Run**.
3. Click **Browse** and select the file, **IncomingDocumentToLearnHowToHandleOptionalAttributes.xml**.
4. Click **OK**.
5. Review the generated file. Note that same file has been imported as the format design now consider the 'id' attribute for the 'document' element as optional.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# (RCS) Import files in XML format with optional attributes

2/18/2021 • 3 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can design ER format configuration to import files in XML format containing optional attributes. To complete these steps, you must first complete the steps in the "Create a configuration provider and mark it as active" procedure. Before you begin, download and save locally the IncomingDocumentToLearnHowToHandleOptionalAttributes.xml file from [Microsoft Download Center](#).

1. Go to **All workspaces** > **Electronic reporting**.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as **Active**. If you don't see this configuration provider, complete the steps in the procedure [Create configuration providers and mark them as active](#).
3. Click **Reporting configurations**.

## Create a new data model configuration

1. Click **Create configuration** to open the drop dialog.
2. In the **Name** field, type 'Model to import xml file'.
3. Click **Create configuration**.
4. Click **Designer**.
5. Click **New** to open the drop dialog.
6. In the **Name** field, type 'Root'.
7. Click **Add**.
8. Click **New** to open the drop dialog.
9. In the **Name** field, type 'List'.
10. In the **Item type** field, select **Record list**.
11. Click **Add**.
12. Click **New** to open the drop dialog.
13. In the **Name** field, type 'Code'.
14. In the **Item type** field, select **String**.
15. Click **Add**.
16. Click **Save**.
17. Close the page.
18. Click **Change status**.
19. Click **Complete**.
20. Click **OK**.

## Create a format for data import

1. Click **Create configuration** to open the drop dialog.
2. In the **New** field, enter 'Format based on data model Model to import xml file'.
3. In the **Name** field, type 'Format to import xml file'.
4. Select **Yes** in the **Supports data import** field.

5. Click **Create configuration**.

## Design a format to parse incoming file in xml format

1. Click **Designer**.
2. Click **Add root** to open the drop dialog.
3. In the tree, select **XML\Element**.
4. In the **Name** field, type 'root'.
5. Click **OK**.
6. Click **Add** to open the drop dialog.
7. In the tree, select **XML\Element**.
8. In the **Name** field, type 'document'.
9. In the **Multiplicity** field, select **One many**.
10. Click **OK**.
11. In the tree, select **root\document**.
12. Click **Add** to open the drop dialog.
13. In the tree, select **XML\Attribute**.
14. In the **Name** field, type 'ID'.
15. Click **OK**.
16. Click **Save**.

## Design a format mapping to save parsed information to data model

1. Click **Map format to model**.
2. Click **New**.
3. In the **Definition** field, enter or select a value.
4. In the list, click the link in the selected row.
5. In the **Name** field, type 'Mapping'.
6. Click **Save**.
7. Click **Designer**.
8. In the tree, expand **format**.
9. In the tree, expand **format\root: XML Element(root)**.
10. In the tree, select *\*format\root: XML Element(root)\document: XML Element 1.. (document)\*\**.
11. Click **Bind**.
12. In the tree, expand *\*format\root: XML Element(root)\document: XML Element 1.. (document)\*\**.
13. In the tree, select *format\root: XML Element(root)\document: XML Element 1.. (document)\id\**.
14. In the tree, expand **List = format.root.document**.
15. In the tree, select **List = format.root.document\Code**.
16. Click **Bind**.
17. Click **Save**.
18. Close the page.

## Run format mapping

1. Click **Run**.
2. Click **Browse** and select **IncomingDocumentToLearnHowToHandleOptionalAttributes.xml**.
3. Click **OK**.

#### NOTE

The selected file has not been imported as the format design assumes the existence of 'id' attribute for the 'document' element, but the imported file contains no such attribute.

## Modify format structure to handle xml attribute as optional

1. Close the page.
2. In the tree, expand **root\document**.
3. In the tree, select **root\document\id**.
4. Select **Yes** in the **Empty string for missing attribute** field.
5. Click **Save**.

## Run format mapping to test changes

1. Click **Map format to model**.
2. Click **Run**.
3. Click **Browse** and select the **IncomingDocumentToLearnHowToHandleOptionalAttributes.xml** file.
4. Click **OK**.
5. Review the generated file.

#### NOTE

The same file has been imported as the format design now consider the 'id' attribute for the 'document' element as optional.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Split generated XML files based on file size and content quantity

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can design Electronic reporting (ER) formats to generate outgoing documents in XML format. Sometimes, those documents can be accepted only when they meet specific criteria, such a maximum file size or a maximum number of some XML nodes. You can design ER formats to generate electronic documents that satisfy the requirements that the recipients of those documents specify.

- For the FILE format element, you can define a limit on the file size as an ER expression. If the defined limit is exceeded when an ER report is generated, ER finishes creating the current file and then moves on to create the next file.
- For any XML ELEMENT format, you can define a limit on the number of elements as an ER expression. If the number of XML nodes in the file that is generated exceeds the defined limit when an ER report is run, ER finishes creating the current file and then moves on to create the next file.
- For any XML SEQUENCE format element, you can define a limit on the number of child elements as an ER expression. If the number of nested XML nodes of the format element in the generated file exceeds the defined limit when an ER report is run, ER finishes creating the current file and then moves on to create the next file.
- You can mark any XML ELEMENT format element as non-breakable. In this way, you can keep the nested items of XML nodes that are generated under the format element in a single generated file.

In addition to using the XML ELEMENT and XML SEQUENCE format elements to add XML nodes to the generated file, you can use the RAW XML format element. However, nodes that you add by using the RAW XML format element aren't considered when the number of nodes is calculated to evaluate the limits on the number of elements.

If you configured file destinations for a FILE format element that has been configured to split the generated output whenever specific limits are exceeded, each piece of generated output is sent to the configured file destination as an individual file. To uniquely name the files that are created by splitting the output, you must configure an ER expression for the FILE format element. If you include an ER data source of the NUMBER SEQUENCE type, the number sequence will be incremented for each piece of the split output.

To learn more about this feature, play the **ER Split XML files based on the file size or content item quantity** task guide, which is part of the **7.5.4.3 Acquire/Develop IT service/solution components (10677)** business process and can be downloaded from the [Microsoft Download Center](#). This task guide walks you through the process of configuring an ER format to split generated files based on limits on the file size and content item quantity. To complete the task guide, you must download the following files:

- [ER model configuration - XmlFilesSplittingModel.xml](#)
- [ER format configuration - XmlFilesSplittingFormat.xml](#)

## Additional resources

[Electronic reporting \(ER\) destinations](#)

[Formula designer in Electronic reporting \(ER\)](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Support parameterized calls of ER data sources of the Calculated field type

2/18/2021 • 10 minutes to read • [Edit Online](#)

This topic explains how you can design an Electronic reporting (ER) data source by using the **Calculated field** type. This data source may contain an ER expression that, when executed, can be controlled by the values of the parameter arguments that are configured in a binding that calls this data source. By configuring parameterized calls of such a data source, you can reuse a single data source in many bindings, which reduces the total number of data sources that must be configured in ER model mappings or ER formats. It also simplifies the configured ER component, which reduces the maintenance costs and the cost of use by other consumers.

## Prerequisites

To complete the examples in this topic, you must have the following access:

- Access to one of these roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
- Access to Regulatory Configuration Services (RCS) that have been provisioned for the same tenant as Finance and Operations for one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator

You must also download and locally store the following files.

CONTENT	FILE NAME
Sample ER data model configuration	<a href="#">Model to learn parameterized calls.version.1.xml</a>
Sample ER metadata configuration	<a href="#">Metadata to learn parameterized calls.version.1.xml</a>
Sample ER model mapping configuration	<a href="#">Mapping to learn parameterized calls.version.1.1.xml</a>
Sample ER format configuration	<a href="#">Format to learn parameterized calls.version.1.1.xml</a>

## Sign in to your RCS instance

In this example, you will create a configuration for the sample company, Litware, Inc. First, in RCS, you must complete the steps in the [Create configuration providers and mark them as active](#) procedure:

1. On the default dashboard, select **Electronic reporting**.
2. Select **Reporting configurations**.
3. Import the downloaded configurations to RCS in the following sequence: data model, metadata, model mapping, format. Complete the following steps for each ER configuration:

- a. Select **Exchange**.
- b. Select **Load from XML file**.
- c. Select **Browse**, and then select the required ER configuration in XML format.
- d. Select **OK**.

## Review the provided ER solution

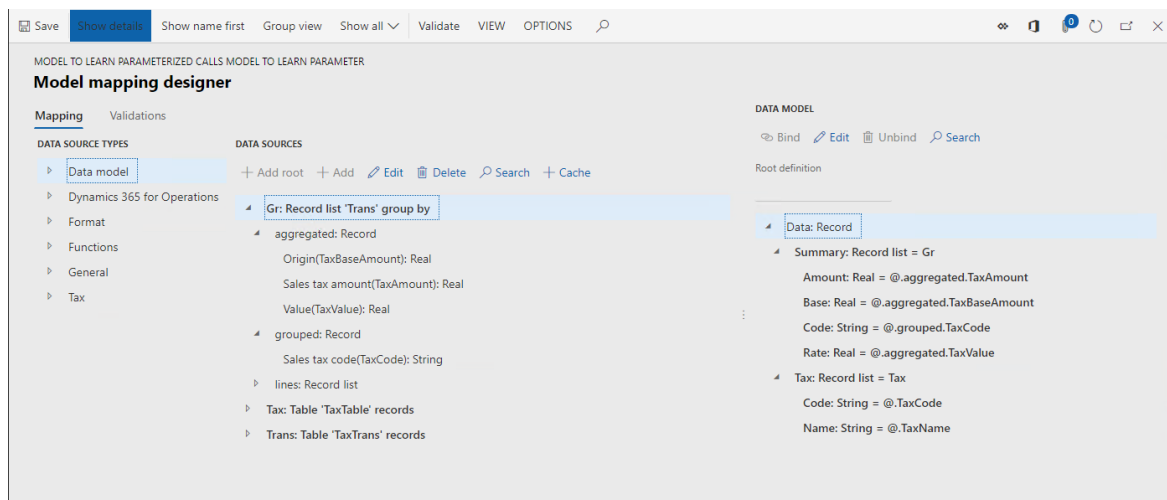
### Review model mapping

1. In the configuration tree, expand the content of the **Model to learn parameterized calls** item.
2. Select **Mapping to learn parameterized calls**.
3. Select **Designer**.
4. Select **Designer**.

This ER model mapping is designed to do the following:

- Fetch the list of tax codes (**Tax** data source) residing in the **TaxTable** table.
- Fetch the list of tax transactions (**Trans** data source) residing in the **TaxTrans** table:
  - Group the list of fetched transactions (**Gr** data source) by tax code.
  - Calculate for grouped transactions following aggregated values per tax code:
    - Sum of tax base values.
    - Sum of tax values.
    - Minimum value of applied tax rate.

The model mapping in this configuration implements the base data model for any of the ER formats created for this model and executed in Finance and Operations. As a result, the content of the **Tax** and **Gr** data sources is exposed for ER formats such as abstract data sources.

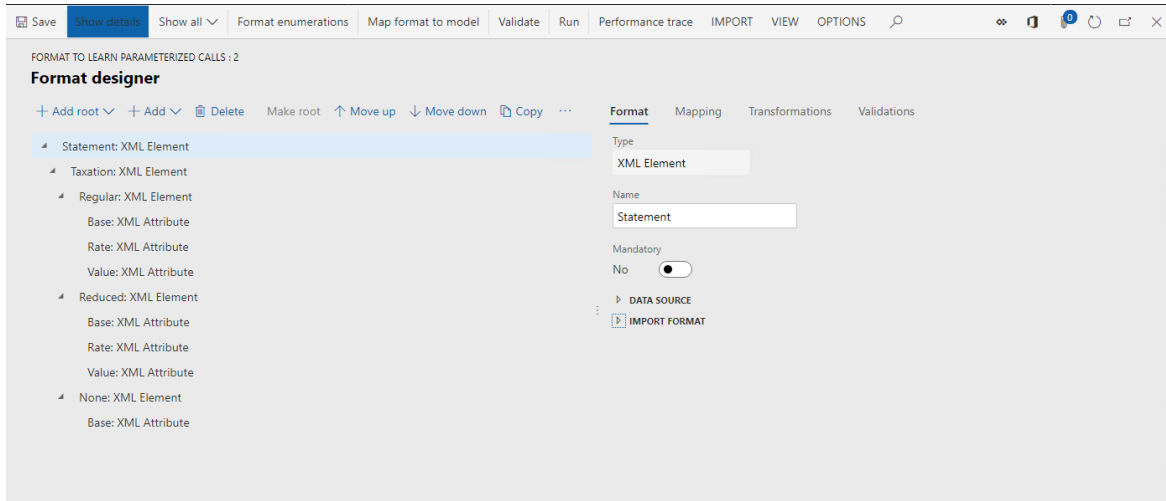


5. Close the **Model mapping designer** page.
6. Close the **Model mapping** page.

### Review format

1. In the configuration tree, expand the content of the **Model to learn parameterized calls** item.
2. Select **Format to learn parameterized calls**.
3. Select **Designer**. This ER format is designed to do the following:

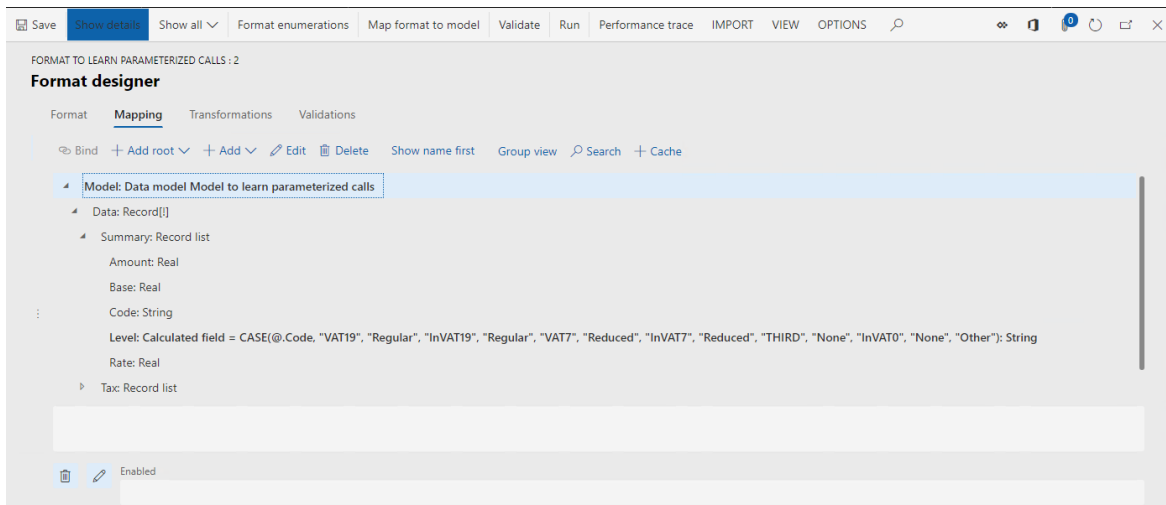
- Generate a tax statement in XML format.
- Present the following levels of taxation in the tax statement: regular, reduced, and none.
- Present multiple details at each taxation level, having a different number of details in each level.



4. Select **Mapping**.

5. Expand the **Model**, **Data**, and **Summary** items.

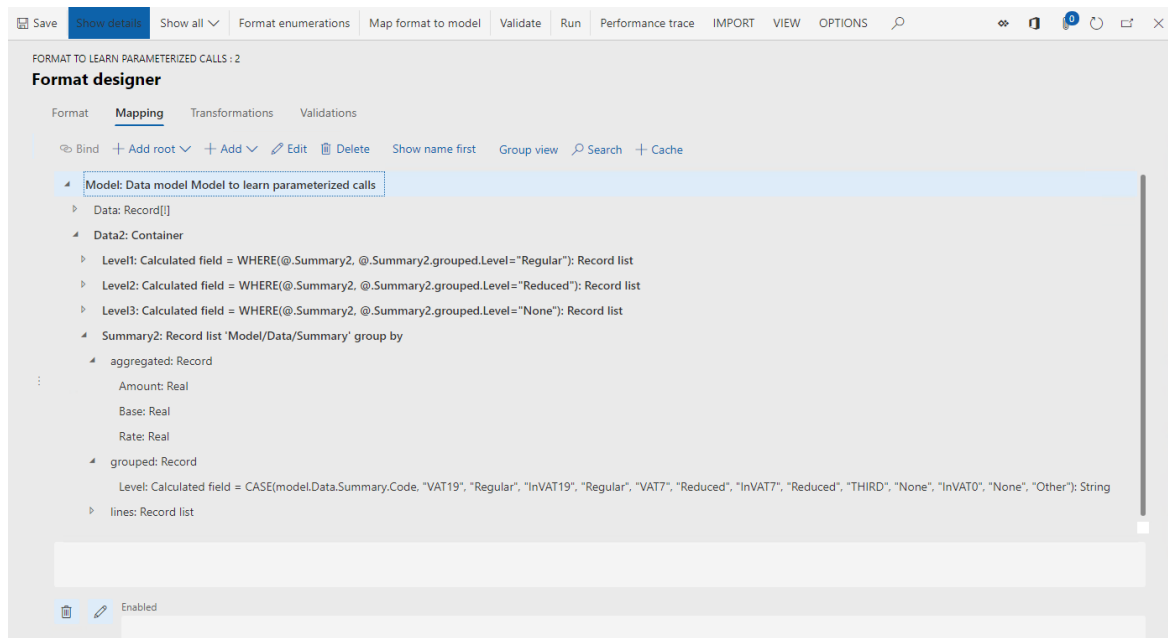
The calculated field **Model.Data.Summary.Level** contains the expression that returns the code of the taxation level (**Regular**, **Reduced**, **None**, or **Other**) as a text value for any tax code that can be retrieved from the **Model.Data.Summary** data source at run time.



6. Expand the **Model.Data2** item.

7. Expand the **Model.Data2.Summary2** item.

The **Model.Data2.Summary2** data source is configured to group the **Model.Data.Summary** data source transaction details by taxation level (returned by the **Model.Data.Summary.Level** calculated field) and compute the aggregations.



8. Review the calculated fields **Model.Data2.Level1**, **Model.Data2.Level2**, and **Model.Data2.Level3**. These calculated fields are used to filter the **Model.Data2.Summary2** records list and return only records that represent a particular taxation level.
9. Close the **Format designer** page.

## Create a derived format

You can improve the provided format by adding one calculated field to filter the required taxation level instead of using the existing three fields: **Model.Data2.Level1**, **Model.Data2.Level2**, and **Model.Data2.Level3**. The required taxation level can be specified in the location where this new calculated field will be called.

1. In the configuration tree, expand the content of the **Model to learn parameterized calls** item.
2. Select **Format to learn parameterized calls**.
3. Select **Create configuration**.
4. Select **Derive from Name: Format to learn parameterized calls, Microsoft**.
5. In the **Name** field, enter **Format to learn parameterized calls (custom)**.
6. Select **Create configuration**.

## Configure a parameterized calculated field that returns a list of records

### Start adding a new calculated field

1. Select **Designer**.
2. Select **Expand/collapse** to expand all format items.
3. Select **Mapping**.
4. Expand the **Model** item.
5. Select the **Model.Data2** item.
6. Select **Add**.
7. Select **Functions\Calculated field**.
8. In the **Name** field, enter **Levels**.
9. Select **Edit formula**.

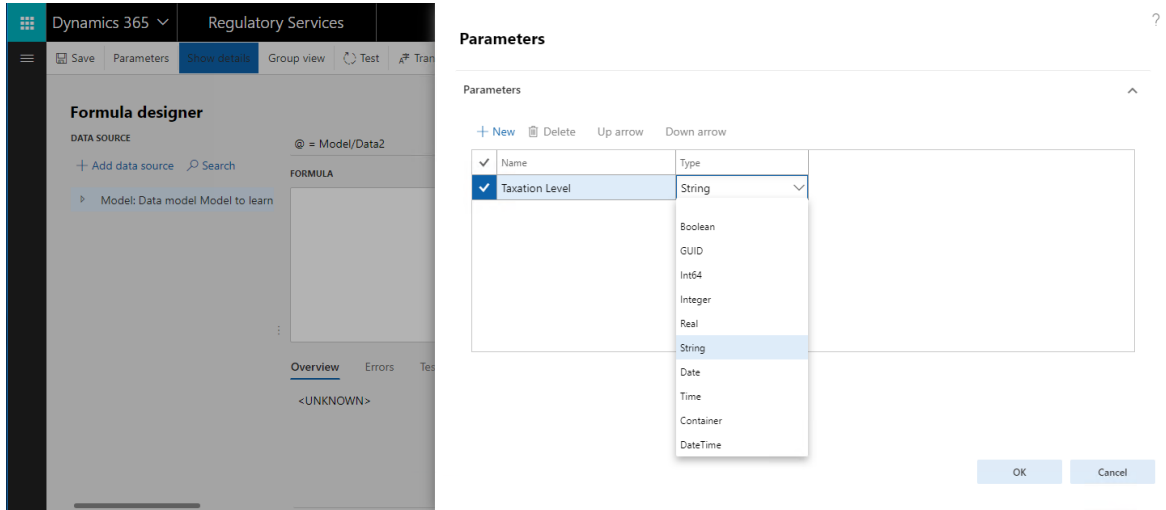
### Define a parameter for adding a calculated field

1. Select **Parameters**.

2. Select **New**.
3. In the **Name** field, enter **Taxation Level**.
4. In the **Type** field, select **String**.

Only primitive data types can be used to specify the type of the parameter's argument. Therefore, **Record list**, **Record**, and **Enum** types cannot be used for this purpose.

The maximum number of parameters that can be specified for a single calculated field is 8.

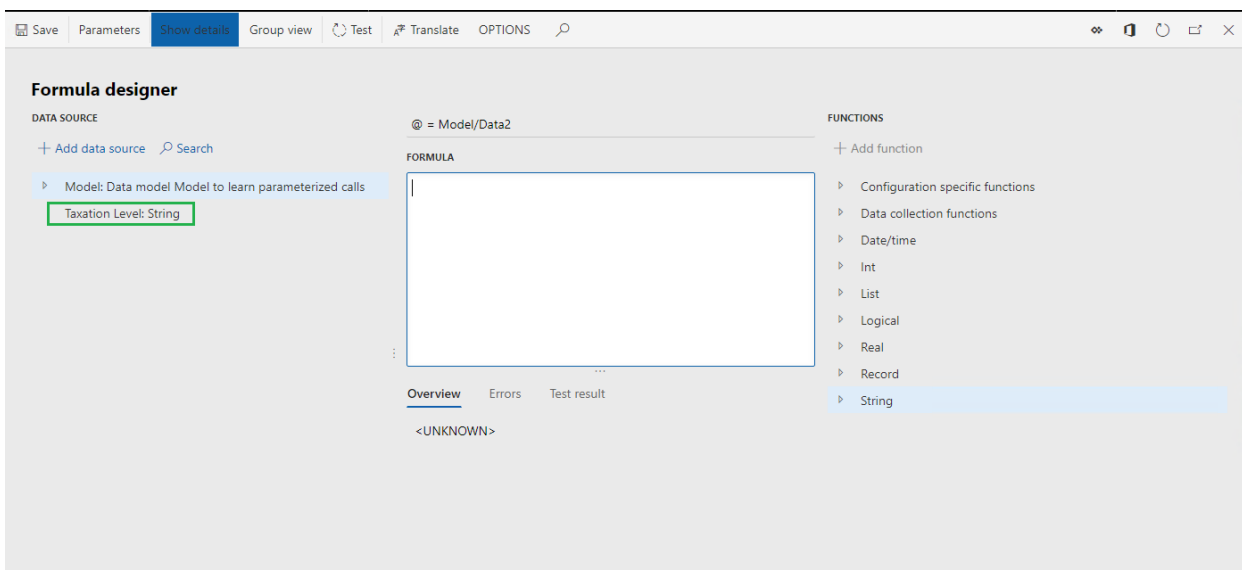


5. Select **OK**.

By adding this parameter, you specify the condition that must be in place to call this calculated field. When you call this calculated field, you need to specify the argument of the **Taxation Level** parameter as a value with **String** format.

Make sure that you define parameters only for those calculated fields that reside in a container (either **Record list**, **Record**, or **Container**).

The configured parameter is available in the list of data sources for this calculated field. You can add the parameter to the configured expression by selecting **Add data source**.



## Define an expression for adding a calculated field

1. In the **Formula** field, enter:

**WHERE(@.Summary2, @.Summary2.grouped.Level =**

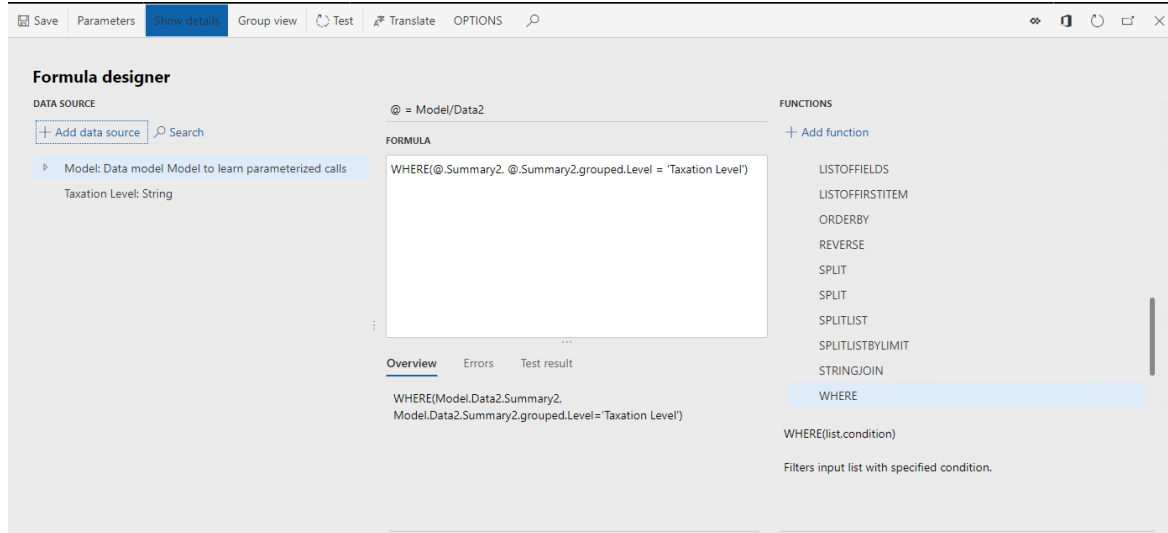
2. Select the **Taxation Level** parameter in the list of data sources.

3. Select **Add data source**.

4. In the **Formula** field, finalize the expression as:

**WHERE(@.Summary2, @.Summary2.grouped.Level = 'Taxation Level')**

5. Select **Save**.

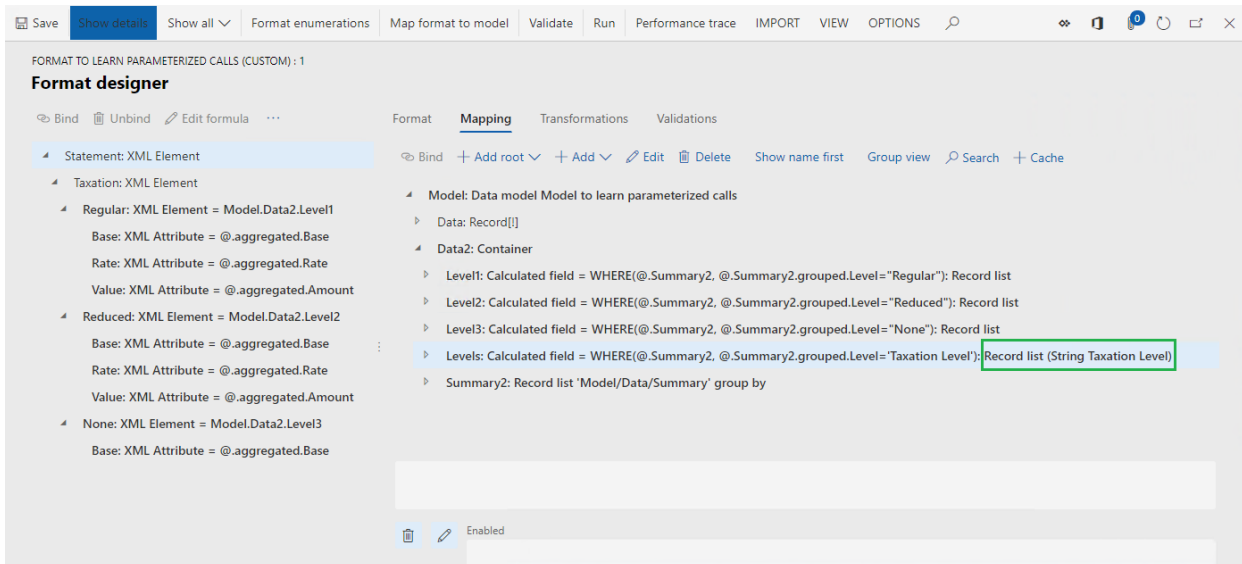


6. Close the **Formula designer** page.

### Finish adding a new calculated field

- Select **OK**.

On the **Format designer** page, the configured parameterized calculated field **Levels** requires a **String** argument.



### Use the configured calculated field for binding format elements

1. Select **Model.Data2.Levels** to select the configured calculated field.

2. Select the **Statement.Taxation.Regular** format element.

3. Select **Bind**.

4. Select **Yes** to confirm the replacement of the currently used data source, **Level1**, by the new data source, **Levels**, in all nested format elements of the selected format element.

Applied binding has been built as a call of the parameterized calculated field. By default, the name of the bound format element is used as an argument for parameterized calculated field under the following conditions:

- The calculated field is configured to use a single parameter.
- The data type of this parameter is defined as **String**.

When the name of the bound format element is blank, the data source name of this element is used in applied binding.

5. Select the **Statement.Taxation.Reduced** format element.
6. Select **Bind**.
7. Select **Yes** to confirm the replacement of the currently used data source, **Level2**, by the new data source, **Levels**, in all nested format elements under the selected format element.
8. Select the **Statement.Taxation.None** format element.
9. Select **Bind**.
10. Select **Yes** to confirm the replacement of the currently used data source, **Level3**, by the new data source, **Levels**, in all nested format elements under the selected format element.

When you specify the argument of the parameterized calculated field for the XML element representing taxation level (for example, **Model.Data2.Levels("Reduced")** as a text value), you don't need to do the same for nested XML attributes—their bindings will automatically inherit the value of the argument defined on the parent level (**Model.Data2.Levels.aggregated.Base**, not **Model.Data2.Levels("Reduced").aggregated.Base**).

Recurrent calls of any parameterized calculated field are not supported.

You can select **Edit formula**, and change the applied-by-default argument of the parameterized calculated field in the selected binding. If this argument is missing, it can cause errors at run time — users are informed about such a situation when the current format is validated.

The screenshot shows the 'Format designer' interface. At the top, a yellow banner displays a warning: 'Validation warnings exist'. Below this, the 'Format designer' window is open, showing a tree view on the left and a 'Mapping' tab on the right. The tree view shows a hierarchy: 'Statement: XML Element' > 'Taxation: XML Element' > 'None: XML Element = Model.Data2.Levels'. The 'None' element is selected, and its base is set to '@.aggregated.Base'. The 'Mapping' tab shows two calculated fields: 'Level3: Calculated field = WHERE(@.Summary2, @.Summary2.grouped.Level="None"): Record list' and 'Levels: Calculated field = WHERE(@.Summary2, @.Summary2.grouped.Level="Taxation Level"): Record list (String Taxation Level)'. Below the mapping, a 'DETAILS' section shows 'Warnings (1)'. A table lists the warning:

Description	Path	Resolutions
Value of mandatory parameter 1 is not provided in 'Model.Data2.Levels'	Binding: Statement: XML Element/Taxation: XML Element/None: XML Element	None

## Configure a parameterized calculated field to return a record

When a parameterized calculated field returns a record, you need to support binding of individual fields of this record to format elements. In such cases there will be no parent binding that contains the value of an argument



to call a parameterized calculated field — this value must be defined in the binding of a single record's field.

### **Start adding a new calculated field**

1. Select the **Model.Data2** item.
2. Select **Add**.
3. Select **Functions\Calculated field**.
4. In the **Name** field, enter **LevelRecord**.
5. Select **Edit formula**.

### **Define a parameter for adding a calculated field**

1. Select **Parameters**.
2. Select **New**.
3. In the **Name** field, enter **Taxation Level**.
4. In the **Type** field, select **String**.
5. Select **OK**.

### **Define an expression for adding a calculated field**

1. In the **Formula** field, enter the following:

```
FIRSTORNULL(@.Levels(
```

2. Select the **Taxation Level** parameter.

3. Select **Add data source**.

4. In the **Formula** field, append **'Taxation Level'))** to what you entered in Step 1 to finalize the expression to:

```
FIRSTORNULL(@.Levels('Taxation Level'))
```

5. Select **Save**.

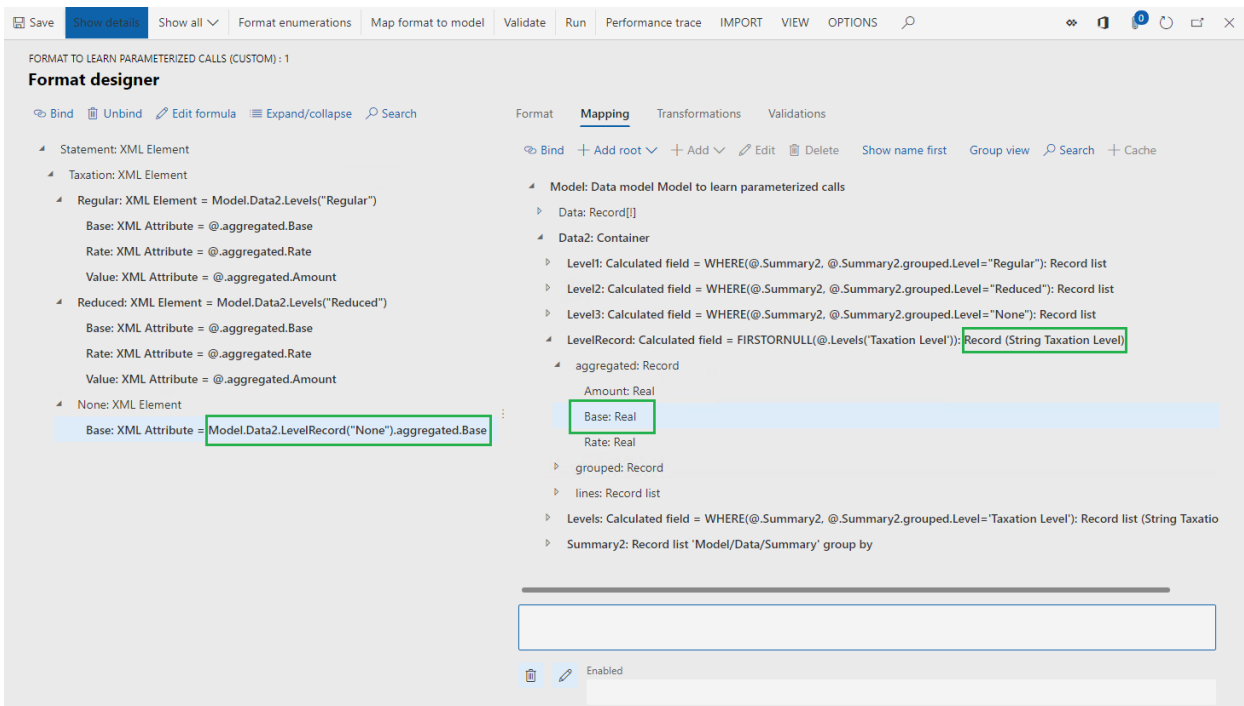
6. Close the **Formula designer** page.

### **Finish adding a new calculated field**

- Select **OK**.

### **Use the configured calculated field to bind format elements**

1. Expand **Model.Data2.LevelRecord** to select the configured calculated field.
2. Expand the **Model.Data2.LevelRecord.agggregated** container of the configured calculated field.
3. Select the **Model.Data2.LevelRecord.agggregated.Base** field.
4. Select the **Statement.Taxation.None** format element.
5. Select **Unbind**.
6. Select the **Statement.Taxation.None.Base** format element.
7. Select **Bind**.
8. Select **Edit formula**.
9. Change the expression to **Model.Data2.LevelRecord("None").agggregated.Base**.



## Remove calculated fields that are not used

1. Select **Model.Data2.Level1**.
2. Select **Delete**.
3. Select **Model.Data2.Level2**.
4. Select **Delete**.
5. Select **Model.Data2.Level3**.
6. Select **Delete**.
7. Select **Save**.

### NOTE

You reused the same calculated field **Model.Data2.Levels** several times in format bindings. It is much easier to use and maintain a single calculated field instead of doing this for multiple similar fields.

8. Close the **Format designer** page.

## Complete adjusted version of a derived format

1. In the **Versions** FastTab, select **Change status**.
2. Select **Complete**.

## Export completed version of a derived format

1. Select **Format to learn parameterized calls (custom)** format in the configurations tree.
2. In the **Versions** FastTab, select the completed version 1.1.1.
3. Select **Exchange**.
4. Select **Export as XML file**.
5. Store the downloaded configuration locally, in XML format.

## Test ER formats

You can run the initial and improved ER formats to make sure that configured parameterized calculated fields work properly.

### Import ER configurations

You can import reviewed configurations from RCS by using the ER repository of the RCS type. If you already went through the steps in the topic, [Import Electronic reporting \(ER\) configurations from Regulatory Configuration Services \(RCS\)](#), use the configured ER repository to import configurations discussed earlier in this topic to your environment. Otherwise, follow these steps:

1. Select the **DEMF** company and on the default dashboard, select **Electronic reporting**.
2. Select **Reporting configurations**.
3. Import the configurations from Microsoft Download Center in the following sequence: data model, model mapping, format. Complete the following steps for each ER configuration:
  - a. Select **Exchange**.
  - b. Select **Load from XML file**.
  - c. Select **Browse** to select the required ER configuration in XML format.
  - d. Select **OK**.
4. Import the exported from RCS completed version 1.1.1 of the **Format to learn parameterized calls (custom)** format:
  - a. Select **Exchange**.
  - b. Select **Load from XML file**.
  - c. Select **Browse** to select the locally stored **Format to learn parameterized calls (custom)** file in XML format.
  - d. Select **OK**.

### Run ER formats

1. In the configuration tree, expand the content of the **Model to learn parameterized calls** item.
2. Select **Format to learn parameterized calls**.
3. Select **Run** on the top-most ribbon.
4. Save the locally generated output.
5. Select the **Format to learn parameterized calls (custom)** item.
6. Select **Run** on the top-most ribbon.
7. Save the generated output locally.
8. Compare the contents of the generated outputs.

## Additional resources

- [Formula designer in Electronic reporting \(ER\)](#)
- [Improve performance of ER solutions by adding parameterized CALCULATED FIELD data sources](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage the Electronic reporting (ER) configuration lifecycle

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to manage the lifecycle of Electronic reporting (ER) configurations for Microsoft Dynamics 365 Finance.

## Overview

Electronic reporting (ER) is an engine that supports statutory required and country-specific electronic documents. In general, ER assumes an ability to perform the following tasks for a single electronic document. For more details, see [Electronic reporting \(ER\) overview](#).

- Design a template for an electronic document:
  - Identify the required sources of data that can be presented in the document:
    - Underlying data, such as data tables, data entities, and classes.
    - Process-specific properties, such as execution date and time, and time zone.
    - User input parameters, specified by the end user at run time.
  - Define the required document elements and their topology to specify a final document format.
  - Configure the desired flow of data from selected data sources to defined document elements (via data source bindings to document format components), and specify process control logic.
- Make a template available so that it can be used in other instances:
  - Transform a document template that was created into an ER configuration, and export the configuration from the current application instance as an XML package that can be stored either locally or in LCS.
  - Transform an ER configuration into an application document template.
  - Import an XML package that is stored either locally or in LCS into the current instance.
- Customize the template of an electronic document:
  - Bring a template from LCS into the current instance as an ER configuration.
  - Design a custom version of an ER configuration, and keep a reference to the base version.
- Integrate a template with a particular business process, so that it's available in the application:
  - Configure settings so that the application starts to use an ER configuration, by referring to that configuration in a process-related parameter. For example, refer to the ER configuration in a specific Accounts payable payment method to generate an electronic payment message for processing invoices.
- Use a template in a specific business process:
  - Run an ER configuration in a specific business process. For example, to generate an electronic payment message for processing invoices when a payment method that references the ER configuration is selected.

## Concepts

The following roles and related activities are associated with the ER configuration lifecycle.

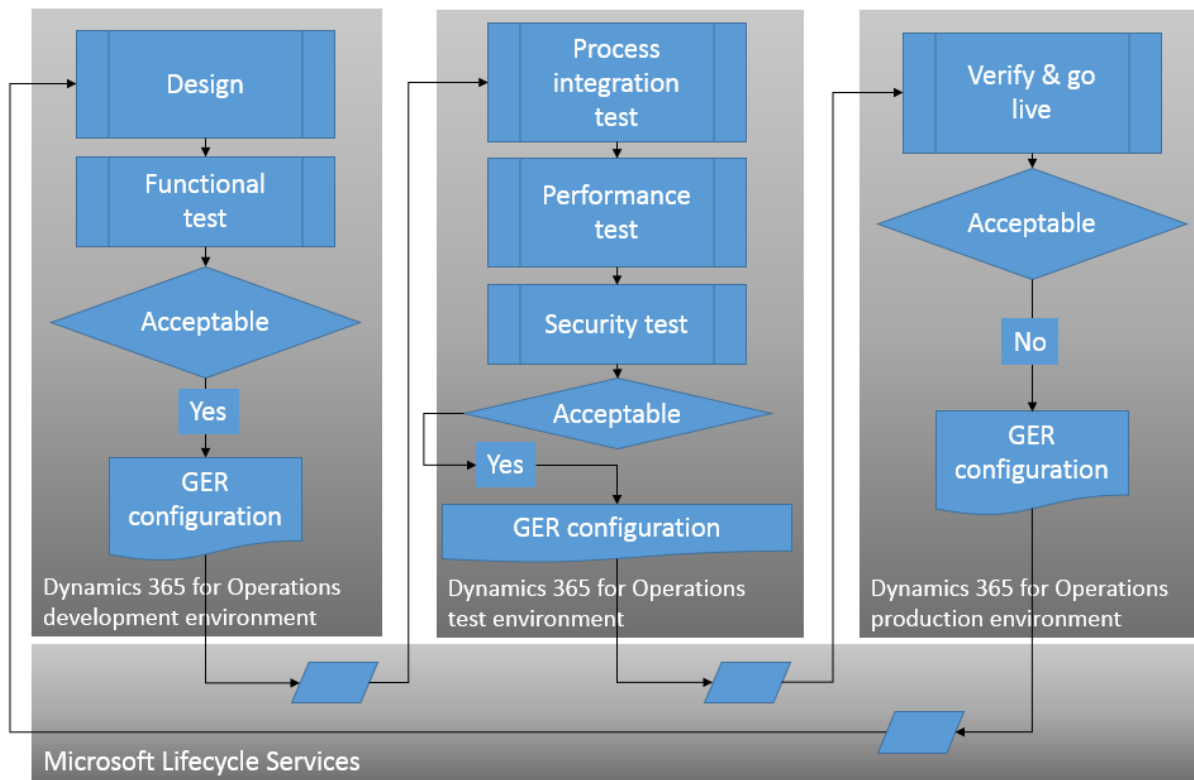
ROLE	ACTIVITIES	DESCRIPTION
Electronic reporting functional consultant	Create and manage ER components (models and formats).	A business person who designs ER domain-specific data models, designs the required templates for electronic documents, and binds them accordingly.
Electronic reporting developer	Create and manage data model mappings.	A specialist who selects the required Finance data sources and binds them to ER domain-specific data models.
Accounting supervisor	Configure process-related settings that reference ER artifacts.	For example, an <b>Accounting supervisor</b> role that allows the settings of an ER configuration to be used in a particular Accounts payable payment method to generate an electronic payment message for processing invoices.
Accounts payable payments clerk	Use ER artifacts in a specific business process.	For example, an <b>Accounts payable payments clerk</b> role that allows electronic payment messages to be generated for processing invoices, based on the ER format that is configured for a specific payment method.

## ER configuration development lifecycle

For the following ER-related reasons, we recommend that you design ER configurations in the development environment, as a separate instance of Finance and Operations:

- Users in either the **Electronic reporting developer** role or the **Electronic reporting functional consultant** role can edit configurations and run them for testing purposes. This scenario can cause calls of methods of classes and tables that might harm business data and the performance of the instance.
- Calls of methods of classes and tables as ER data sources of ER configurations aren't restricted by entry points and logged company content. Therefore, users in either the **Electronic reporting developer** role or the **Electronic reporting functional consultant** role can access business-sensitive data.

ER configurations that are designed in the development environment can be uploaded to the test environment for the configuration evaluation (proper process integration, correctness of results, and performance) and quality assurance, such as correctness of role-driven access rights and segregation of duties. The features that enable ER configuration interchange can be used for this purpose. Finally, proven ER configurations can be uploaded either to LCS, where they can be shared with service subscribers, or to the production environment for internal use, such as shown in the following illustration.



## Additional resources

[Electronic reporting \(ER\) overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upload a configuration into Lifecycle Services

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how a user in the System administrator or Electronic reporting developer role can create a new [Electronic reporting \(ER\) configuration](#) and upload it into the [project-level Asset library](#) in Microsoft Dynamics Lifecycle Services (LCS).

In this example, you will create a configuration and upload it into LCS for a sample company that is named Litware, Inc. These steps can be completed in any company, because ER configurations are shared among companies. To complete these steps, you must first complete the steps in [Create configuration providers and mark them as active](#). Access to LCS is also required.

1. Sign in to the application by using one of the following roles:
  - Electronic reporting developer
  - System administrator
2. Go to **Organization administration > Workspaces > Electronic reporting**.
3. Select **Litware, Inc.**, and mark it as **Active**.
4. Select **Configurations**.

## NOTE

Make sure that the current Dynamics 365 Finance user is a member of the LCS project that contains the [Asset library](#) that is used to import ER configurations.

You can't access an LCS project from an ER repository that represents a different domain than the domain that is used in Finance. If you try, an empty list of LCS projects will be shown, and you won't be able to import ER configurations from the project-level Asset library in LCS. To access project-level Asset libraries from an ER repository that is used to import ER configurations, sign in to Finance by using the credentials of a user who belongs to the tenant (domain) that the current Finance instance has been provisioned for.

## Create a new data model configuration

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the **Configurations** page, select **Create configuration** to open the drop-down dialog box.

In this example, you will create a configuration that contains a sample data model for electronic documents. This data model configuration will be uploaded into LCS later.
3. In the **Name** field, enter **Sample model configuration**.
4. In the **Description** field, enter **Sample model configuration**.
5. Select **Create configuration**.
6. Select **Model designer**.
7. Select **New**.
8. In the **Name** field, enter **Entry point**.
9. Select **Add**.

10. Select **Save**.
11. Close the page.
12. Select **Change status**.
13. Select **Complete**.
14. Select **OK**.
15. Close the page.

## Register a new repository

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. In the **Configuration providers** section, select the **Litware, Inc.** tile.
3. On the **Litware, Inc.** tile, select **Repositories**.

You can now open the list of repositories for the Litware, Inc. configuration provider.

4. Select **Add** to open the drop-down dialog box.

You can now add a new repository.

5. In the **Configuration repository enter** field, select **LCS**.
6. Select **Create repository**.
7. In the **Project** field, enter or select a value.

For this example, select the desired LCS project. You must have [access](#) to the project.

8. Select **OK**.

Complete a new repository entry.

9. In the list, mark the selected row.

For this example, select the **LCS** repository record.

Note that a registered repository is marked by the current provider. In other words, only configurations that are owned by that provider can be put in this repository and therefore uploaded into the selected LCS project.

10. Select **Open**.

You open the repository to view the list of ER configurations. If the selected project hasn't yet been used for ER configurations sharing, the list will be empty.

11. Close the page.
12. Close the page.

## Upload a configuration into LCS

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the **Configurations** page, in the configurations tree, select **Sample model configuration**.

You must select a created configuration that has been already completed.

3. In the list, find and select the desired record.



For this example, select the version of the selected configuration that has a status of **Completed**.

4. Select **Change status**.

5. Select **Share**.

The status of the configuration is changed from **Completed** to **Shared** when the configuration is published in LCS.

6. Select **OK**.

7. In the list, find and select the desired record.

For this example, select the configuration version that has a status of **Shared**.

Note that the status of the selected version was changed from **Completed** to **Shared**.

8. Close the page.

9. Select **Repositories**.

You can now open the list of repositories for the Litware, Inc. configuration provider.

10. Select **Open**.

For this example, select the **LCS** repository, and open it.

Notice that the selected configuration is shown as an asset of the selected LCS project.

11. Open LCS by going to <https://lcs.dynamics.com>.

12. Open a project that was used earlier for repository registration.

13. Open the Asset library of the project.

14. Select the **GER configuration** asset type.

The ER configuration that you uploaded should be listed.

Note that the uploaded LCS configuration can be imported into another instance if providers have access to this LCS project.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Import a configuration from Lifecycle Services

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how a user in the System administrator or Electronic reporting developer role can import a new version of an [Electronic reporting \(ER\) configuration](#) from the [project-level Asset library](#) in Microsoft Dynamics Lifecycle Services (LCS).

In this example, you will select the desired version of the ER configuration and import it for a sample company that is named Litware, Inc. These steps can be completed in any company, because ER configurations are shared among companies. To complete these steps, you must first complete the steps in [Upload a configuration into Lifecycle Services](#). Access to LCS is also required.

1. Sign in to the application by using one of the following roles:
  - Electronic reporting developer
  - System administrator
2. Go to **Organization administration > Workspaces > Electronic reporting**.
3. Select **Configurations**.

## NOTE

Make sure that the current Dynamics 365 Finance user is a member of the LCS project that contains the Asset library that the user wants to [access](#) to import ER configurations.

You can't access an LCS project from an ER repository that represents a different domain than the domain that is used in Finance. If you try, an empty list of LCS projects will be shown, and you won't be able to import ER configurations from the project-level Asset library in LCS. To access project-level Asset libraries from an ER repository that is used to import ER configurations, sign in to Finance by using the credentials of a user who belongs to the tenant (domain) that the current Finance instance has been provisioned for.

## Delete a shared version of a data model configuration

1. On the **Configurations** page, in the configurations tree, select **Sample model configuration**.

You created the first version of a sample data model configuration and published it to LCS when you completed the steps in [Upload a configuration into Lifecycle Services](#). In this procedure, you will delete that version of the ER configuration. You will then import that version from LCS later in this topic.

2. In the list, find and select the desired record.

For this example, select the version of the configuration that has a status of **Shared**. This status indicates that the configuration has been published to LCS.

3. Select **Change status**.
4. Select **Discontinue**.

By changing the status of the selected version from **Shared** to **Discontinued**, you make the version available for deletion.

5. Select **OK**.
6. In the list, find and select the desired record.

For this example, select the version of the configuration that has a status of **Discontinued**.

7. Select **Delete**.

8. Select **Yes**.

Notice that the only draft version 2 of the selected data model configuration is now available.

9. Close the page.

## Import a shared version of a data model configuration from LCS

1. Go to **Organization administration > Workspaces > Electronic reporting**.

2. In the **Configuration providers** section, select the **Litware, Inc.** tile.

3. On the **Litware, Inc.** tile, select **Repositories**.

You can now open the list of repositories for the Litware, Inc. configuration provider.

4. Select **Open**.

For this example, select the **LCS** repository, and open it. You must have **access** to the LCS project and to the Asset library that is accessed by the selected ER repository.

5. In the list, mark the selected row.

For this example, select the first version of **Sample model configuration** in the version list.

6. Select **Import**.

7. Select **Yes** to confirm the import of the selected version from LCS.

An informational message confirms that the selected version was successfully imported.

8. Close the page.

9. Close the page.

10. Select **Configurations**.

11. In the tree, select **Sample model configuration**.

12. In the list, find and select the desired record.

For this example, select the version of the configuration that has a status of **Shared**.

Notice that shared version 1 of the selected data model configuration is also available now.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

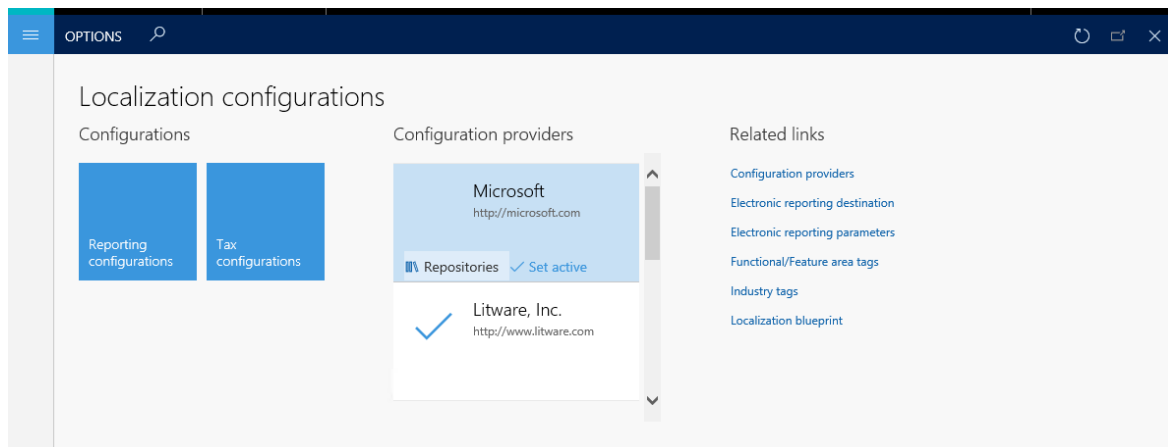
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Download Electronic reporting configurations from Lifecycle Services

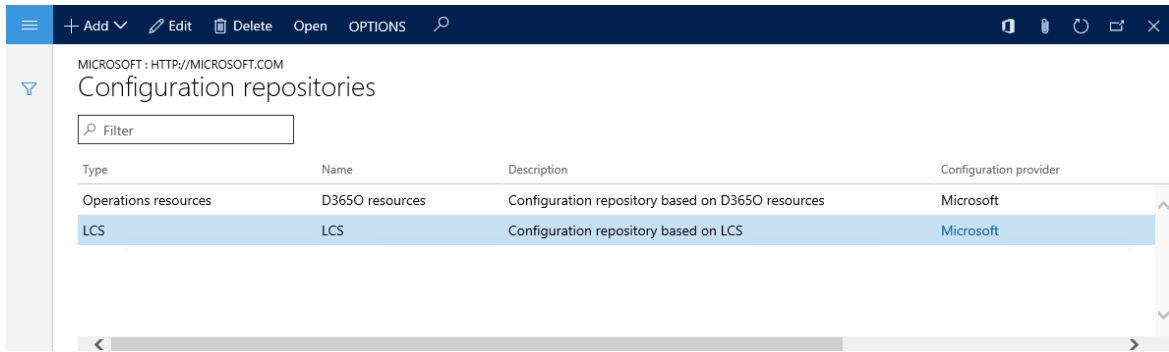
2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to download the newest version of [Electronic reporting \(ER\) configurations](#) from the [Shared asset library](#) in Microsoft Dynamics Lifecycle Services (LCS).

1. Sign in to the application by using one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
2. Go to **Organization administration > Workspaces > Electronic reporting**.
3. In the **Configuration providers** section, select the **Microsoft** tile.
4. On the **Microsoft** tile, select **Repositories**.



5. On the **Configuration repositories** page, in the grid, select the existing repository of the **LCS** type. If this repository doesn't appear in the grid, follow these steps:
  - a. Select **Add** to add a repository.
  - b. Select **LCS** as the repository type.
  - c. Select **Create repository**.
  - d. If you're prompted about authorization, follow the on-screen instructions.
  - e. Enter a name and description for the repository.
  - f. Select **OK** to confirm the new repository entry.
  - g. In the grid, select the new repository of the **LCS** type.
6. Select **Open** to view the list of ER configurations for the selected repository.



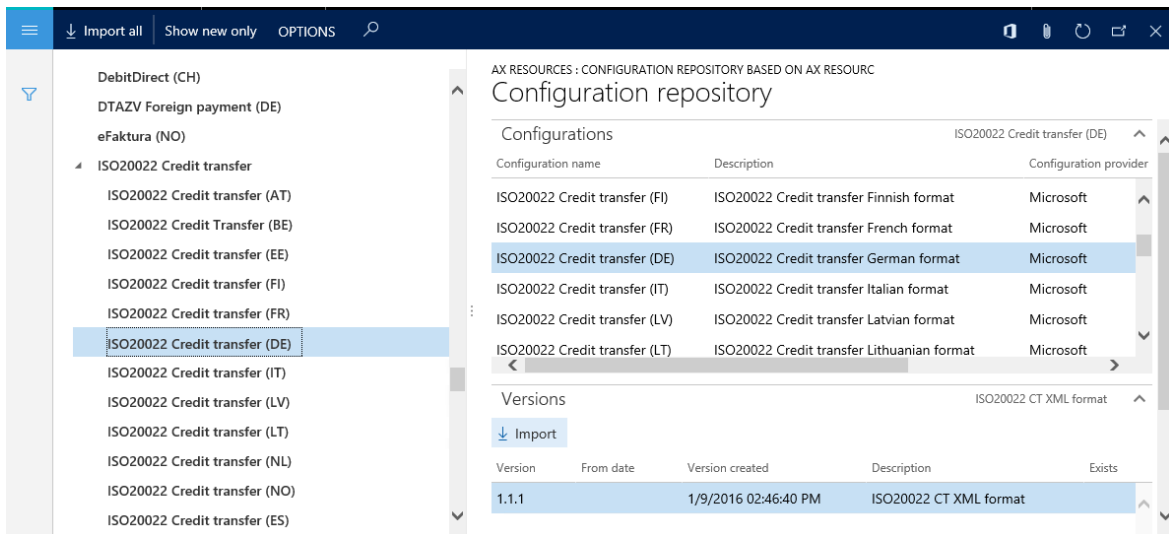
**TIP**

If you have trouble accessing the LCS repository to download configurations from the Shared asset library in LCS, you can download configurations from the [Global repository](#) instead.

7. In the configurations tree in the left pane, select the required ER configuration.
8. On the **Versions** FastTab, select the required version of the selected ER configuration.
9. Select **Import** to download the selected version from LCS to the current instance.

**NOTE**

The **Import** button is unavailable for ER configuration versions that are already present in the current instance.



**NOTE**

Depending on the ER settings, configurations are validated after they are imported. You might be notified about any inconsistency issues that are discovered. You must resolve those issues before you can use the imported configuration version. For more information, see the list of related topics for this topic.

## Additional resources

[Electronic reporting \(ER\) overview](#)

[Download ER configurations from the Global repository of Configuration service](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Import Electronic reporting (ER) configurations

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to download Electronic reporting (ER) configurations from Microsoft Dynamics Lifecycle Services (LCS) to a local business data application. It also explains how to upload the ER configurations from an ER repository to the local business data (LBD) application.

1. Sign in to your local business data application by using one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
2. Go to **Organization administration** > **Electronic reporting**.
3. In the **Configuration providers** section, select the card for the ER provider that is associated with your company.

## NOTE

To learn how to register a new ER solution provider, play the **Create a configuration provider and mark it as active** task guide.

4. On the selected tile, click **Repositories**.
5. On the **Configuration repositories** page, in the grid, select the existing repository of the **File system** type. If the repository doesn't appear in the grid, follow these steps:
  - a. Click **Add** to add a new repository.
  - b. Select **FILE SYSTEM** as the repository type.
  - c. Click **Create repository**.
  - d. Enter a name and description for the repository.
  - e. Enter the path of the working directory for this repository. This path should point to a folder of the local file system where the ER configurations that belong to the repository will be stored.
  - f. Click **OK** to confirm and save the new repository.
  - g. In the grid, select the new repository of the **File system** type.
6. In your browser, open another tab, and sign in to LCS.
7. In the Shared asset library, select the **GER Configuration** asset type, and then click **Download all**.

## NOTE

All the ER configurations will be put into a zip file for download.

8. Open the file, select all the ER configurations, and then copy them to the working directory for the repository of the **File system** type.
9. On the **ER repositories** page, on the **Dynamics 365 for Finance and Operations** tab, click **Open** to view the list of ER configurations for the selected repository.
10. In the **Configurations** tree in the left pane, select an ER configuration.

11. On the **Versions** FastTab, select the required version of the ER configuration.
12. Click **Import** to download the selected version from this repository to the current instance.

**NOTE**

The **Import** button is unavailable for existing ER configuration versions.

**NOTE**

Depending on the ER settings, configurations are validated after they are imported. You might be notified about inconsistencies or issues that are discovered. You must resolve these inconsistencies or issues before you can use the imported configuration version.

## Frequently asked questions

**Question:** When I click **Download all** in the Shared asset library, I receive the following warning: "Zip generation is in progress, please try again in a few minutes." Why do I receive this warning?

**Answer:** You receive this warning because a new configuration is added to the Shared asset library, and the ER configuration is being archived.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Import Electronic reporting (ER) configurations from Regulatory Configuration Services (RCS)

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can use Regulatory Configuration Services (RCS) to design Electronic reporting (ER) configurations. The ER tool provides access to the list of configurations that have been configured in each instance of RCS that has been provisioned for your company. You can use this feature to import configurations that you configured in an RCS instance into the current instance. After configurations are imported, they can be used to handle incoming documents or generate outgoing electronic documents.

To learn more about this feature, complete the example in this topic. Alternatively, play the **ER Import configurations from RCS** task guide, which is part of the 7.5.4.3 Acquire/Develop IT service/solution components (10677) business process. This task guide can be downloaded from the [Microsoft Download Center](#). It walks you through the process of importing ER configurations from an RCS instance into the current instance.

## Example: Import an ER configuration from RCS

This example shows how a user in the System Administrator or Electronic Reporting Developer role can import a new version of an ER configuration from RCS. In this example, you select the desired version of the ER configuration that has been configured in an RCS instance, and you import that version into the current instance for a sample company that is named Litware, Inc. These steps can be completed in any company, because ER configurations are shared among companies.

To complete the steps in this example, you must first complete the steps in [Create configuration providers and mark them as active](#). You must also have access to an RCS instance that contains at least one ER configuration that has a status of either **Completed** or **Shared**.

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. On the **Localization configurations** page, in the **Configuration providers** section, make sure that the configuration provider for the Litware, Inc. sample company is listed, and that it's marked as **Active**. If you don't see this configuration provider, follow the steps in [Create configuration providers and mark them as active](#).
3. If no RCS environment has been provisioned for your company, in the **External Links** section, select **Regulatory services – Configuration**. Then follow the instructions to provision an RCS environment.
4. In the **Related links** section, select **Electronic reporting parameters**.
5. On the **Electronic reporting parameters** page, select the **RCS** tab.
6. Use the URLs on this tab to access the RCS environment has been provisioned for your company.
7. Close the **Electronic reporting parameters** page.

### Register a new ER repository

1. On the **Localization configurations** page, select the **Litware, Inc.** configuration provider in the list.
2. Select **Repositories**.
3. Select **Add** to open the drop-down dialog box.
4. Select **RCS** as the configuration repository type, and then select **Create repository**.
5. In the **RCS environment display name** field, select the desired RCS instance. Note that you can have several instances.
6. Select **OK**.

## Import ER configurations from an RCS-based repository

1. On the **Configuration repositories** page, select the **Show filters** button on the left side of the window.
2. For the **Name** filter, select **begins with** as the filter operator, and then enter **RCS** as the filter value.
3. Select the repository, and open it.
4. On the **Connect to Regulatory Configuration Services** page, select the **Click here to connect to Regulatory Configuration Services** link.
5. Select **Open**.
6. Select **Close**.
7. Select the desired version of the ER configuration, and then select **Import** to import that version.

## Additional resource

- [Electronic reporting \(ER\) overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# (ER) Import configurations from RCS

2/18/2021 • 2 minutes to read • [Edit Online](#)

The following steps explain how a user in the System Administrator or Electronic Reporting Developer role can import a new version of an Electronic reporting (ER) configuration from Microsoft Regulatory Configuration Services (RCS). In this example, you will select the version of the ER configuration that has been configured in an RCS instance and import it into the current instance for sample company, Litware, Inc. These steps can be performed in any company because ER configurations are shared among companies. To complete these steps, you must first complete the steps in the topic, [Create configuration providers and mark them as active](#). To complete these steps, you must also have access to an RCS instance containing at least one ER configuration in either **Completed** or **Shared** status.

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Make sure that the configuration provider for the sample company, Litware, Inc., is available and marked as **Active**. If you don't see this configuration provider, complete the steps in the topic, [Create configuration providers and mark them as active](#).
3. If you have no RCS environment provisioned to your company, select **Regulatory services – Configuration** external link and follow the instructions to provision an RCS environment.
4. Select **Electronic reporting parameters**.
5. Select the **RCS** tab.
6. If RCS environment has been already provisioned to your company, use presented on the page URLs to access it.
7. Close the page.

## Register a new ER repository.

1. In the list, mark the selected row.
2. Select Litware, Inc. provider.
3. Select Repositories.
4. Select Add to open the drop dialog.
5. In the Configuration repository type field, enter 'RCS'.
6. Select Create repository.
7. In the RCS environment display name field, enter or select a value.
8. Select the desired RCS instance. You can have several of them.
9. Select OK.

## Import ER configurations from RCS-based repository

1. Select **Show filters**.
2. Enter a filter value of "RCS" on the **Name** field using the **begins with** filter operator.
3. When you open the selected repository, on the **Connect to Regulatory Configuration Services** page, select **Select here to connect to Regulatory Configuration Services** link.
4. Select **Open**.
5. Select **Close**.
6. Select the desired version of ER configuration and select **Import** to bring it in the current instance.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Use JOIN data sources to get data from multiple application tables in Electronic reporting (ER) model mappings

2/18/2021 • 13 minutes to read • [Edit Online](#)

While configuring Electronic reporting (ER) model mappings or formats, you can [add](#) required data sources of the **Join** type. At design time, a **Join** data source is configured as a set of several data sources each of which returns a list of records. For every data source except the first one, you need to define necessary conditions to join records of the current and previous data sources. At runtime, a configured data source of **Join** type [returns](#) a single joined list of records containing fields from the records of nested data sources.

The following types of joins are currently supported:

- Outer (left) join:
  - Join all records of the first (left-most) data source and then any matching in accordance to configured conditions records of the second (right-most) data source.
- Inner (right) join:
  - Join only records of the first (left-most) data source and only records of the second (right-most) data source matching to each other in accordance to configured conditions.

In the configured **Join** data source, when all data sources are the **Table records** type, execution of the **Join** data source can be [performed at the database level](#) using a single SQL statement. This statement reduces the number of database calls, which improves model-mapping performance. Otherwise, execution of **Join data** source is performed in memory.

## NOTE

Using the **VALUEIN** function in ER expressions that specify conditions for joining records in data sources of **Join** type is not supported yet. Visit the [Formula designer in Electronic reporting](#) page for more details about this function.

To learn more about this feature, complete the example in this topic.

## Example: Use JOIN data sources in ER model mappings

The following steps explain how the System administrator or Electronic reporting developer can configure an Electronic reporting (ER) model mapping to get data from multiple application tables at once by using data sources of the **Join** type to improve data access performance. These steps can be performed for any company of Dynamics 365 Finance or Regulatory Configuration Services (RCS).

### Prerequisites

To complete the examples in this topic, you must have access to one of the following depending on what service is used to complete these steps:

#### Access to Finance for one of the following roles:

- Electronic reporting developer
- Electronic reporting functional consultant
- System administrator

## Access to RCS for one of the following roles:

- Electronic reporting developer
- Electronic reporting functional consultant
- System administrator

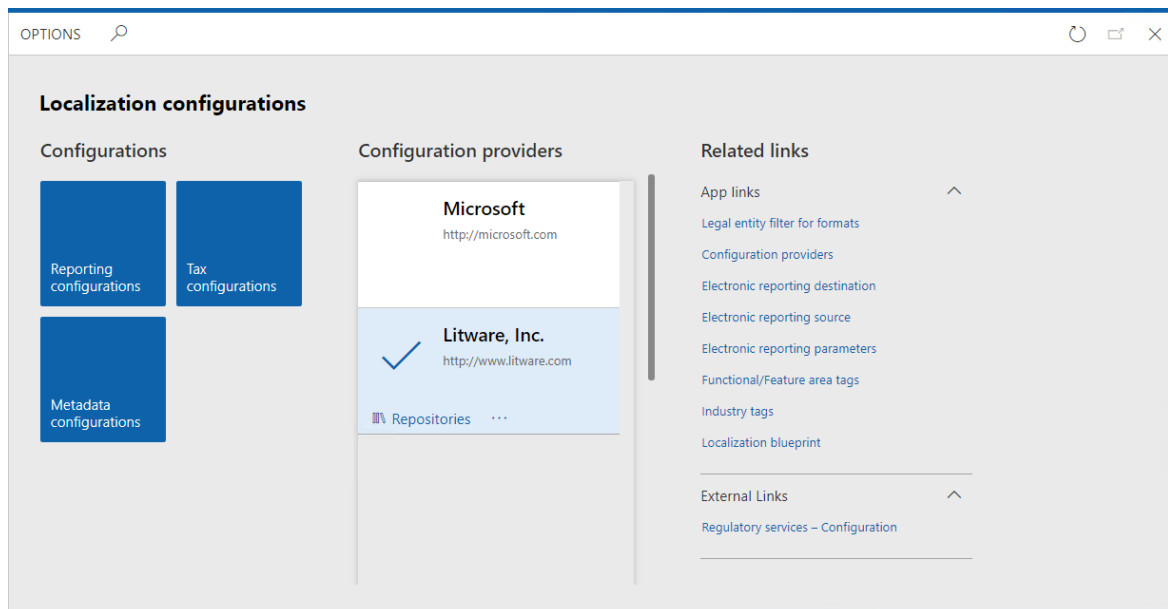
You also must first complete the steps in the [Create a configuration provider and mark it as active](#) procedure.

In advance, you must also download from [Microsoft Download Center](#) and save locally the following sample ER configuration files:

CONTENT DESCRIPTION	FILE NAME
Sample ER <b>data model</b> configuration file, which is used as the data source for the examples.	<a href="#">Model to learn JOIN data sources.version.1.1.xml</a>
Sample ER <b>model mapping</b> configuration file, which implements the ER data model for the examples.	<a href="#">Mapping to learn JOIN data sources.version.1.1.xml</a>
Sample ER <b>format</b> configuration file. This file describes the data to populate the ER format component for the examples.	<a href="#">Format to learn JOIN data sources.version.1.1.xml</a>

## Activate a configurations provider

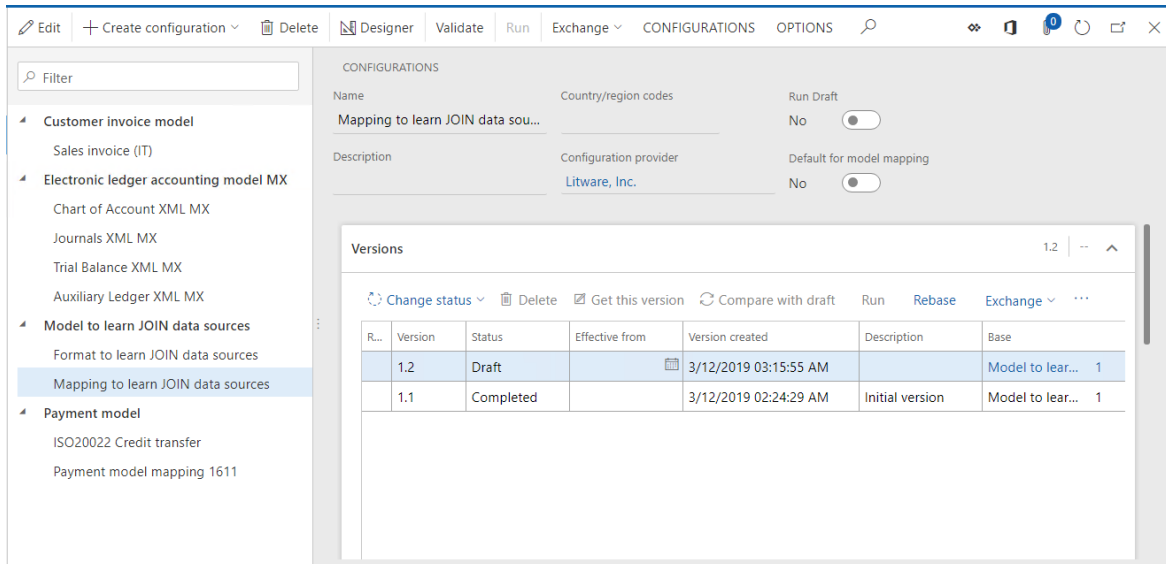
1. Access either Finance or RCS in the first session of your web browser.
2. Go to **Organization administration > Workspaces > Electronic reporting**.
3. On the **Localization configurations** page, in the **Configuration providers** section, make sure that the configuration provider for the [Litware, Inc.](#) sample company is listed, and that it's marked as **Active**. If you don't see this configuration provider, follow the steps in [Create a configuration provider and mark it as active](#) procedure.



## Import sample ER configuration files

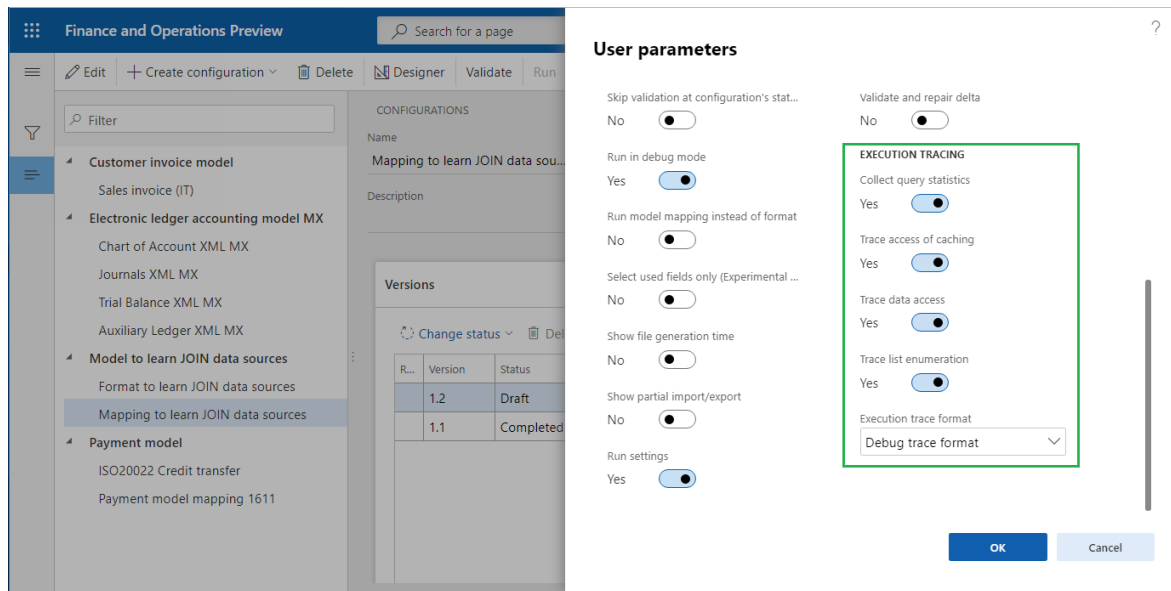
1. Select **Reporting configurations**.
2. Import the ER data model configuration file.
  - a. Select **Exchange**.
  - b. Select **Load from XML file**.

- c. Select **Browse** to find the **Model to learn JOIN data sources.version.1.1.xml** file.
  - d. Select **OK**.
3. Import the ER model-mapping configuration file.
    - a. Select **Exchange**.
    - b. Select **Load from XML file**.
    - c. Select **Browse** to find the **Mapping to learn JOIN data sources.version.1.1.xml** file.
    - d. Select **OK**.
  4. Import the ER format configuration file.
    - a. Select **Exchange**.
    - b. Select **Load from XML file**.
    - c. Select **Browse** to find the **Format to learn JOIN data sources.version.1.1.xml** file.
    - d. Select **OK**.
  5. In the configurations tree, expand the **Model to learn JOIN data sources** item as well as other model items (when available).
  6. Observe the list of ER configurations in the tree as well as version details on the **Versions** fast tab – they will be used as the source of data for your sample report.



### Turn on execution trace options

1. Select **CONFIGURATIONS**.
2. Select **User parameters**.
3. Set execution trace parameters as shown on the screenshot below.



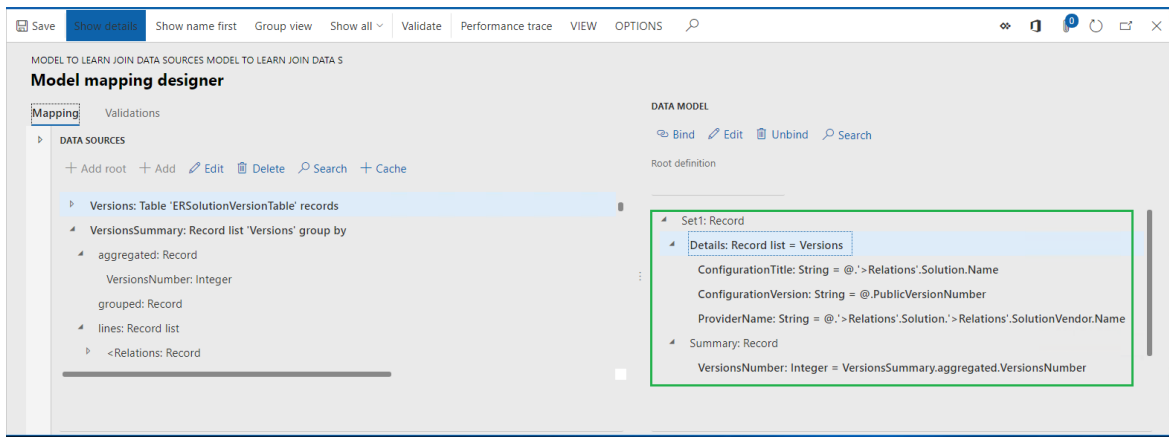
With these parameters turned on, for every execution of the imported ER format file, the execution trace will be generated. Using details of generated execution trace, you can analyze the execution of ER format and ER model-mapping components. Visit the [Trace execution of ER format to troubleshoot performance issues](#) page for more details about ER execution trace feature.

### Review ER model mapping (part 1)

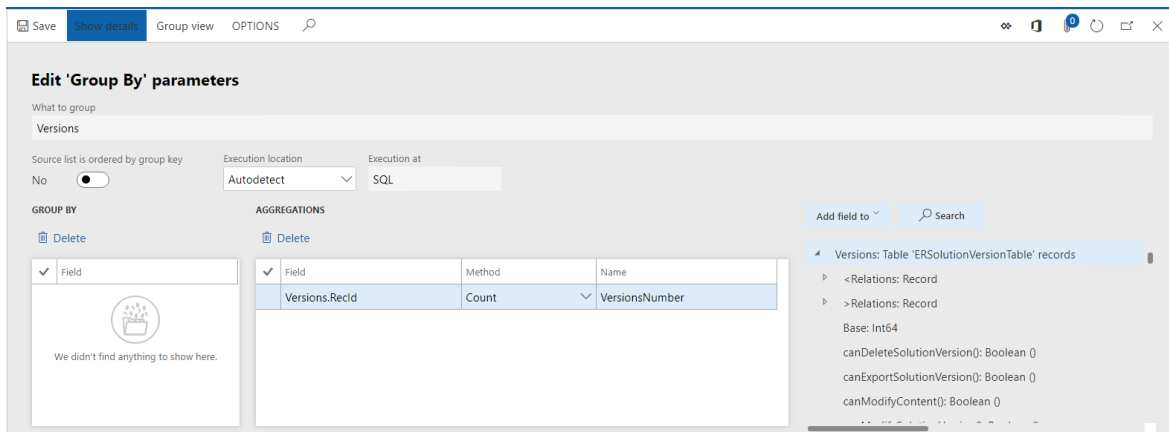
Review settings of the ER model-mapping component. The component is configured to access information about versions of ER configurations, details of configurations and configuration providers without using data sources of the **Join** type.

1. Select **Mapping to learn JOIN data sources** configuration.
2. Select **Designer** to open the list of mappings.
3. Select **Designer** to review the mapping details.
4. Select **Show details**.
5. In the configurations tree, expand the **Set1** and **Set1.Details** data model items:
  - a. Binding **Details: Record list = Versions** indicates that the **Set1.Details** item is bound to the **Versions** data source returning records of the **ERSolutionVersionTable** table. Each record of this table represents a single version of an ER configuration. The content of this table is presented in the **Versions** fast tab on the **Configurations** page.
  - b. Binding **ConfigurationVersion: String = @.PublicVersionNumber** means that the value of the public version of each ER configuration's version is taken from the **PublicVersionNumber** field of the **ERSolutionVersionTable** table and placed to the **ConfigurationVersion** item.
  - c. Binding **ConfigurationTitle: String = @.'>Relations'.Solution.Name** indicates that the name of an ER configuration is taken from the **Name** field of the **ERSolutionTable** table assessing by using the many-to-one relation ('>Relations') between the **ERSolutionVersionTable** and **ERSolutionTable** tables. Names of ER configurations of the current application instance are presented in the configurations tree on the **Configurations** page.
  - d. Binding **@.'>Relations'.Solution.'>Relations'.SolutionVendor.Name** means that the name of the configuration provider that owns the current configuration is taken from the **Name** field of the **ERVendorTable** table assessing by using the many-to-one relation between **ERSolutionTable** and **ERVendorTable** tables. Names of ER configuration providers are presented in the configurations tree on the **Configurations** page on the page header for each configuration. The entire list of ER configuration providers can be found on the **Organization administration > Electronic reporting > Configuration provider** table page.





6. In the configurations tree, expand the Set1.Summary data model item:
  - a. Binding **VersionsNumber: Integer = VersionsSummary.aggregated.VersionsNumber** indicates that the Set1.Summary.VersionsNumber item is bound to the VersionsNumber aggregation field of the VersionsSummary data source of the GroupBy type that was configured to return the number of records of the ERSolutionVersionTable table via the Versions data source.

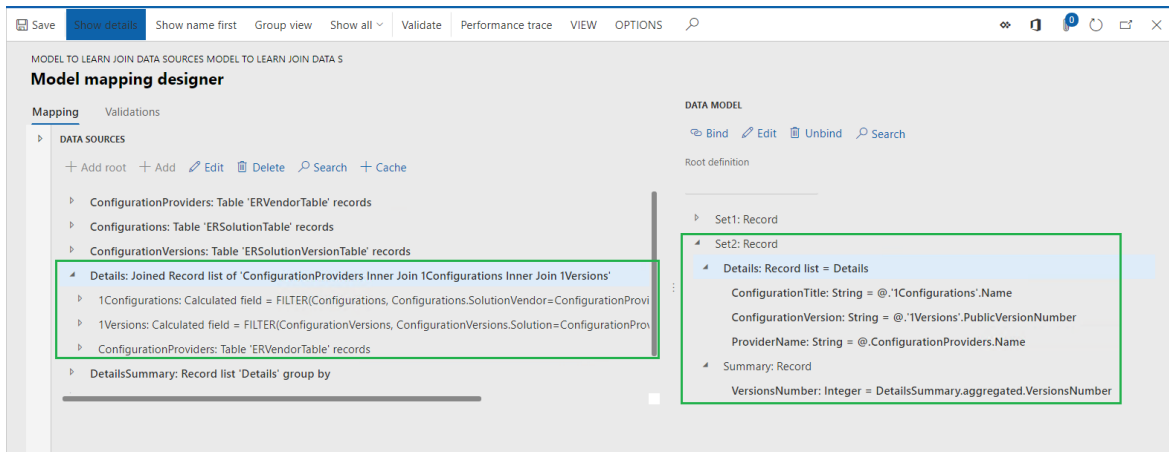


7. Close the page.

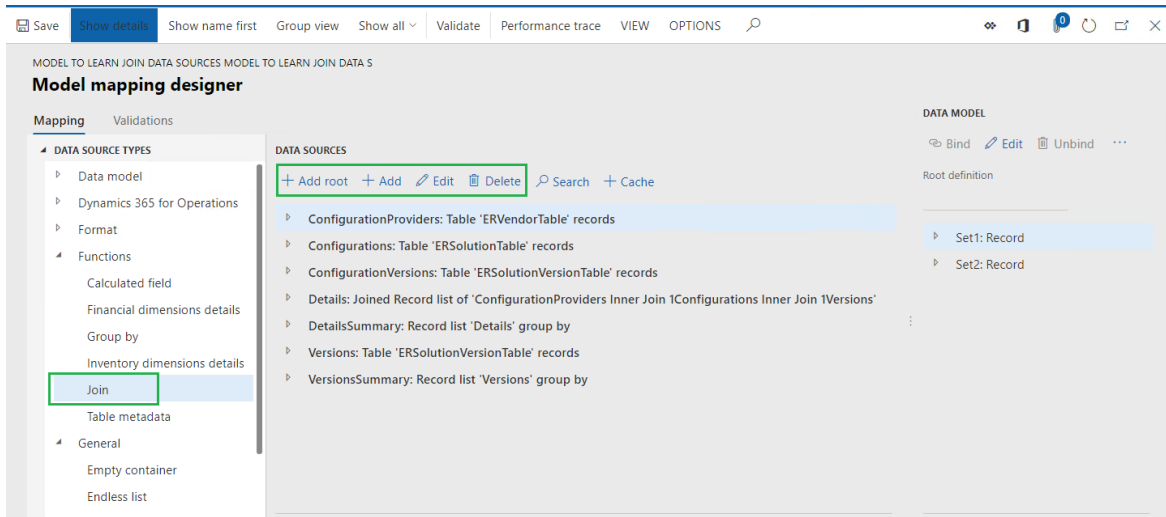
## Review ER model mapping (part 2)

Review settings of the ER model-mapping component. The component is configured to access information about versions of ER configurations, details of configurations and configuration providers with using a data source of the Join type.

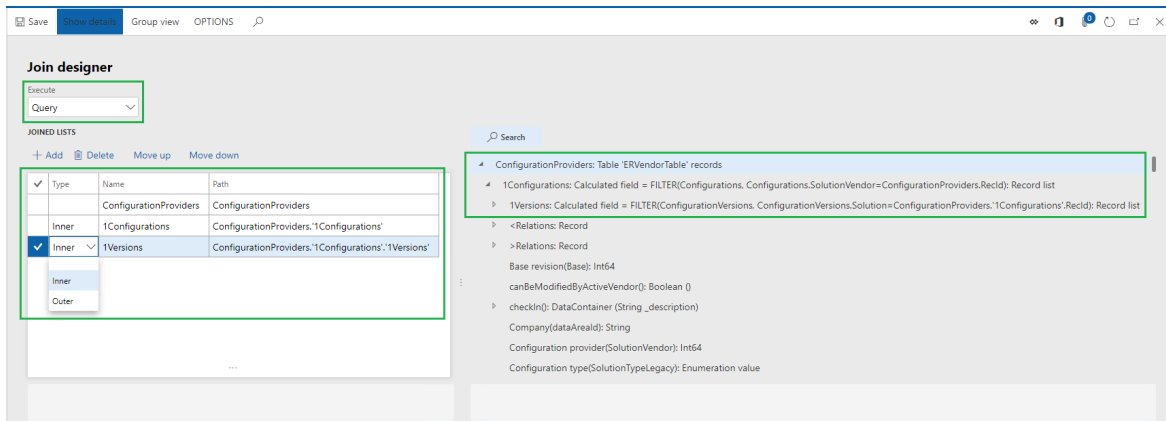
1. In the configurations tree, expand the Set2 and Set2.Details data model items. The binding **Details: Record list = Details** indicates that the Set2.Details item is bound to the Details data source configured as the data source of the Join type.



The Join data source can be added by selecting the Functions\Join data source:



2. Select **Details** data source.
3. Select **Edit** in the **Data sources** pane.
4. Select **Edit join**.
5. Select **Show details**.



This page is used to design the required data source of the **Join type**. At runtime, this data source will create a single joined list of records from the data sources in the **Joined list** grid. Join of records will start from the **ConfigurationProviders** data source that is in the grid as a first one (the **Type** column is blank for it). Records of every other data source will be joined consequently to records of the parent data source based on its order in this grid. Every joining data source must be configured as a data source nested under a target data source ( **1Versions** data source is nested under **1Configurations** one; **1Configurations** data source is nested under **ConfigurationProviders** one). Each configured data source must contain the conditions for the join. In the data source for this particular **Join**, the following joins are defined:

- Each record of the **ConfigurationProviders** data source (referred to the **ERVendorTable** table) is joined with only records of the **1Configurations** one (referred to in the **ERSolutionTable** table) having the same value in the **SolutionVendor** and **ReclId** fields. The **Inner join** type is used for this join as well as the following conditions for matching records:

FILTER (Configurations, Configurations.SolutionVendor = ConfigurationProviders.ReclId)

- Each record of the **1Configurations** data source (referred to the **ERSolutionTable** table) is joined with the only records of the **1Versions** one (referred to the **ERSolutionVersionTable** table) having the same value in the **Solution** and **ReclId** fields. **Inner join** type is used for this join as well as the following conditions for matching records:

FILTER (ConfigurationVersions, ConfigurationVersions.Solution =

ConfigurationProviders.'1Configurations'.ReclId)

- **Execute** option is configured as **Query** meaning that this join data source will be executed at runtime on database level as a direct SQL call.

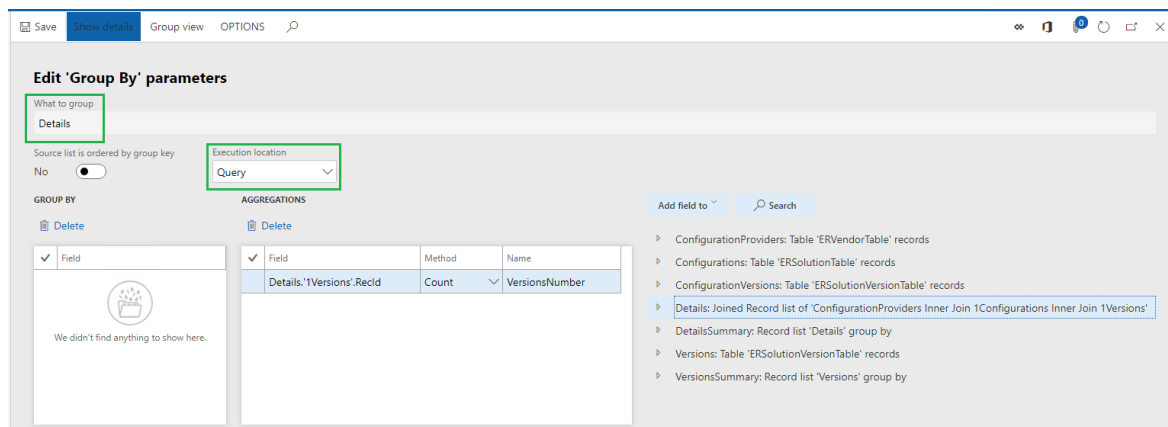
For joining records of data sources representing application tables, you can specify join conditions by using pairs of fields other than ones that describe existing in AOT relations between these tables. This type of join can be configured to execute at the database level as well.

6. Close the page.

7. Select **Cancel**.

8. In the configurations tree, expand the **Set2.Summary** data model item:

- Binding **VersionsNumber: Integer = DetailsSummary.aggregated.VersionsNumber** indicates that the **Set2.Summary.VersionsNumber** item is bound to the **VersionsNumber** aggregation field of the **DetailsSummary** data source of the **GroupBy** type that was configured to return the number of joined records of the **Details** data source of the **Join** type.
- The **Execution** location option is configured as **Query** meaning that this **GroupBy** data source will be run at runtime as a direct SQL call at the database level. This behavior is possible because the base data source **Details** of the **Join** type is configured as executed at the database level.



9. Close the page.

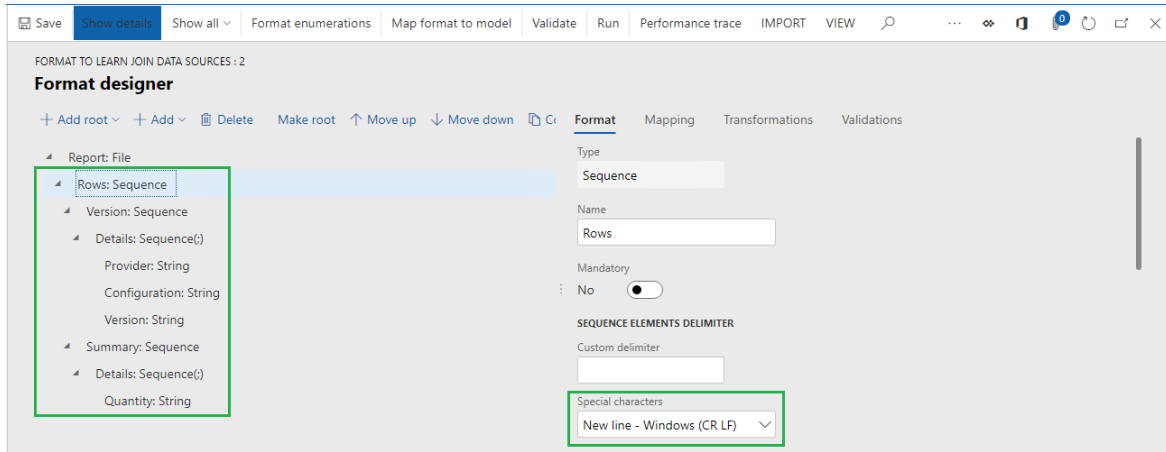
10. Select **Cancel**.

### Execute ER format

1. Access Finance or RCS in the second session of your web browser using same credentials and company as in the first session.
2. Go to **Organization administration > Electronic reporting > Configurations**.
3. Expand **Model** to learn **JOIN data sources** configuration.
4. Select **Format** to learn **JOIN data sources** configuration.
5. Select **Designer**.
6. Select **Show details**.
7. Select **Mapping**.
8. Select **Expand/Collapse**.

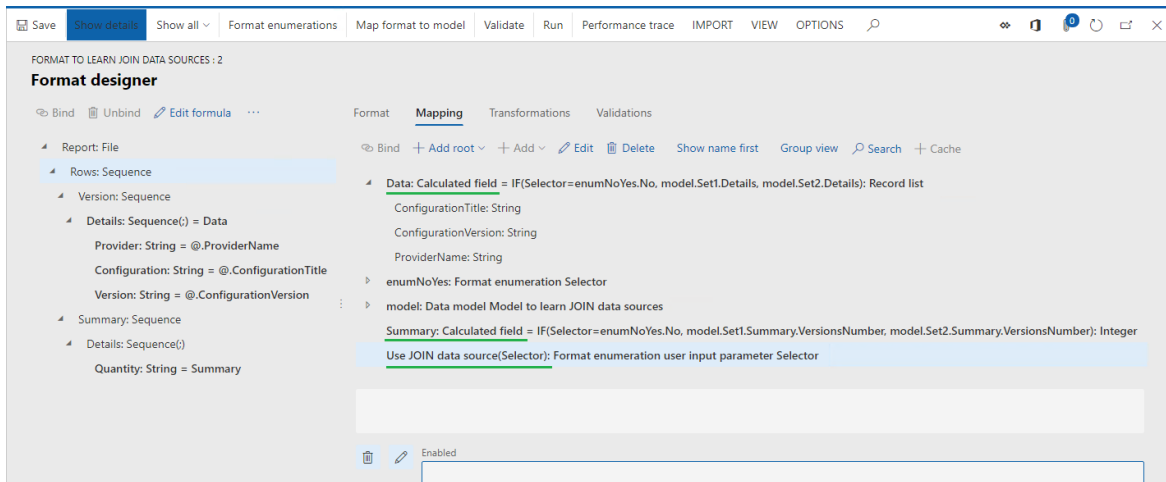
This format is designed to populate a generated text file with a new line for every version of an ER configuration (**Version** sequence). Each generated line will contain the name of a configuration provider owning the current configuration, the configuration name, and the configuration version separated by semicolon mark. The final line of generated file will contain the number of discovered versions of ER

configurations (Summary sequence).



The **Data** and **Summary** data sources are used to populate configuration version details to the generated file:

- Information from the **Set1** data model is used when you choose **No** for the **Selector** data source at runtime on the user dialog page when running ER format.
- Information from the **Set2** data model is used when you choose **Yes** for the **Selector** data source at runtime on the user dialog page.

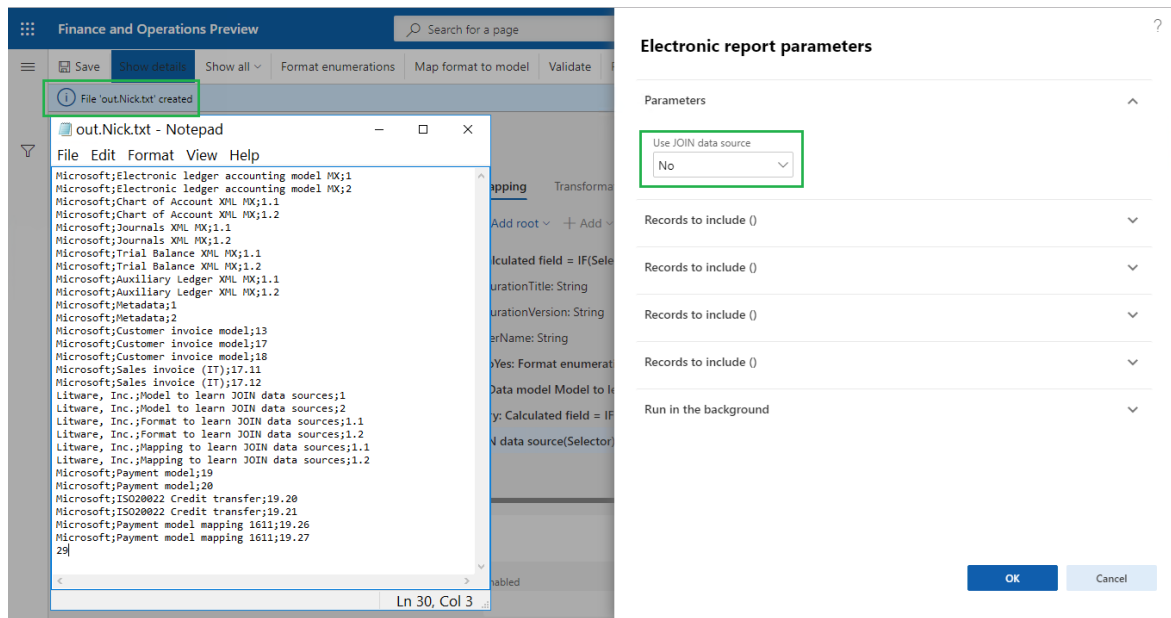


9. Select **Run**.

10. On the dialog page, select **No** in the **Use JOIN data source** field.

11. Select **OK**.

12. Review generated file.

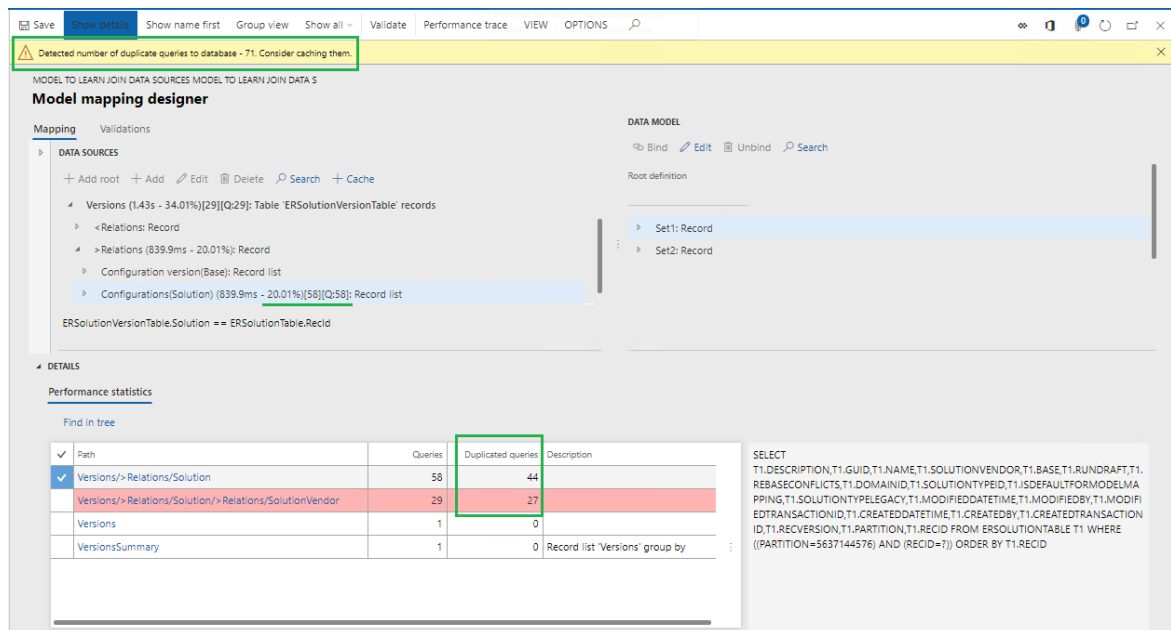


### Analyze ER format execution trace

1. In the first session of Finance or RCS, select **Designer**.
2. Select **Performance trace**.
3. In the **Performance trace** grid, select the top-most record of the latest execution trace of an ER format that used the current model mapping component.
4. Select **OK**.

Execution statistics informs you about duplicated calls to application tables:

- **ERSolutionTable** has been called as many times as you have configuration version records in the **ERSolutionVersionTable** table, while the number of such calls could be reduced in times for performance improvement.
- **ERVendorTable** has been called twice for every configuration version record that was discovered in the **ERSolutionVersionTable** table, while the number of such calls could be reduced as well.

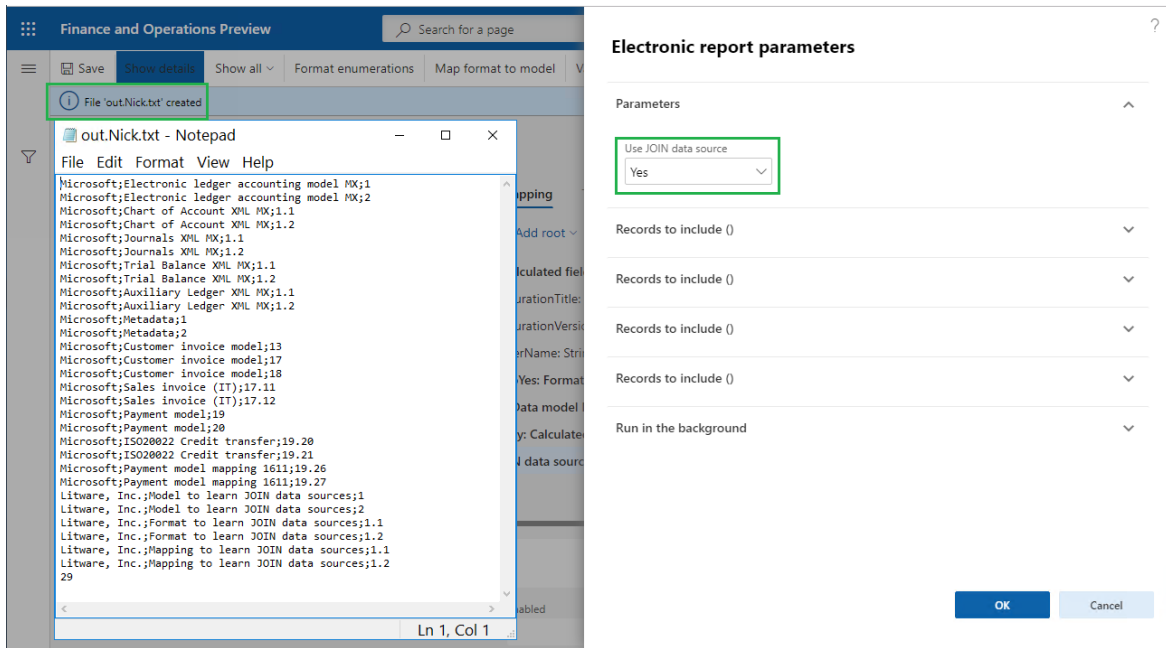


5. Close the page.

### Execute ER format

1. Switch to your web browser tab with the second session of Finance or RCS.

2. Select **Run**.
3. On the dialog page, select **Yes** in the **Use JOIN data source** field.
4. Select **OK**.
5. Review generated file.

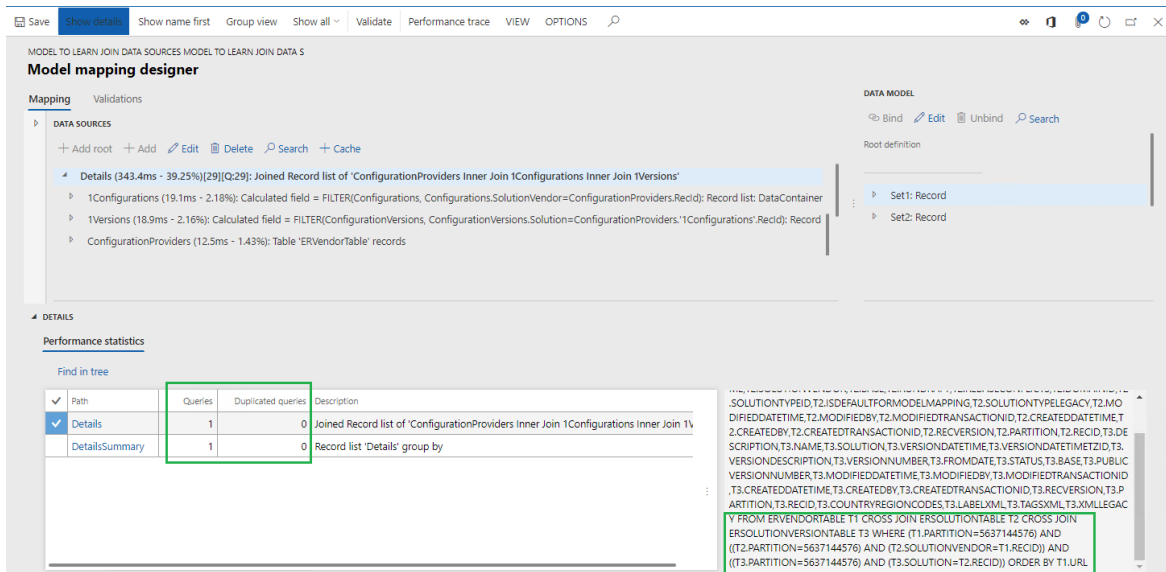


### Analyze ER format execution trace

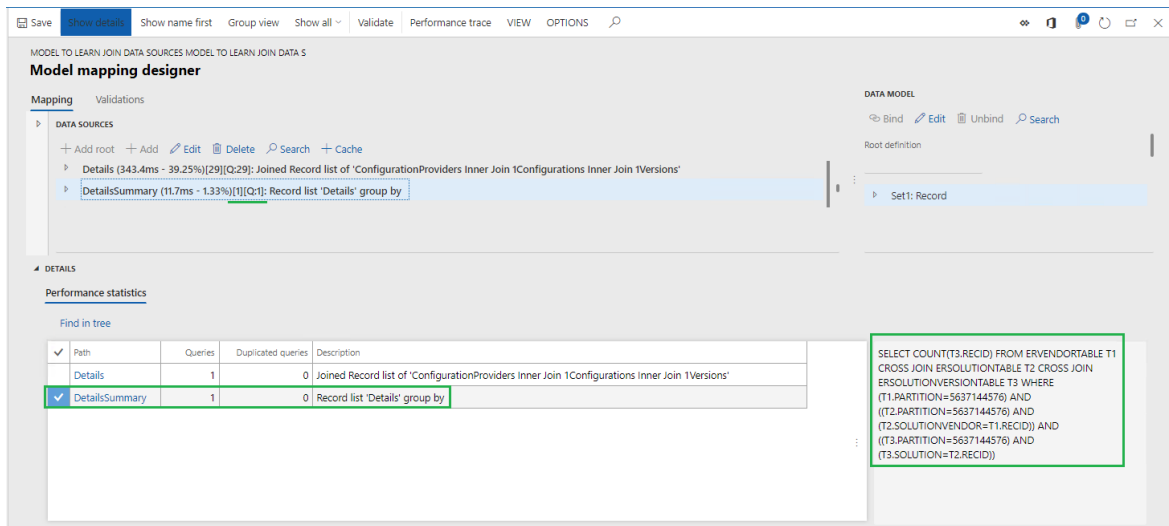
1. In the first session of Finance or RCS, select **Designer**.
2. Select **Performance trace**.
3. In the **Performance trace** grid, select top-most record representing the latest execution trace of an ER format that used the current model mapping component.
4. Select **OK**.

Statistics informs you about the following:

- Application database has been called once to get records from **ERVendorTable**, **ERSolutionTable**, and **ERSolutionVersionTable** tables to access required fields.



- Application database has been called once to calculate the number of configuration versions by using joins that were configured in the **Details** data source.



## Limitations

As you can see from the example in this topic, the **JOIN** data source can be built from several data sources that describe the individual datasets of the records that must eventually be joined. You can configure those data sources by using the built-in ER **FILTER** function. When you configure the data source so that it's called beyond the **JOIN** data source, you can use company ranges as part of the condition for data selection. The initial implementation of the **JOIN** data source doesn't support data sources of this type. For example, when you call a **FILTER**-based data source within the scope of execution of a **JOIN** data source, if the called data source contains company ranges as part of the condition for data selection, an exception occurs.

In Microsoft Dynamics 365 Finance version 10.0.12 (August 2020), you can use company ranges as part of the condition for data selection in **FILTER**-based data sources that are called within the scope of execution of a **JOIN** data source. Because of the limitations of the application **query** builder, the company ranges are supported only for the first data source of a **JOIN** data source.

### Example

For example, you must make a single call to the application database to get the list of foreign trade transactions of multiple companies and the details of the inventory item that is referred to in those transactions.

In this case, you configure the following artifacts in your ER model mapping:

- **Intrastat** root data source that represents the **Intrastat** table.
- **Items** root data source that represents the **InventTable** table.
- **Companies** root data source that returns the list of companies (**DEMF** and **GBSI** in this example) where transactions must be accessed. The company code is available from the **Companies.Code** field.
- **X1** root data source that has the expression `FILTER (Intrastat, VALUEIN(Intrastat.dataAreaId, Companies, Companies.Code))`. As part of the condition for data selection, this expression contains the definition of company ranges `VALUEIN(Intrastat.dataAreaId, Companies, Companies.Code)`.
- **X2** data source as a nested item of the **X1** data source. It includes the expression `FILTER (Items, Items.ItemId = X1.ItemId)`.

Finally, you can configure a **JOIN** data source where **X1** is the first data source and **X2** is the second data source. You can specify **Query** as the **Execute** option to force ER to run this data source on the database level as a direct SQL call.

When the configured data source is run while the ER execution is **traced**, the following statement is shown in the ER model mapping designer as part of the ER performance trace.

```
SELECT ... FROM INTRASTAT T1 CROSS JOIN INVENTTABLE T2 WHERE ((T1.PARTITION=? ) AND (T1.DATAAREAID IN (N'DEMF',N'GBSI' ) ) ) AND ((T2.PARTITION=? ) AND (T2.ITEMID=T1.ITEMID AND (T2.DATAAREAID = T1.DATAAREAID) AND (T2.PARTITION = T1.PARTITION))) ORDER BY T1.DISPATCHID,T1.SEQNUM
```

#### **NOTE**

An error occurs if you run a **JOIN** data source that has been configured so that it contains data selection conditions that have company ranges for additional data sources of the executed **JOIN** data source.

## Additional resources

[Formula designer in Electronic reporting](#)

[Trace execution of ER format to troubleshoot performance issues](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure country context dependent ER model mappings

2/18/2021 • 19 minutes to read • [Edit Online](#)

You can configure Electronic reporting (ER) model mappings so that they implement a generic ER data model but are specific to Dynamics 365 Finance. This topic explains how to design multiple ER model mappings for an ER data model to control how they are used by corresponding ER formats that are run from companies that have different country/region contexts.

## Prerequisites

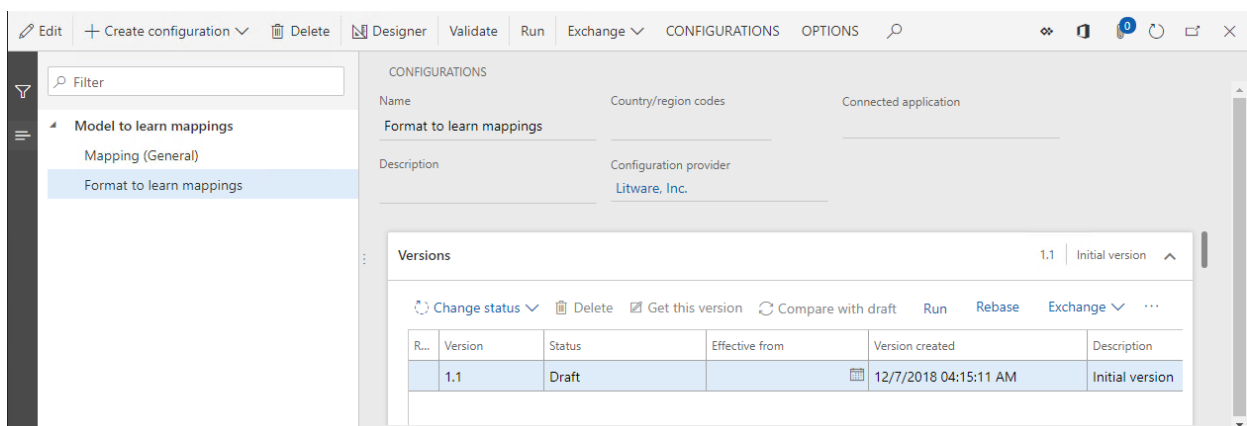
To complete the examples in this topic, you must have the following access:

- Access to Finance for one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
- Access to the instance of Regulatory Configuration Services (RCS) that has been provisioned for the same tenant as Finance for one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator

Some steps in this topic require execution of an ER format. In some cases, execution of an ER format is affected by the country/region context of the company that you're currently signed in to. You can run an ER format in the current RCS instance if the company that has the required country/region context is available in RCS. Otherwise, you must upload a completed version of the ER model mapping and ER format configurations that use the ER data model to your Finance instance, and then run the ER format in that Finance instance. For information about how to import configurations that reside in RCS into a Finance instance, see [Import configurations from RCS](#).

## Single model mapping case

Follow the steps in [Appendix 1](#) of this topic to design the required ER components. You now have the **Mapping (General)** model mapping configuration that contains the model mapping for the **Entry point 1** definition.



The screenshot shows the Dynamics 365 configuration interface. The left sidebar contains a navigation menu with 'Model to learn mappings' expanded, showing 'Mapping (General)' and 'Format to learn mappings'. The main area displays the configuration for 'Format to learn mappings'. The configuration details include:

- Name: Country/region codes
- Country/region codes: Connected application
- Description: Configuration provider
- Configuration provider: Litware, Inc.

Below the configuration details is a 'Versions' section showing a table of configuration versions:

R...	Version	Status	Effective from	Version created	Description
	1.1	Draft		12/7/2018 04:15:11 AM	Initial version

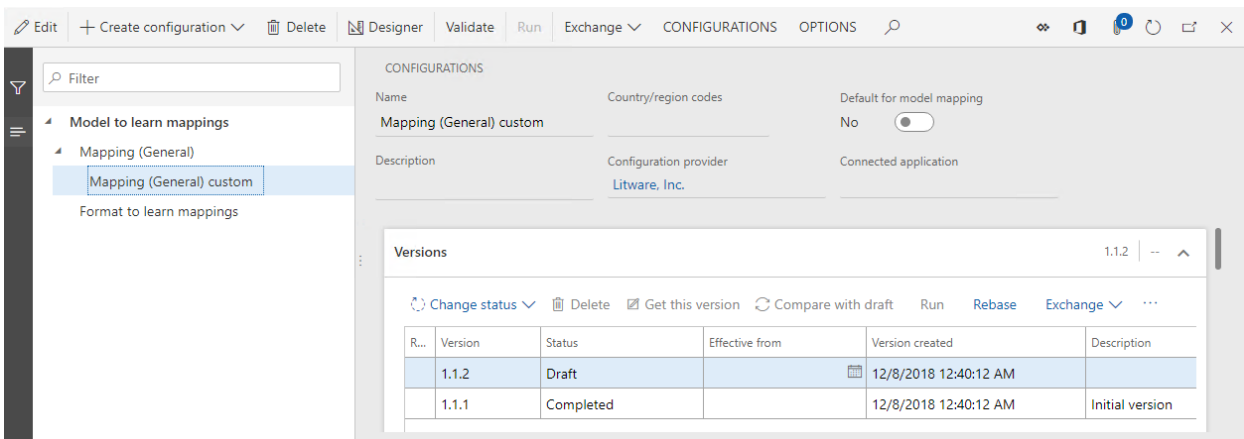
### Run the configured format

1. On the **Configurations** page, on the **Versions** FastTab, select **Run**.
2. Select **OK**.

Notice that the web browser offers to download the text file that was generated by executed ER format. Because this format was configured to use the **Entry point 1** definition, and only a single model mapping is currently available for the base model that contains a mapping for this definition, the executed ER format used the **Mapping (General)** model mapping of the **Mapping (General)** configuration as a data source. Therefore, the downloaded file contains the **Generic functionality 1** text.

## Multiple shared model mappings case

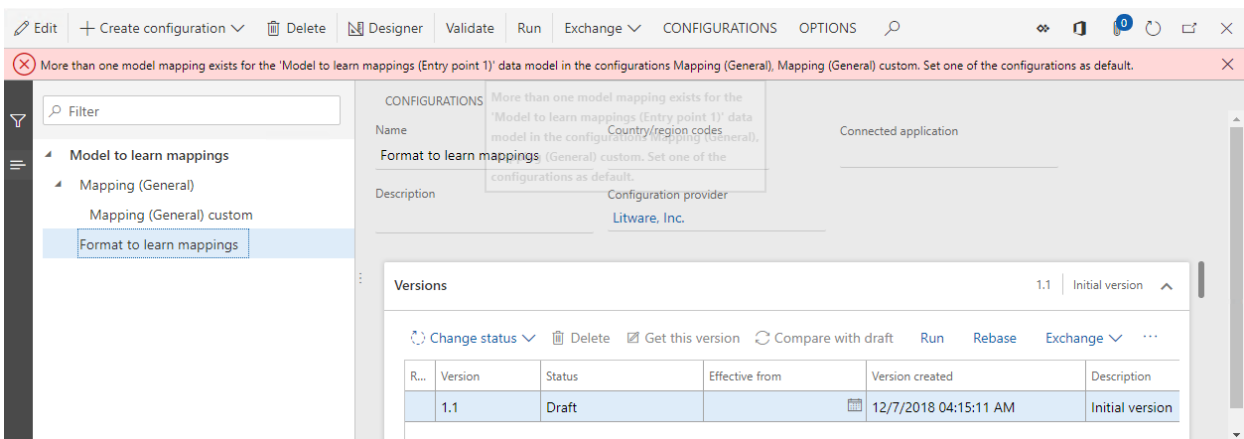
Follow the steps in [Appendix 2](#) of this topic to design the required ER components. You now have **Mapping (General)** and **Mapping (General) custom** model mapping configurations, each of which contains the model mapping for the **Entry point 1** definition.



### Run the configured format

1. On the **Configurations** page, in the configurations tree, select **Format to learn mappings**.
2. On the **Versions** FastTab, select **Run**.
3. Select **OK**.

Notice that execution of the selected ER format is unsuccessful. An error message informs you that more than one model mapping exists for the **Model to learn mappings** model and the **Entry point 1** definition in the **Mapping (General)** and **Mapping (General) custom** model mapping configurations. The message also recommends that you select one of those configurations as the default configuration.

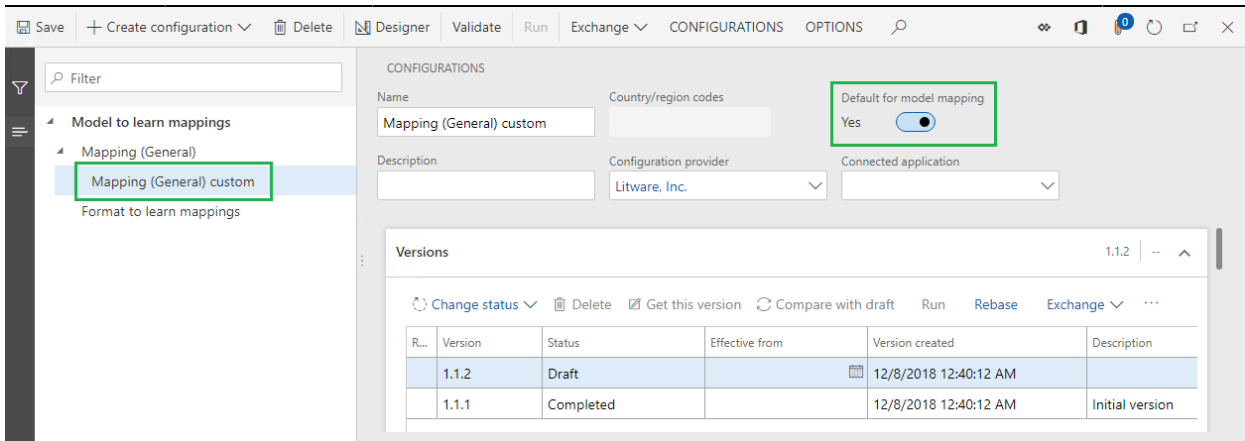


### Define a default mapping configuration

Follow these steps to define the **Mapping (General) custom** model mapping configuration as the default configuration, so that its mappings can be used as data sources for the **Format to learn mappings** ER format.

1. On the **Configurations** page, in the configurations tree, select **Mapping (General) custom**.

2. As required, select **Edit** to make the current page ready for editing.
3. Set the **Default for model mapping** option to **Yes**.
4. Select **Save**.



### Run the configured format

1. On the **Configurations** page, in the configurations tree, select **Format to learn mappings**.
2. On the **Versions** FastTab, select **Run**.
3. Select **OK**.

Notice that execution of the selected ER format succeeds. The web browser offers to download the text file that was generated by executed ER format. Because this format was configured to use the **Entry point 1** definition, and the **Mapping (General) custom** model mapping configuration was selected as the default configuration, the executed ER format used the **Mapping (General) copy** model mapping of the **Mapping (General) custom** configuration as a data source. Therefore, the downloaded file contains the **Generic functionality 1 custom** text.

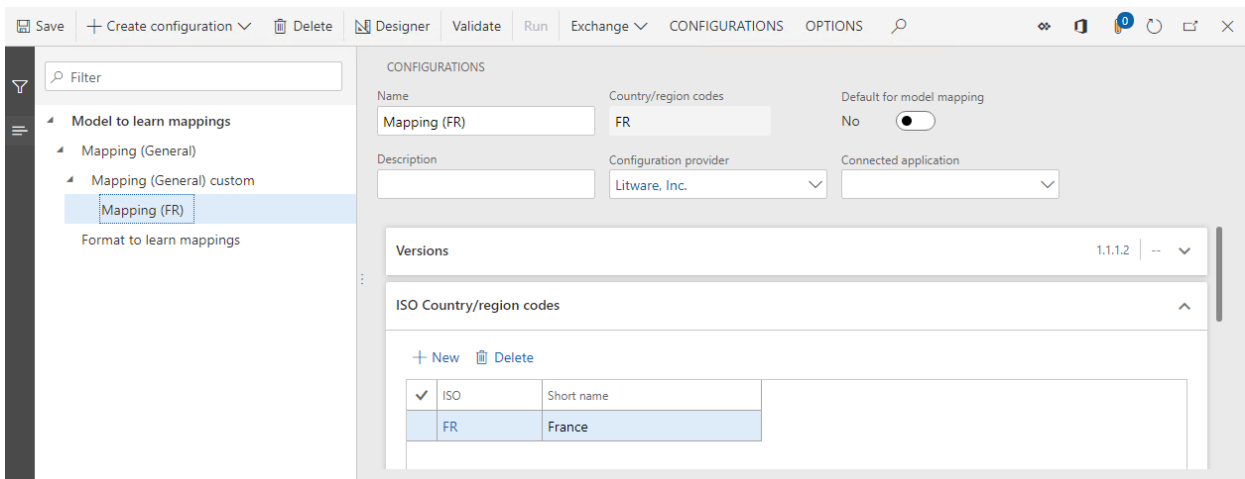
#### NOTE

If you change the company that you're currently signed in to and run this ER format again, you get the same content in the generated file, because the default ER model mapping configuration doesn't contain any company-dependent restrictions.

## Multiple mixed model mappings case

Follow the steps in [Appendix 3](#) of this topic to design the required ER components. You now have **Mapping (General)**, **Mapping (General) custom**, and **Mapping (FR) model mapping** configurations that contain the model mapping for the **Entry point 1** definition.

Notice that version 1 of the **Mapping (FR) model mapping** configuration is configured so that it applies only to ER formats of the **Model to learn mappings** model that are run in Finance companies that have French country/region context.



### Run the configured format

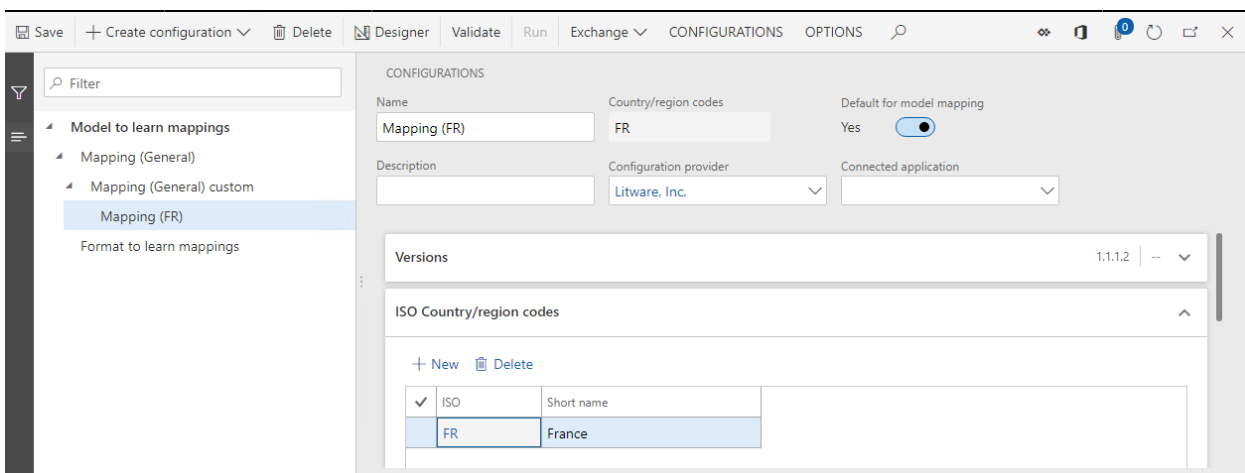
1. Change the company to FRSI.
2. On the **Configurations** page, in the configurations tree, select **Format to learn mappings**.
3. On the **Versions** FastTab, select **Run**.
4. Select **OK**.

Notice that execution of the selected ER format succeeds. The web browser offers to download the text file that was generated by the executed ER format. Because this format was configured to use the **Entry point 1** definition, and the **Mapping (General) custom** model mapping configuration was selected as the default configuration, the executed ER format used the **Mapping (General) copy** model mapping of the **Mapping (General) custom** configuration as a data source. Therefore, the downloaded file contains the **Generic functionality 1** custom text.

### Define the France-specific mapping configuration as the default configuration

Follow these steps to define the custom **Mapping (FR)** model mapping configuration as the default configuration. Note that, because this mapping is specific to France, it will be considered the default mapping between all model mapping configurations that have the **FR** country code specified in the **ISO country/region codes** field.

1. On the **Configurations** page, in the configurations tree, select **Mapping (FR)**.
2. As required, select **Edit** to make the current page ready for editing.
3. Set the **Default for model mapping** option to **Yes**.
4. Select **Save**.



### Run the configured format

1. On the **Configurations** page, in the configurations tree, select **Format to learn mappings**.
2. On the **Versions** FastTab, select **Run**.

3. Select **OK**.

Notice that execution of the selected ER format succeeds. The web browser offers to download the text file that was generated by the executed ER format. Because this format was configured to use the **Entry point 1** definition, and the **Mapping (FR)** model mapping configuration was selected as the default configuration, the executed ER format used the **Mapping (FR)** model mapping of the **Mapping (FR)** configuration as a data source. Therefore, the downloaded file contains the **FR functionality 1** text.

#### NOTE

If you change the company that you're currently signed in to and run this ER format again, the output will depend on the country/region context of the selected company.

## Other model mapping cases

As you've seen, the selection of a model mapping for the execution of an ER format works in the following way:

- The model mapping definition that an ER format uses is specified (**Entry point 1** in the examples in this topic).
- All mapping configurations that contain a mapping that has the specified definition, and that satisfy any country/region context restrictions that are configured, can potentially be used to run the ER format (**Mapping (General)**, **Mapping (General) custom**, and **Mapping (FR)** in the examples in this topic).
- Any default model mapping that has country/region context restrictions has the highest priority for selection (**Mapping (FR)** in the examples in this topic).
- Any default model mapping that doesn't have country/region context restrictions has the next higher priority for selection (**Mapping (General) custom** in the examples in this topic).
- Any model mapping that has country/region context restrictions has higher priority for selection than a model mapping that doesn't have country/region context restrictions.

The following table provides information about the results of model mapping selection for all possible cases for model mapping settings:

- Column 1 indicates whether the first model mapping that doesn't have country/region context restrictions (for example, the shared **Mapping (General)** mapping) is presented and, if it is, whether the **Default for model mapping** option is set to **Yes** for it.
- Column 2 indicates whether the second model mapping that doesn't have country/region context restrictions (for example, the shared **Mapping (General) custom** mapping) is presented and, if it is, whether the **Default for model mapping** option is set to **Yes** for it.
- Column 3 indicates whether the first model mapping that has country/region A context restrictions (for example, the France-specific **Mapping (FR)** mapping) is presented and, if it is, whether the **Default for model mapping** option is set to **Yes** for it.
- Column 4 indicates whether the second model mapping that has country/region A context restrictions is presented and, if it is, whether the **Default for model mapping** option is set to **Yes** for it.
- Column 5 presents the result of a model mapping selection for execution of an ER format under the control of a company that has country/region A context.
- Column 6 presents the result of a model mapping selection for execution of an ER format under the control of a company that has country/region B context.

In the table, a plus sign (+) indicates the presence of a model mapping configuration in the current instance of the Microsoft Azure service that is used to run an ER format (either Finance or RCS).

CASE	MODEL MAPPING 1 WITHOUT COUNTRY/REGION CONTEXT (MM1)	MODEL MAPPING 2 WITHOUT COUNTRY/REGION CONTEXT (MM2)	MODEL MAPPING 1 WITH COUNTRY/REGION A CONTEXT (MM1A)	MODEL MAPPING 2 WITH COUNTRY/REGION A CONTEXT (MM2A)	RUN UNDER CONTROL OF A COMPANY THAT HAS COUNTRY/REGION A CONTEXT	RUN UNDER THE CONTROL OF A COMPANY THAT HAS COUNTRY/REGION B CONTEXT
	1	2	3	4	5	6
1					Error (missing mapping)	Error (missing mapping)
2	+				MM1	MM1
3	+	+			Error (multiple mappings)	Error (multiple mappings)
4	+		+		MM1A	MM1
5	+		+	+	Error (multiple mappings)	MM1
6	+	default	+	+	MM2	MM2
7	+		default		MM1A	MM1
8	+		default	+	MM1A	MM1
9	+		default	default	Error (multiple mappings)	MM1
10	default				MM1	MM1
11	default	+			MM1	MM1
12	default		+		MM1	MM1
13	default	default			Error (multiple mappings)	Error (multiple mappings)
14	default		default		MM1A	MM1
15	default		default	default	MM1A	MM2A
16			+	+	MM1A	MM2A
17			default	default	MM1A	MM2A

## Learn what mapping was used in the execution of an ER format

### Configure ER user parameters

1. On the **Configurations** page, on the Action Pane, on the **CONFIGURATIONS** tab, select **User parameters**.
2. Set the **Run in debug mode** option to **Yes**.

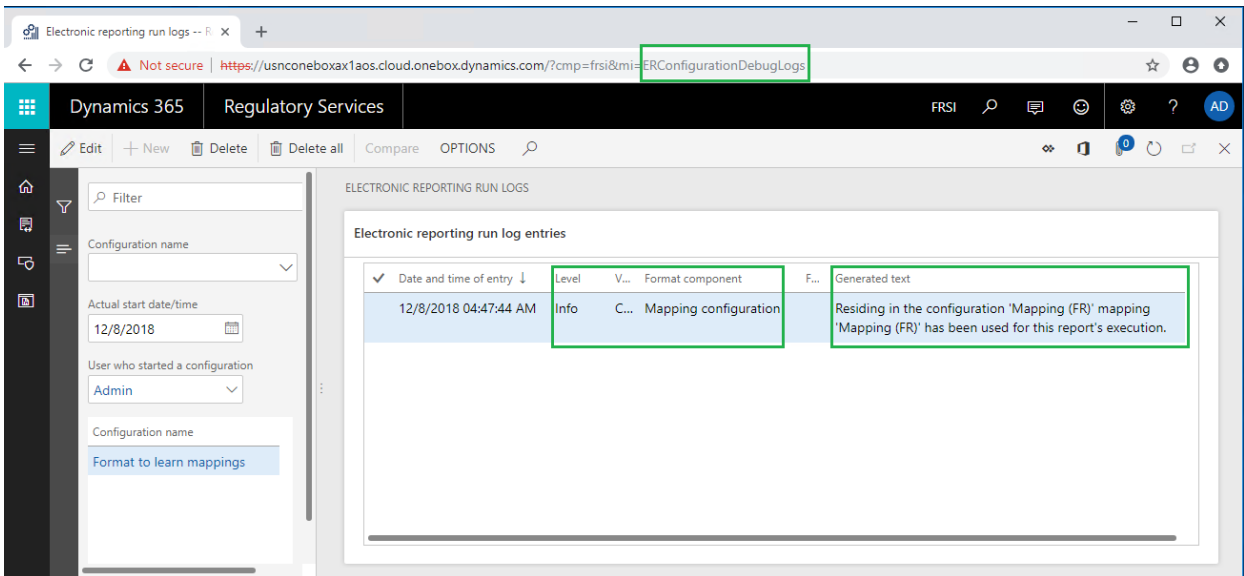
3. Select **Ok**.

### Run the configured format

1. On the **Configurations** page, in the configurations tree, select **Format to learn mappings**.
2. On the **Versions** FastTab, select **Run**.
3. Select **Ok**.

### Review the ER debug log

1. In the navigation pane, go to **Modules > Organization administration > Electronic reporting > Configuration debug log**.
2. Select the **Reload this page** button.



Notice that a new record has been added to the ER debug log for the executed ER format. Because the **Level** field of this record is set to **Info**, the record is informational. Because the **Format component** field is set to **Mapping configuration**, the record informs you about a model mapping that was used during execution of the **Format to learn mappings** ER format (selected in the **Configuration name** field). The content of the **Generated text** field informs you that the **Mapping (FR)** mapping component that resides in the **Mapping (FR)** configuration has been used to run this report.

## Appendix 1

### Configure a sample data model

Sign in to your RCS instance.

In this example, you will create a configuration for sample company, Litware, Inc. To complete these steps, you must first complete, in RCS, the steps in the [Create a configuration provider and mark it as active](#) procedure.

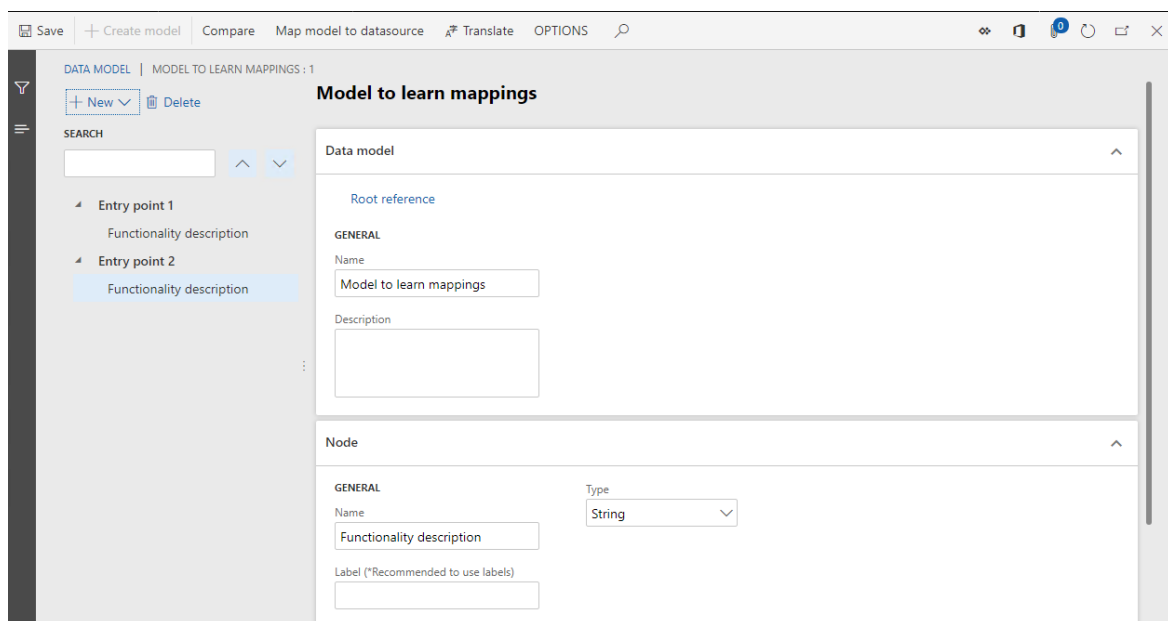
#### Create an ER data model configuration

1. On the default dashboard, select **Electronic reporting**.
2. Select the **Reporting configurations** tile.
3. On the **Configurations** page, select **Create configuration**.
4. In the drop-down dialog box, in the **Name** field, enter **Model to learn mappings**.
5. Select **Create configuration**.
6. Select the **Configuration components** FastTab.

Notice that draft version 1 of this ER configuration is ready for editing. This version contains the data model component.

#### Design a sample data model

1. On the **Configurations** page, select **Designer**.
2. Select **New**.
3. In the drop-down dialog box, in the **Name** field, enter **Entry point 1**.
4. Select **Add**.
5. Select **New**.
6. In the drop-down dialog box, in the **Name** field, enter **Functionality description**.
7. Select **Add**.
8. Select **New**.
9. In the drop-down dialog box, in the **New node** field group, select **Model root**.
10. In the **Name** field, enter **Entry point 2**.
11. Select **Entry point 2**.
12. Select **Add**.
13. Select **New**.
14. In the drop-down dialog box, in the **Name** field, enter **Functionality description**.
15. Select **Add**.



16. Select **Save**.
17. Close the page.

**Complete the modified version of the model configuration**

1. On the **Configurations** page, on the **Versions** FastTab, select **Change status**.

Change the status of designed model configuration from **Draft** to **Completed**, so that it can be used to design the required model mappings and formats.

2. Select **Complete**.
3. Select **OK**.



Notice that the configuration that you created is saved as completed version 1.

### Configure a sample model mapping

#### Create an ER model mapping configuration

1. On the **Configurations** page, select **Create configuration**.
2. In the drop-down dialog box, in the **New** field group, select **Model mapping based on data model Model to learn mappings**.
3. In the **Name** field, enter **Mapping (General)**.
4. In the **Data model definition** field, select **Entry point 1**.
5. Select **Create configuration**.

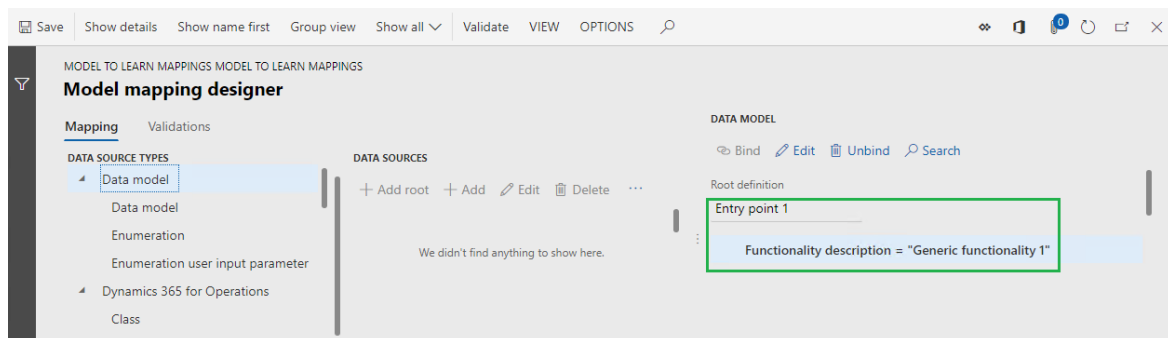
Notice that draft version 1 of this ER configuration is ready for editing. This version contains the model mapping component.

#### Design a sample model mapping

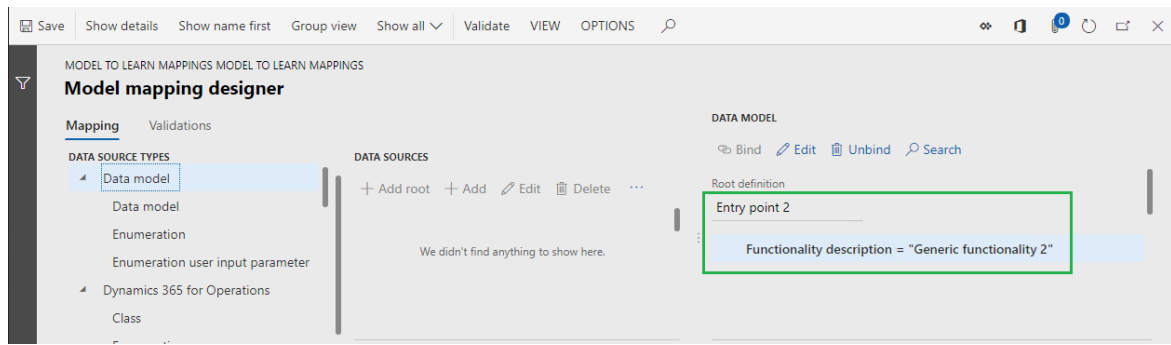
1. On the **Configurations** page, select **Designer**.

Notice that the model mapping of the **To model** direction type has been automatically added to this component for the **Entry point 1** definition.

2. Select **Designer** to start editing the added model mapping.
3. In the **Data model** section, select **Edit**.
4. In the **Formula** field, enter "**Generic functionality 1**".
5. Select **Save**.
6. Close the **Formula designer** page.

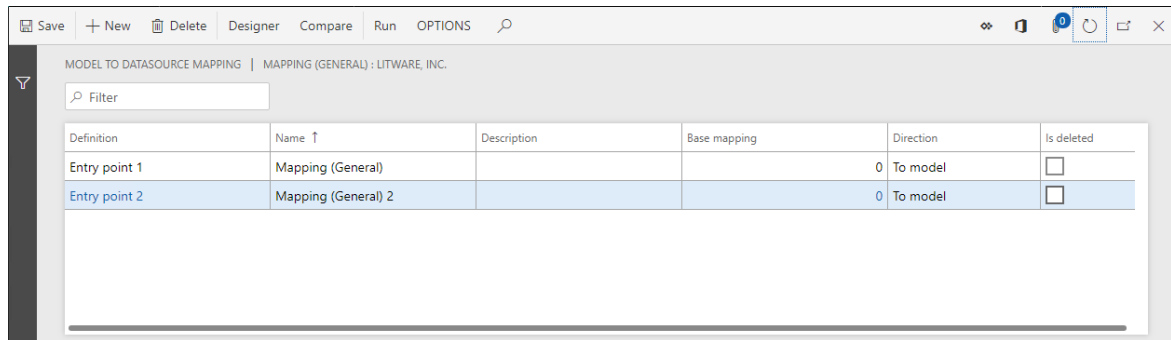


7. Select **Save**.
8. Close the **Model mapping designer** page.
9. Select **New**.
10. In the **Definition** field, select **Entry point 2**.
11. In the **Name** field, enter **Mapping (General) 2**.
12. Select **Designer**.
13. In the **Data model** section, select **Edit**.
14. In the **Formula** field, enter "**Generic functionality 2**".
15. Select **Save**.
16. Close the **Formula designer** page.



17. Select **Save**.

18. Close the **Model mapping designer** page.



19. Close the **Model mappings** page.

#### Complete the modified version of the model mapping configuration

1. On the **Configurations** page, on the **Versions** FastTab, select **Change status**.

Change the status of designed model mapping configuration from **Draft** to **Completed**, so that it can be used by ER formats.

2. Select **Complete**.

3. Select **OK**.

Notice that the configuration that is created is saved as completed version 1.

#### Configure a sample format

##### Create an ER format configuration

1. On the **Configurations** page, in the configurations tree, select **Model to learn mappings**.

2. Select **Create configuration**.

3. In the drop-down dialog box, in the **New** field group, select **Format based on data model Model to learn mappings**.

4. In the **Name** field, enter **Format to learn mappings**.

5. In the **Data model definition** field, select **Entry point 1**.

6. Select **Create configuration**.

Notice that draft version 1 of this ER configuration is ready for editing. This version contains the format component.

##### Design a sample format

1. On the **Configurations** page, select **Designer**.

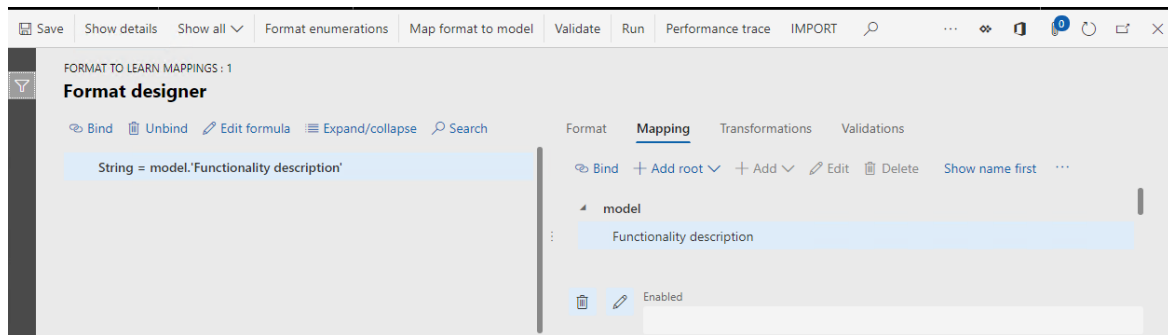
2. Select **Add root**.

3. In the **Text** group, select the **String** item.

4. Select **OK**.

### Bind format elements to a data source

1. On the **Format designer** page, on the **Mapping** tab, expand the model data source.
2. Select the **Functionality description** field.
3. Select **Bind**.



4. Select **Save**.
5. Close the page.

## Appendix 2

### Configure a sample model mapping for general customization

You might want to customize a model mapping that a configuration provider (partner) provided to you, and then use the customized version as a data source for your ER formats. In this case, you must create a custom ER model mapping configuration to make the required changes in existing model mappings. The procedures in this appendix use the **Mapping (General)** model mapping as an example.

#### Create an ER model mapping configuration

1. On the **Configurations** page, in the configurations tree, select **Mapping (General)**.
2. Select **Create configuration**.
3. In the drop-down dialog box, in the **New** field group, select **Derive from Name: Mapping (General), Litware, Inc..**
4. In the **Name** field, enter **Mapping (General) custom**.
5. Select **Create configuration**.

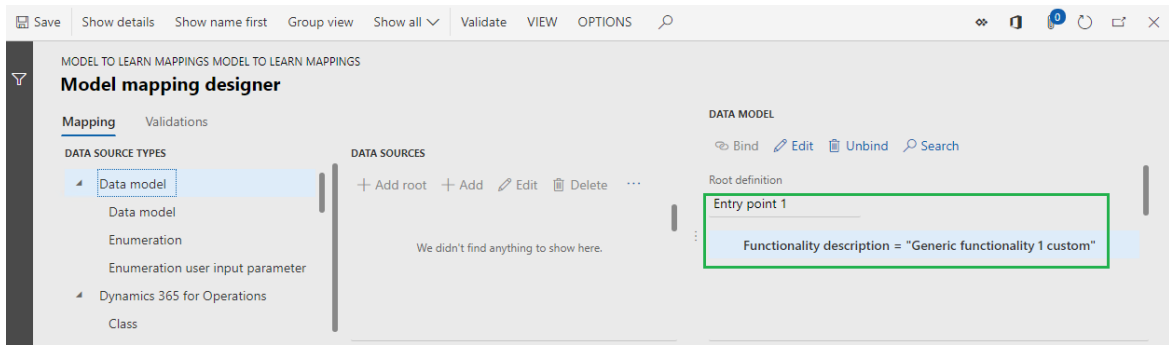
Notice that draft version 1 of this ER configuration is ready for editing.

#### Design a sample model mapping

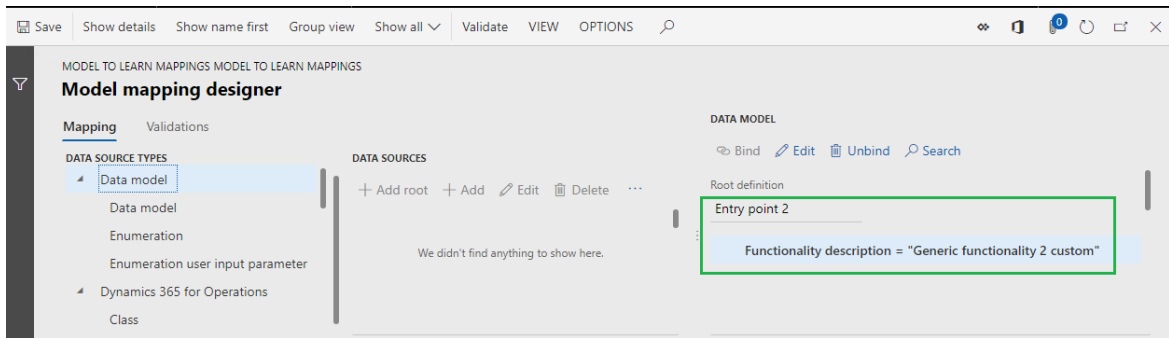
1. On the **Configurations** page, select **Designer**.

Notice that the model mappings of the base configuration have been automatically copied to this configuration.

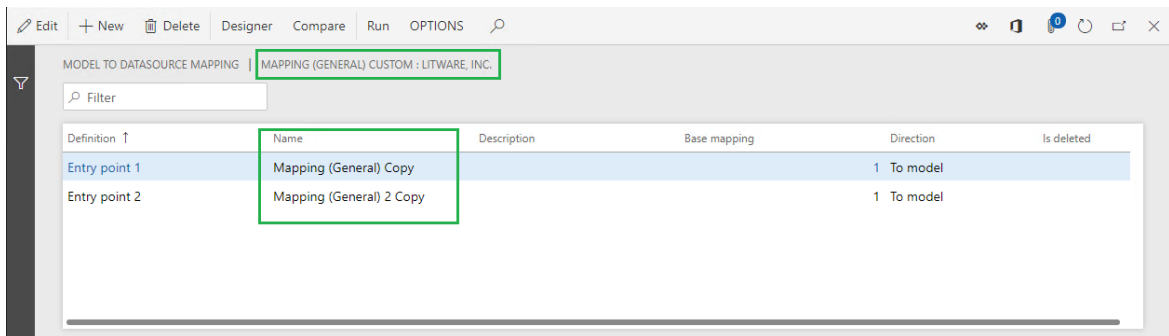
2. Select the **Mapping (General) Copy** mapping.
3. Select **Designer**.
4. In the **Data model** section, select **Edit**.
5. In the **Formula** field, enter **"Generic functionality 1 custom"**.
6. Select **Save**.
7. Close the page.



8. Select **Save**.
9. Close the page.
10. Select the **Mapping (General) 2 Copy** mapping.
11. Select **Designer**.
12. In the **Data model** section, select **Edit**.
13. In the **Formula** field, enter "**Generic functionality 2 custom**".
14. Select **Save**.
15. Close the page.



16. Select **Save**.
17. Close the page.



18. Close the page.

**Complete the modified version of the model mapping configuration**

1. On the **Configurations** page, on the **Versions** FastTab, select **Change status**.

Change the status of designed model mapping configuration from **Draft** to **Completed**, so that it can be used by ER formats.

2. Select **Complete**.

3. Select **OK**.

Notice that the configuration that is created is saved as completed version 1.

## Appendix 3

### Configure a sample model mapping for country/region-specific customization

For some ER formats, there might be country/region-specific requirements for data preparation. In this case, you can manage a separate ER model mapping configuration and isolate the implementation of these country/region-specific requirements from the general implementation. The procedures in this appendix use the **Format to learn mappings** ER format and French-specific requirements as an example.

#### Create an ER model mapping configuration

First, create a new ER model mapping configuration to implement the country/region-specific requirements. Use your custom ER model mapping configuration as a base.

1. On the **Configurations** page, in the configurations tree, select **Mapping (General) custom**.
2. Select **Create configuration**.
3. In the drop-down dialog box, in the **New** field group, select **Derive from Name: Mapping (General) custom, Litware, Inc.**
4. In the **Name** field, enter **Mapping (FR)**.
5. Select **Create configuration**.

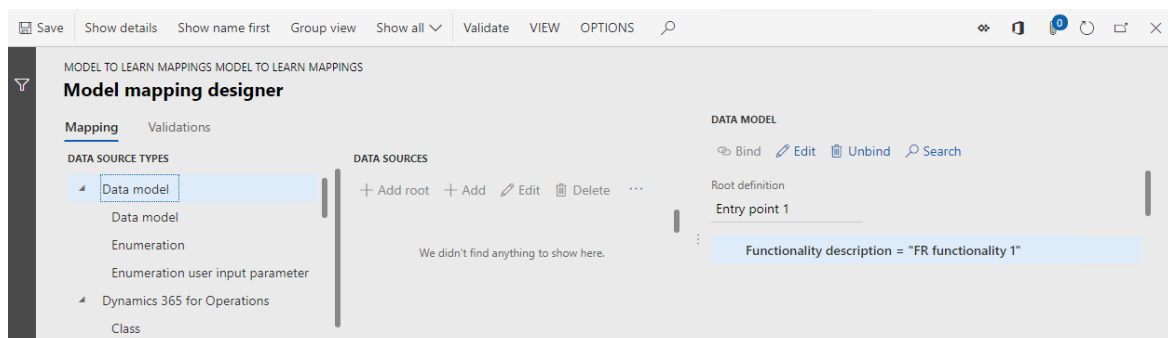
Notice that draft version 1 of this ER configuration is ready for editing.

#### Design a sample model mapping

1. On the **Configurations** page, select **Designer**.

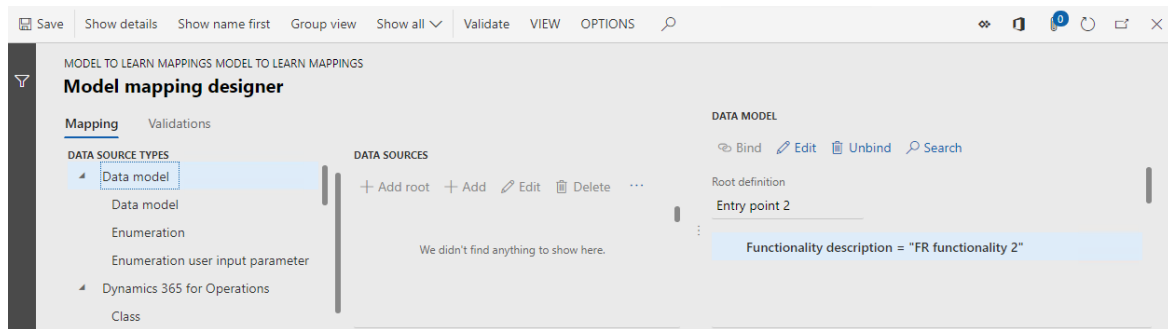
Notice that model mappings of the base configuration have been automatically copied to this configuration.

2. Select the **Mapping (General) Copy Copy** mapping.
3. Rename it **Mapping (FR)**.
4. Select **Designer**.
5. In the **Data model** section, select **Edit**.
6. In the **Formula** field, enter **"FR functionality 1"**.
7. Select **Save**.
8. Close the page.

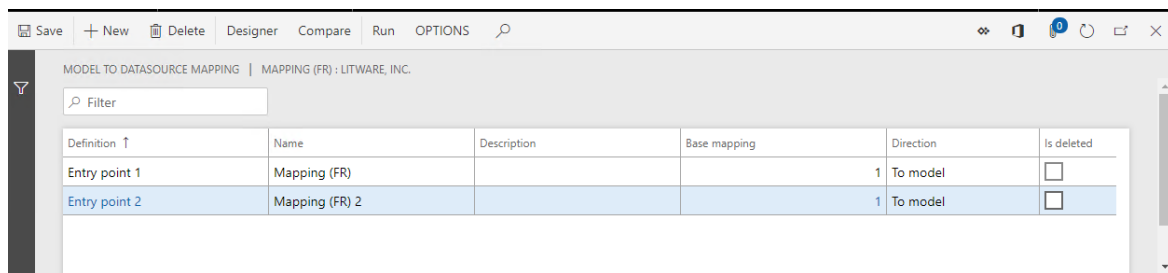


9. Select **Save**.
10. Close the page.

11. Select the **Mapping (General) 2 Copy Copy** mapping.
12. Rename it **Mapping (FR) 2**.
13. Select **Designer**.
14. In the **Data model** section, select **Edit**.
15. In the **Formula** field, enter "**FR functionality 2**".
16. Select **Save**.
17. Close the page.



18. Select **Save**.
19. Close the page.



20. Close the page.

**Specify country/region context restrictions for use**

1. On the **Configurations** page, on the **ISO Country/region codes** FastTab, select **New**.
2. In the **ISO** field, select **FR**.
3. Select **Save**.

Note that you must sign in to a specific company in Finance to run an ER format. Therefore, this company can be considered a party that controls both ER format execution and selection of the correct ER model mapping of the base ER data model. By adding the **FR** country code, you specify that this model mapping is available for selection by an ER format of the base data model only when that format is run under the control of a company that has French country/region context.

You can add multiple country/region codes for a single version of an ER model mapping configuration. In this way, model mappings that reside in that model mapping configuration can be used for an ER format that is run under the control of companies that have a different country/region context.

Note that the list of country/region codes is specified for each version of an ER model mapping configuration and can vary from version to version.

**Complete the modified version of the model mapping configuration**

1. On the **Configurations** page, on the **Versions** FastTab, select **Change status**.

Change the status of designed model mapping configuration from **Draft** to **Completed**, so that it

can be used by ER formats.

2. Select **Complete**.
3. Select **OK**.

Notice that the configuration that is created is saved as completed version 1.

## Additional resources

[Electronic reporting \(ER\) overview](#)

[Manage ER model mapping in separate ER configurations](#)

[Apply country/region context](#)

## Frequently asked questions

**I configured two shared ER model mapping configurations in RCS and marked one of them as the default model mapping configuration. I successfully ran an ER format that was created for the same base ER data model configuration, to test model mappings. I then imported the whole ER solution (ER data model, two ER model mapping configurations, and ER format configuration) into Finance. Why do I receive an error message when I try to run the same ER format in Finance?**

The default model mapping setting is environment-specific. It's configured in RCS but isn't exported to Finance. To successfully run this ER format, you must mark one of ER model mapping configurations as the default model mapping configuration in Finance too.

**I configured one model mapping as a shared model mapping and completed the draft version of it. I then added a new model mapping configuration for same data model and configured it as French-specific. Why is the shared model mapping selected when I run an ER format, even though this ER format uses the correct root definition and execution is done under the control of the company that has French country/region context?**

Make sure that the shared model mapping configuration isn't marked as the default model mapping configuration. Otherwise, it will have higher priority during mapping selection. Also make sure that the French-specific model mapping configuration is considered when a mapping is selected during ER format execution. An ER model mapping configuration is available for selection only if at least one of the following conditions is met:

- At least one version of the ER model mapping configuration has either **Completed** or **Shared** status. In this case, the version that has the highest version number will be used for ER format execution.
- The **Run draft** option for the ER model mapping configuration is turned on. In this case, the version that has **Draft** status will be used for ER format execution.

The **Run draft** option becomes available on the **Configurations** page for each ER model mapping configuration when the **Run setting** ER user parameter is turned on.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure ER formats to use parameters that are specified per legal entity

2/18/2021 • 13 minutes to read • [Edit Online](#)

## Overview

In many of the Electronic reporting (ER) formats that you will design, you must filter data by using a set of values that are specific to each legal entity of your instance (for example, a set of tax codes to filter tax transactions). Currently, when filtering of this type is configured in an ER format, values that are dependent on the legal entity (for example, tax codes) are used in expressions of the ER format to specify data filtering rules. Therefore, the ER format is made legal entity-specific, and to generate the required reports, you must create derived copies of the original ER format for each legal entity where you have to run the ER format. Each derived ER format must be edited to bring legal entity-specific values into it, rebased whenever the original (base) version has been updated, exported from a test environment and imported into a production environment when it must be deployed for production use, and so on. Therefore, maintenance of this type of configured ER solution is quite complex and time-consuming for several reasons:

- The more legal entities there are, the more ER format configurations must be maintained.
- Maintenance of ER configurations requires that business users have ER knowledge.

The ER application-specific parameters feature lets power users configure data filtering in an ER format so that it's based on a set of abstract rules. This set of rules can be configured to use the data sources that are available in an ER format. Business users can then specify real rules beyond the ER framework by using the user interface (UI) that is automatically generated based on the settings of the corresponding ER format and the current legal entity data that will be accessed by the ER format's data sources. The set of rules that is specified for an ER format can be exported from the current legal entity of the Dynamics 365 Finance (Finance) instance. It can then be imported into another legal entity of either the same Finance instance or a different instance as a set of rules for the same ER format.

## Prerequisites

To complete the examples in this topic, you must have access to the instance of Regulatory Configuration Services (RCS) that has been provisioned for the same tenant as Finance for one of the following roles:

- Electronic reporting developer
- Electronic reporting functional consultant
- System administrator

We recommend that you complete the steps in the [Support parameterized calls of ER data sources of CALCULATED FIELD type](#) topic. If you've already completed those steps, you can skip the steps in the **Import ER configurations into RCS** section that follows.

## Import ER configurations into RCS

From [Microsoft Download Center](#), download the **Support parameterized calls of ER data sources of CALCULATED FIELD type** zip file. This zip file contains the following ER configurations that must be extracted and stored locally.



CONTENT DESCRIPTION	FILE NAME
Sample ER <b>data model</b> configuration file	Model to learn parameterized calls.version.1.xml
Sample ER <b>metadata</b> configuration file	Metadata to learn parameterized calls.version.1.xml
Sample ER <b>model mapping</b> configuration file	Mapping to learn parameterized calls.version.1.1.xml
Sample ER <b>format</b> configuration	Format to learn parameterized calls.version.1.1.xml

Next, sign in to your RCS instance.

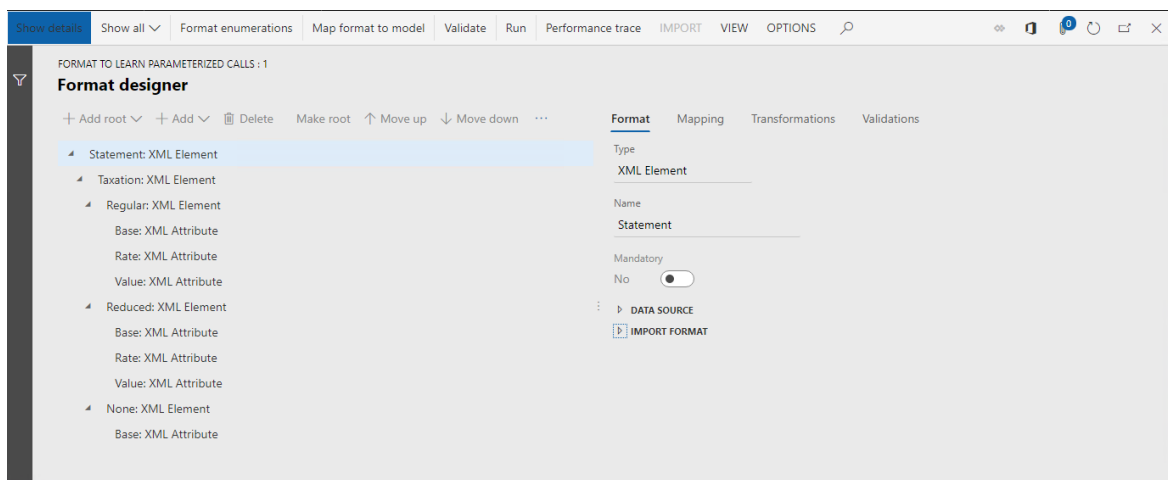
In this example, you will create a configuration for the Litware, Inc sample company. Before you can complete this procedure, you must complete the steps in the [Create a configuration provider and mark it as active](#) topic in RCS.

1. On the default dashboard, select **Electronic reporting**.
2. Select **Reporting configurations**.
3. Import the ER configurations that you downloaded earlier into RCS, in the following order: data model, metadata, model mapping, and format. For each ER configuration, follow these steps:
  - a. Select **Exchange**.
  - b. Select **Load from XML file**.
  - c. Select **Browse** to select the file for the required ER configuration in XML format.
  - d. Select **OK**.

## Review the ER solution that is provided

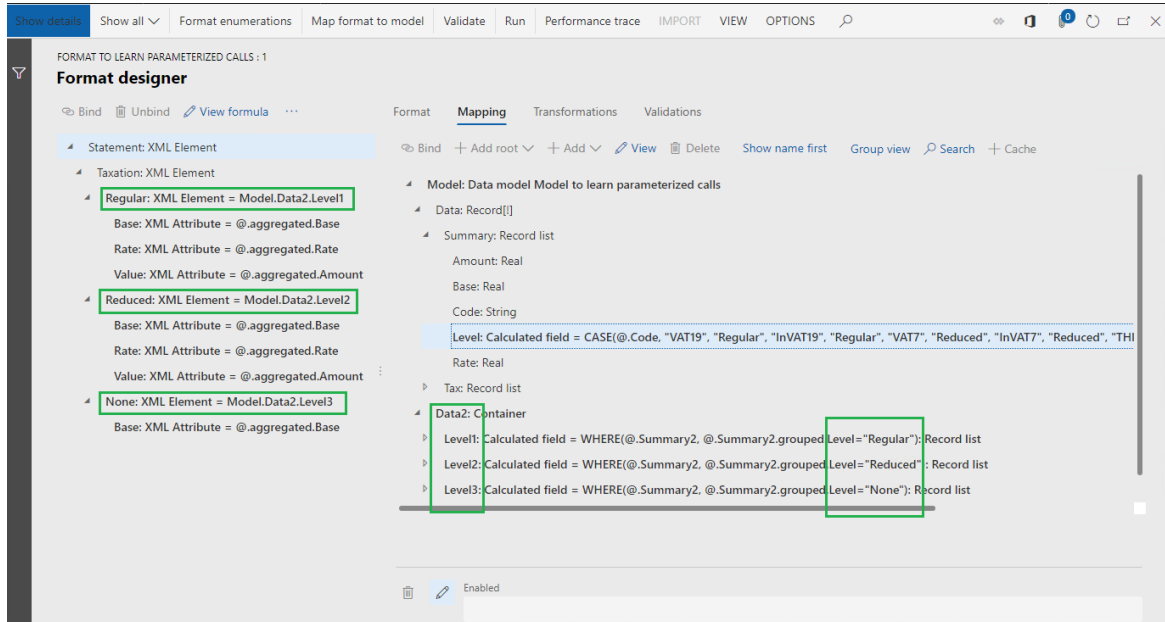
1. In the configuration tree, expand the contents of the **Model to learn parameterized calls** item.
2. Select the **Format to learn parameterized calls** item.
3. Select **Designer**.
4. Select **Expand/Collapse**.

The **Format to learn parameterized calls** ER format is designed to generate a tax statement in XML format that presents several levels of taxation (regular, reduced, and none). Each level has a different number of details.

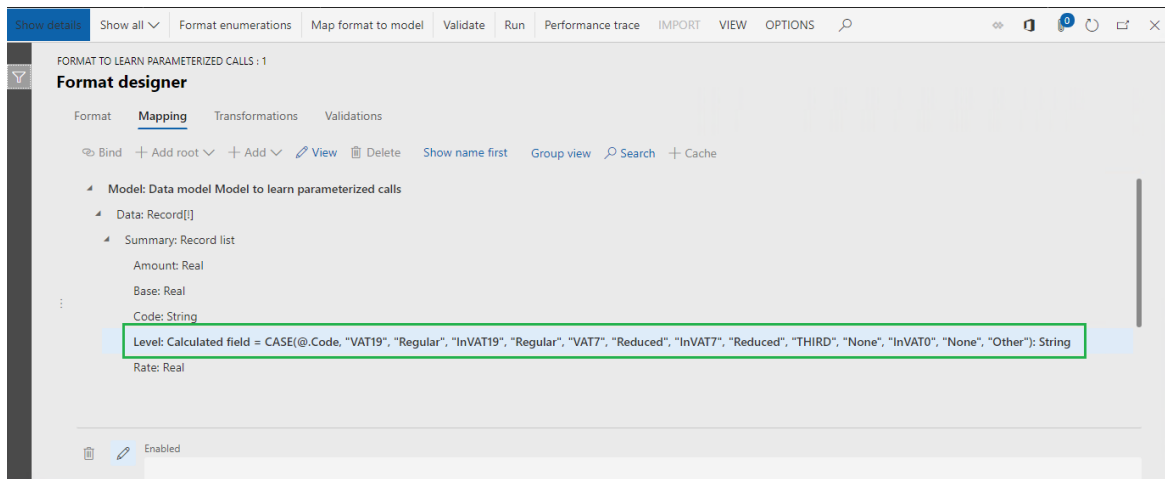


5. On the **Mapping** tab, expand the **Model, Data, and Summary** items.

The **Model.Data.Summary** data source returns the list of tax transactions. These transactions are summarized by tax code. For this data source, the **Model.Data.Summary.Level** calculated field has been configured to return the code for the taxation level of each summarized record. For any tax code that can be retrieved from the **Model.Data.Summary** data source at runtime, the calculated field returns the taxation level code (**Regular**, **Reduced**, **None**, or **Other**) as a text value. The **Model.Data.Summary.Level** calculated field is used to filter records of the **Model.Data.Summary** data source and enter the filtered data in each XML element that represents a taxation level by using the **Model.Data2.Level1**, **Model.Data2.Level2**, and **Model.Data2.Level3** fields.



The **Model.Data.Summary.Level** calculated field has been configured so that it contains an ER expression. Note that tax codes (**VAT19**, **InVAT19**, **VAT7**, **InVAT7**, **THIRD**, and **InVAT0**) are hardcoded into this configuration. Therefore, this ER format is dependent on the legal entity where these tax codes were configured.



To support a different set of tax codes for each legal entity, you must follow these steps:

- Create a derived version of the ER format for each legal entity.
- Update the tax codes in the **Model.Data.Summary.Level** calculated field, based on the legal entity setting.

6. Close the **Format designer** page.

## Create a derived format

Next, you will use the ER application-specific parameters feature to support a different set of tax codes for each

legal entity in a single ER format.

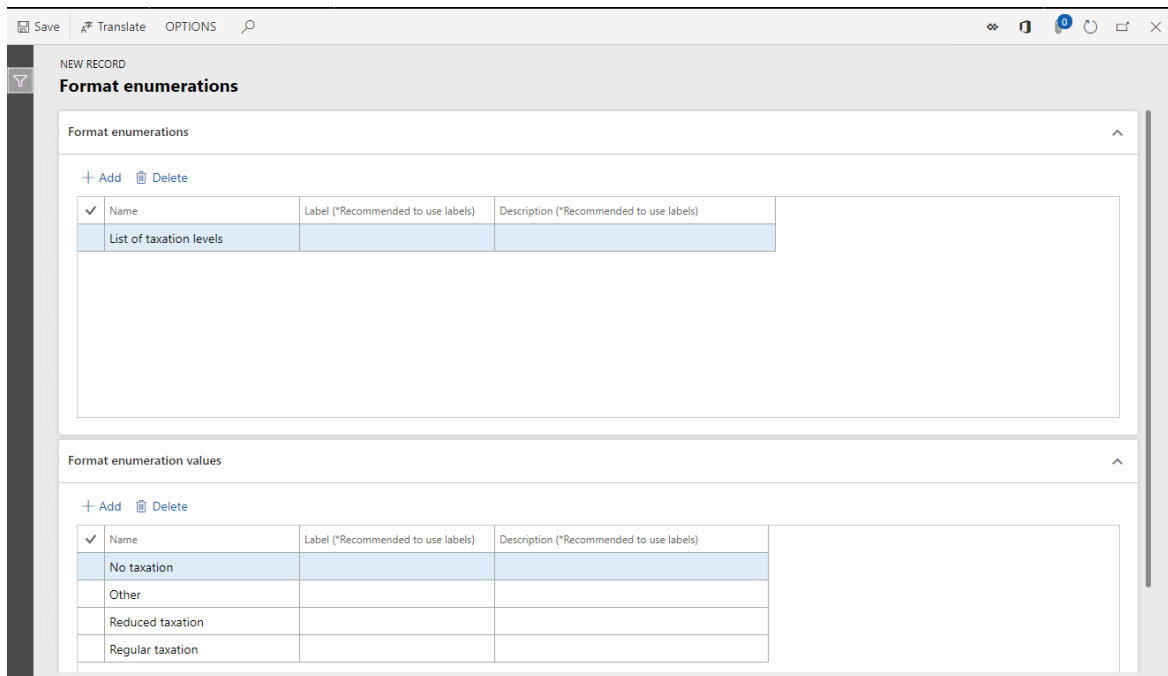
1. In the configuration tree, expand the contents of the **Model to learn parameterized calls** item.
2. Select the **Format to learn parameterized calls** item.
3. Select **Create configuration**.
4. Select the **Derive from Name: Format to learn parameterized calls, Microsoft** option.
5. In the **Name** field, enter **Format to learn how to lookup LE data**.
6. Select **Create configuration**.

## Configure a derived format

### Add a format enumeration

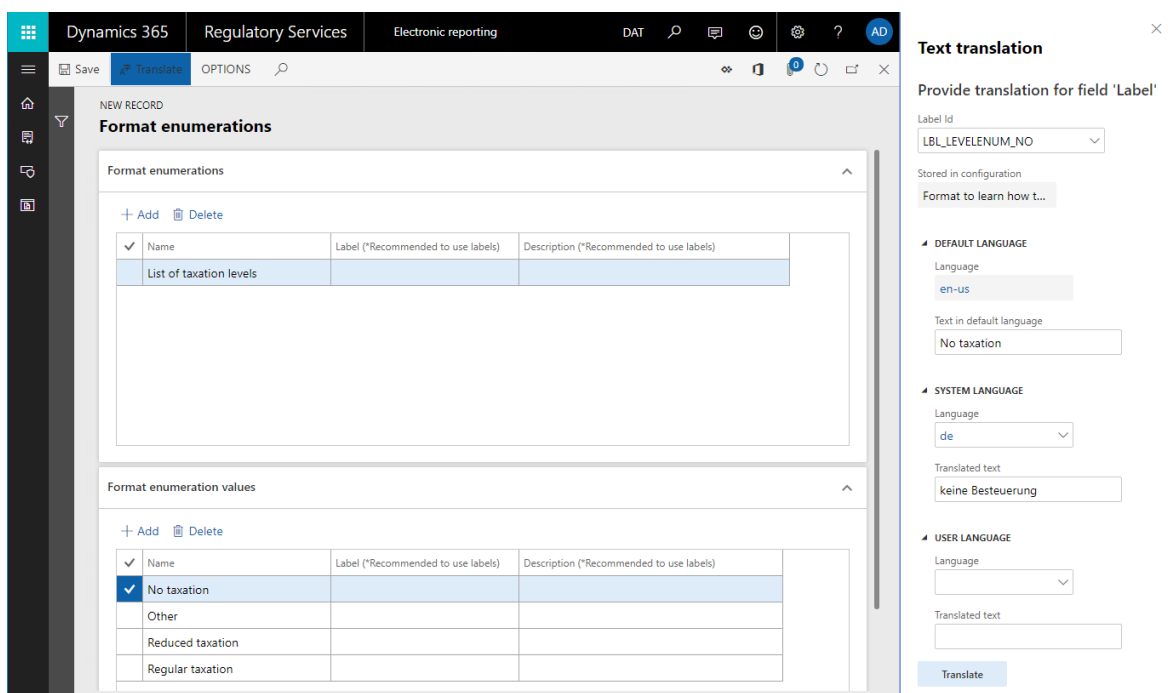
Next, you will add a new ER format enumeration. The values of this format enumeration will be presented to business users, who will specify legal entity–dependent sets of tax codes for the various taxation levels that are used in the ER format.

1. Select **Designer**.
2. Select **Format enumerations**.
3. Select **Add**.
4. In the **Name** field, enter **List of taxation levels**.
5. Select **Save**.
6. On the **Format enumeration values** tab, select **Add**.
7. In the **Name** field, enter **Regular taxation**.
8. Select **Add** again.
9. In the **Name** field, enter **Reduced taxation**.
10. Select **Add** again.
11. In the **Name** field, enter **No taxation**.
12. Select **Add** again.
13. In the **Name** field, enter **Other**.



Because the business users might use different languages to specify legal entity–dependent sets of tax codes, we recommend that you translate the values of this enumeration into the languages that are configured as the preferred languages for those users in Finance.

14. Select the **No taxation** record.
15. Click in the **Label** field.
16. Select **Translate**.
17. In the **Text translation** pane, in the **Label Id** field, enter **LBL\_LEVELENUM\_NO**.
18. In the **Text in default language** field, enter **No taxation**.
19. In the **Language** field, select **DE**.
20. In the **Translated text** field, enter **keine Besteuerung**.
21. Select **Translate**.



22. Select **Save**.

23. Close the **Format enumerations** page.

### Add a new lookup data source

Next, you will add a new data source to specify how business users will specify legal entity–dependent rules to recognize the correct taxation level for each summarized transaction record.

1. On the **Mapping** tab, select **Add**.

2. Select **Format enumeration\Lookup**.

You just identified that each rule that business users specify for taxation level recognition will return a value of an ER format enumeration. Notice that the **Lookup** data source type can be accessed under the **Data model** and **Dynamics 365 for Operations** blocks in addition to the **Format enumeration** block. Therefore, ER data model enumerations and application enumerations can be used to specify the type of values that are returned for data sources of that type.

3. In the **Name** field, enter **Selector**.

4. In the **Format enumeration** field, select **List of taxation levels**.

You just specified that, for each rule that is specified in this data source, a business user must select one of the values of the **List of taxation levels** format enumeration as a returned value.

5. Select **Edit lookup**.

6. Select **Columns**.

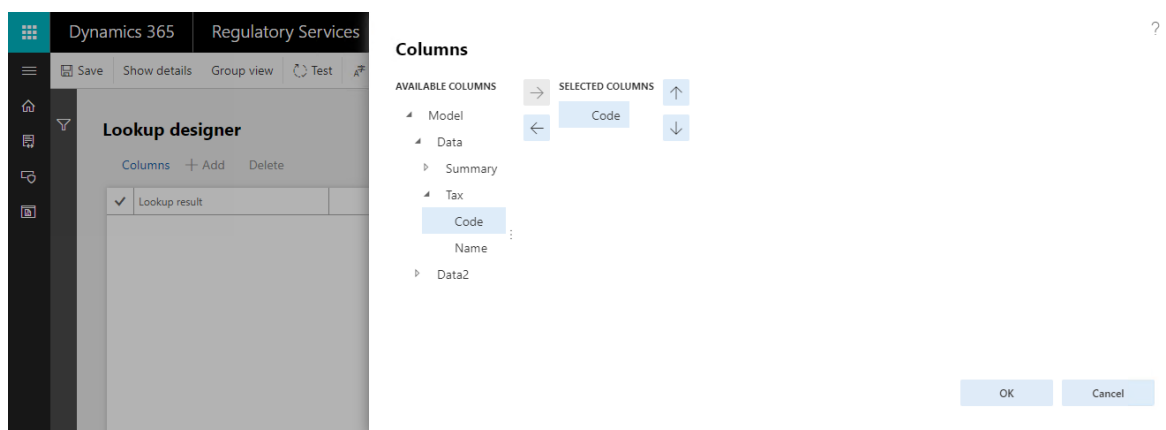
7. Expand the **Model** item.

8. Expand the **Data** item.

9. Expand the **Tax** item.

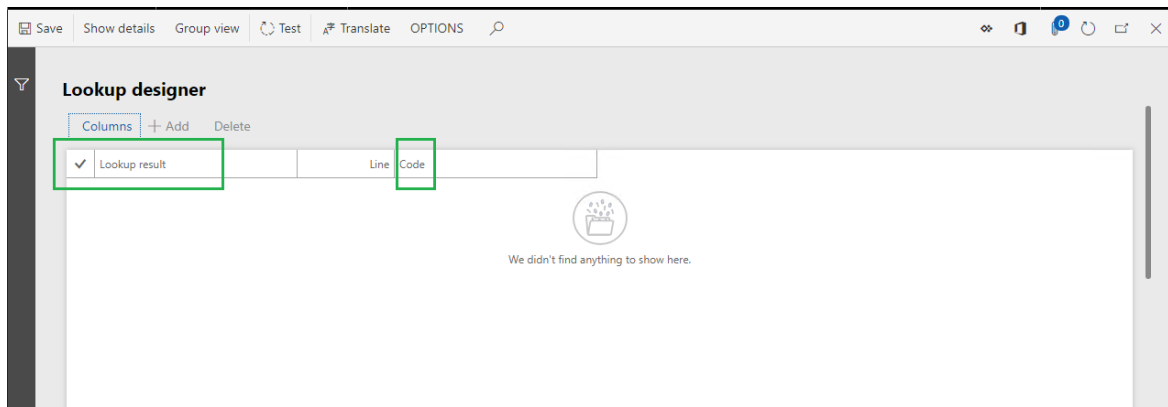
10. Select the **Model.Data.Tax.Code** item.

11. Select the **Add** button (the right arrow).



You just specified that, for each rule that is specified in this data source for taxation level recognition, a business user must select one of the tax codes as a condition. The list of tax codes that the business user can select will be returned by the **Model.Data.Tax** data source. Because this data source contains the **Name** field, the name of the tax code will be shown for each tax code value in the lookup that is presented to the business user.

12. Select **OK**.



Business users can add multiple rules as records of this data source. Each record will be numbered by a line code. Rules will be evaluated in order of increasing line number.

Because you selected the **Tax code** field as a condition for rules in this lookup data source, and because **Tax code** is set up as a field of the **String** data type, each rule will be evaluated at runtime by comparing the tax code that is passed to the data source with the tax code that is defined in this record of the data source.

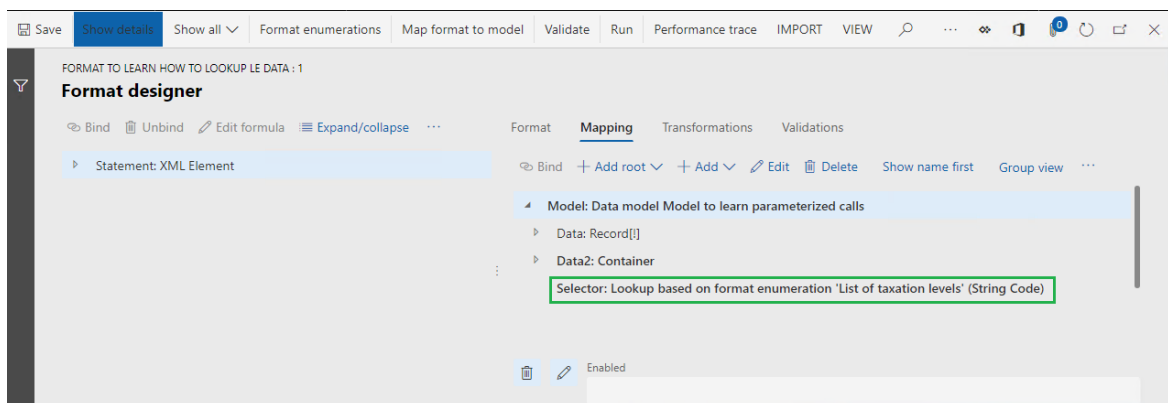
When a rule that satisfies the configured condition is found, this data source returns the lookup value of the rule that is defined in the **Lookup result** field. If no rule is found, an exception is thrown to notify the user that the current data source can't return a correct value.

13. Select **Save**.

14. Close the **Lookup designer** page.

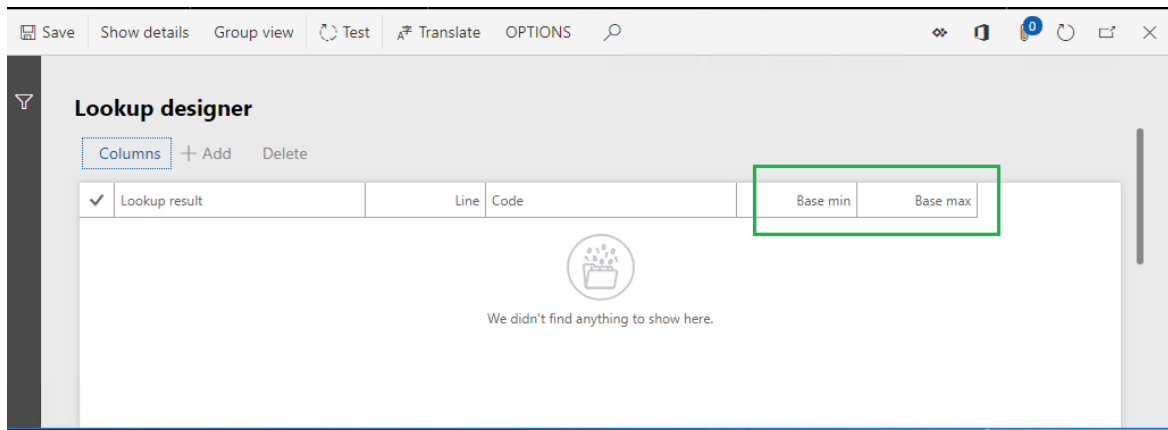
15. Select **OK**.

Notice that you added a new data source that will return the taxation level as the value of the **List of taxation levels** format enumeration for any tax code that is passed to the data source as the argument of the **Code** parameter of the **String** data type.



Note that the evaluation of configured rules depends on the data type of the fields that have been selected to define conditions of those rules. When you select a field that is configured as a field of either the **Numeric** or **Date** data type, the criteria will differ from the criteria that were described earlier for the **String** data type. For **Numeric** and **Date** fields, the rule must be specified as a range of values. The condition of the rule will then be considered satisfied when a value that is passed to the data source is in the configured range.

The following illustration shows an example of this type of setup. In addition to the **Model.Data.Tax.Code** field of the **String** data type, the **Model.Tax.Summary.Base** field of the **Real** data type is used to specify conditions for a lookup data source.



Because the **Model.Data.Tax.Code** and **Model.Tax.Summary.Base** fields are selected for this lookup data source, each rule of this data source will be configured in the following way:

- In the list that is presented, the value of the **List of taxation levels** format enumeration must be selected as a returned value.
- The tax code must be entered as a condition of this rule. Only tax codes that are provided by the **Model.Data.Tax** data source are applicable.
- Minimum and maximum values of the tax base amount must be entered as conditions of this rule.

Here is how each rule of this data source will be evaluated at runtime:

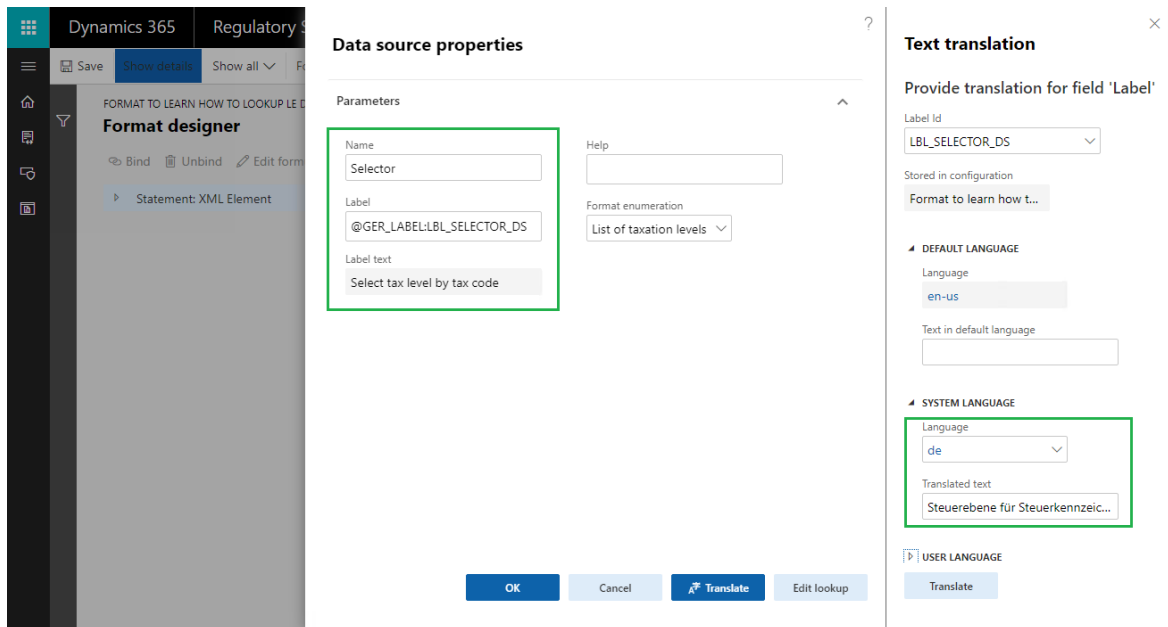
- Does the code of the **String** data type that was passed to this data source equal the tax code of a rule?
- Does the value of the **Real** data type that was passed to this data source fall between specific minimum and maximum values?

A rule will be considered applicable when both conditions are satisfied.

### Translate the label of the lookup data source that was added

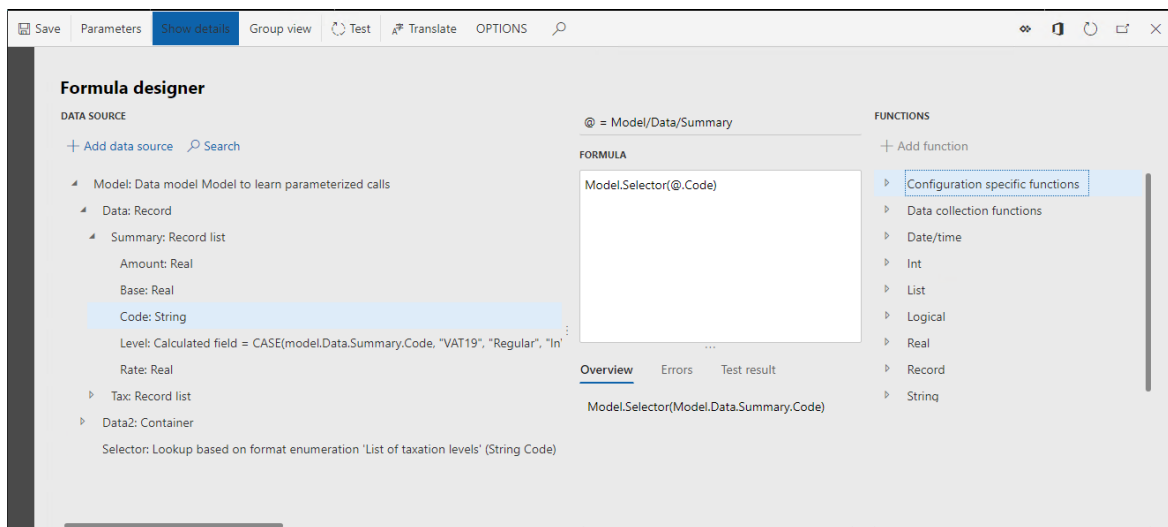
Because business users might use different languages to specify legal entity–dependent sets of tax codes, we recommend that you translate the label of any lookup data source that you add, so that it's presented in each user's preferred language on the corresponding page.

1. Select the **Model.Data.Selector** data source.
2. Select **Edit**.
3. Click in the **Label** field.
4. Select **Translate**.
5. In the **Text translation** pane, in the **Label Id** field, enter **LBL\_SELECTOR\_DS**.
6. In the **Text in default language** field, enter **Select tax level by tax code**.
7. In the **Language** field, select **DE**.
8. In the **Translated text** field, enter **Steuerebene für Steuerkennzeichen auswählen**.
9. Select **Translate**.
10. Select **OK**.



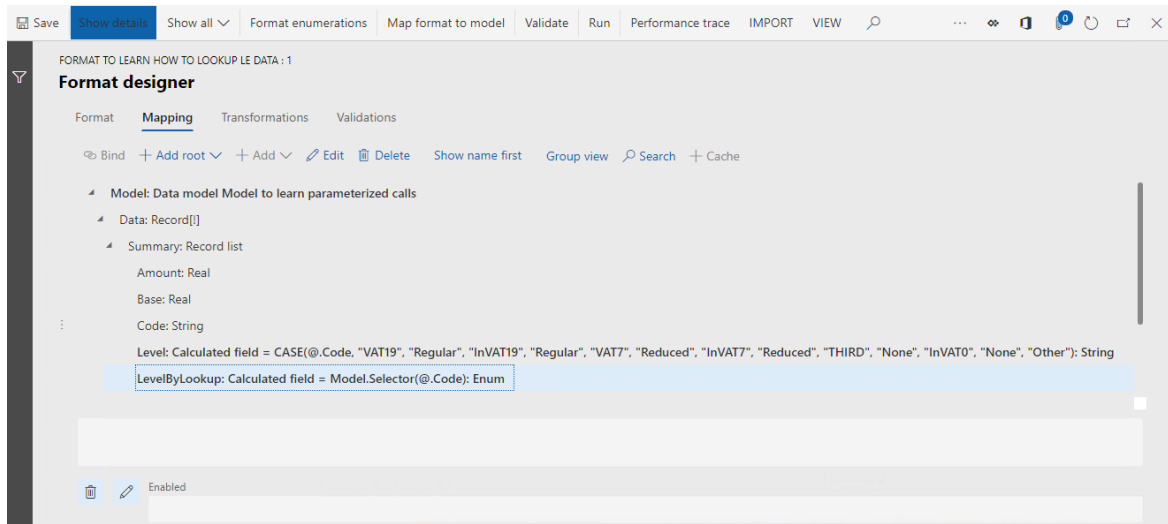
### Add a new field to consume the configured lookup

1. Expand the **Model.Data** item.
2. Select the **Model.Data.Summary** item.
3. Select **Add**.
4. Select **Functions/Calculated field**.
5. In the **Name** field, enter **LevelByLookup**.
6. Select **Edit formula**.
7. In the **Formula** field, enter **Model.Selector(Model.Data.Summary.Code)**.
8. Select **Save**.



9. Close the **Formula editor** page.
10. Select **OK**.





Notice that the **LevelByLookup** calculated field that you added will return the taxation level as the value of the **List of taxation levels** format enumeration for each summarized tax transactions record. The tax code of the record will be passed to the **Model.Selector** lookup data source, and the set of rules for this data source will be used to select the correct taxation level.

### Add a new format enumeration-based data source

Next, you will add a new data source that refers to the format enumeration that you added earlier. Values of this data source will be used in an ER format expression later.

1. Select **Add root**.
2. Select **Format enumerations\Enumeration**.
3. In the **Name** field, enter **TaxationLevel**.
4. In the **Format enumeration** field, select **List of taxation levels**.
5. Select **Save**.

### Modify an existing field to start to use the lookup

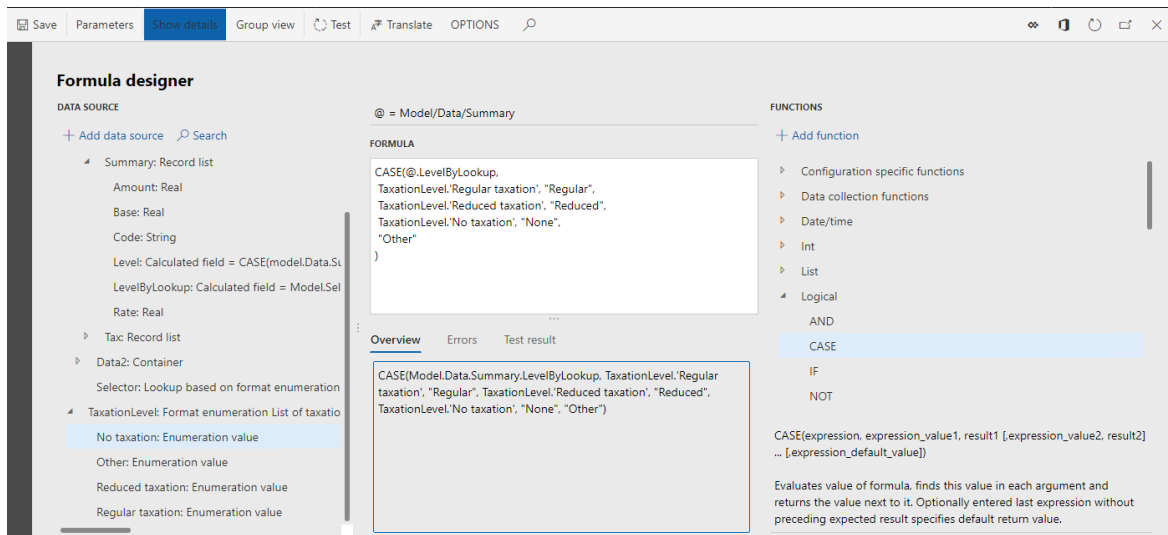
Next, you will modify the existing calculated field so that it uses the configured lookup data source to return the correct taxation level value, depending on the tax code.

1. Select the **Model.Data.Summary.Level** item.
2. Select **Edit**.
3. Select **Edit formula**.

Notice that the current expression of the **Model.Data.Summary.Level** field includes the following hard-coded tax codes:

```
CASE (@.Code, "VAT19", "Regular", "InVAT19", "Regular", "VAT7", "Reduced", "InVAT7", "Reduced", "THIRD", "None", "InVAT0", "None", "Other")
```

4. In the **Formula** field, enter **CASE(@.LevelByLookup, TaxationLevel.'Regular taxation', "Regular", TaxationLevel.'Reduced taxation', "Reduced", TaxationLevel.'No taxation', "None", "Other")**.



Notice that the expression of the **Model.Data.Summary.Level** field will now return the taxation level, based on the tax code of the current record and the set of rules that a business user configures in the **Model.Data.Selector** lookup data source.

5. Select **Save**.
6. Close **Formula designer** page.
7. Select **OK**.
8. Select **Save**.
9. Close **Format designer** page.

## Complete the draft version of a derived format

1. On the **Versions** fast tab, select **Change status**.
2. Select **Complete**.
3. Select **OK**.

## Export completed version of modified format

1. In the configuration tree, select the **Format to learn how to lookup LE data** item.
2. On the **Versions** fast tab, select the record that has a status of **Completed**.
3. Select **Exchange**.
4. Select **Export as XML file**.
5. Select **OK**.
6. The web browser downloads a **Format to learn how to lookup LE data.xml** file. Store this file locally.

Repeat steps in this section for parent items of the **Format to learn how to lookup LE data** format, and store the following files locally:

- **Format to learn parameterized calls.xml**
- **Mapping to learn parameterized calls.xml**
- **Model to learn parameterized calls.xml**

To learn how to use the configured **Format to learn how to lookup LE data** ER format to set up legal entity-dependent sets of tax codes to filter tax transactions by different taxation levels, complete the steps in the [Set up the parameters of an ER format per legal entity](#) topic.

## Additional resources

[Formula designer in Electronic reporting](#)

[Set up the parameters of an ER format per legal entity](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up the parameters of an ER format per legal entity

2/18/2021 • 9 minutes to read • [Edit Online](#)

## Prerequisites

To complete these steps, you must first complete the steps in the [Configure ER formats to use parameters that are specified per legal entity](#) topic.

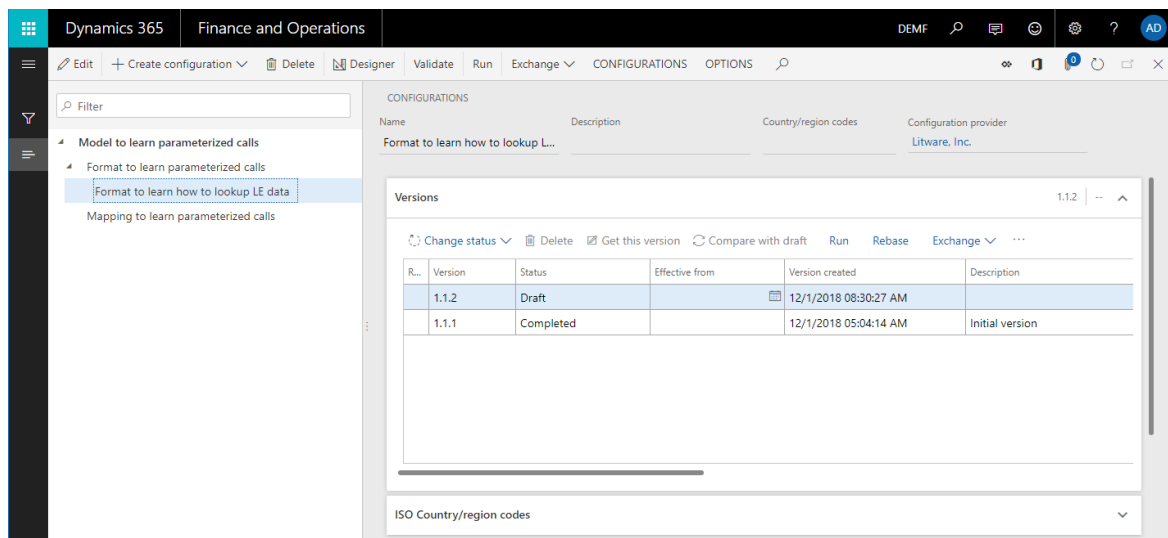
To complete the examples in this topic, you must have access to Microsoft Dynamics 365 Finance (Finance) for one of the following roles:

- Electronic reporting developer
- Electronic reporting functional consultant
- System administrator

## Import ER configurations

1. Sign in to your environment.
2. On the default dashboard, select **Electronic reporting**.
3. Select **Reporting configurations**.
4. Import, into the current instance of Finance, the configurations that you exported from Regulatory Configuration Services (RCS) while you were completing the steps in the [Configure ER formats to use parameters that are specified per legal entity](#) topic. Follow these steps for each Electronic reporting (ER) configuration, in the following order: data model, model mapping, and formats.
  - a. Select **Exchange > Load from XML file**.
  - b. Select **Browse** to select the file for the required ER configuration in XML format.
  - c. Select **OK**.

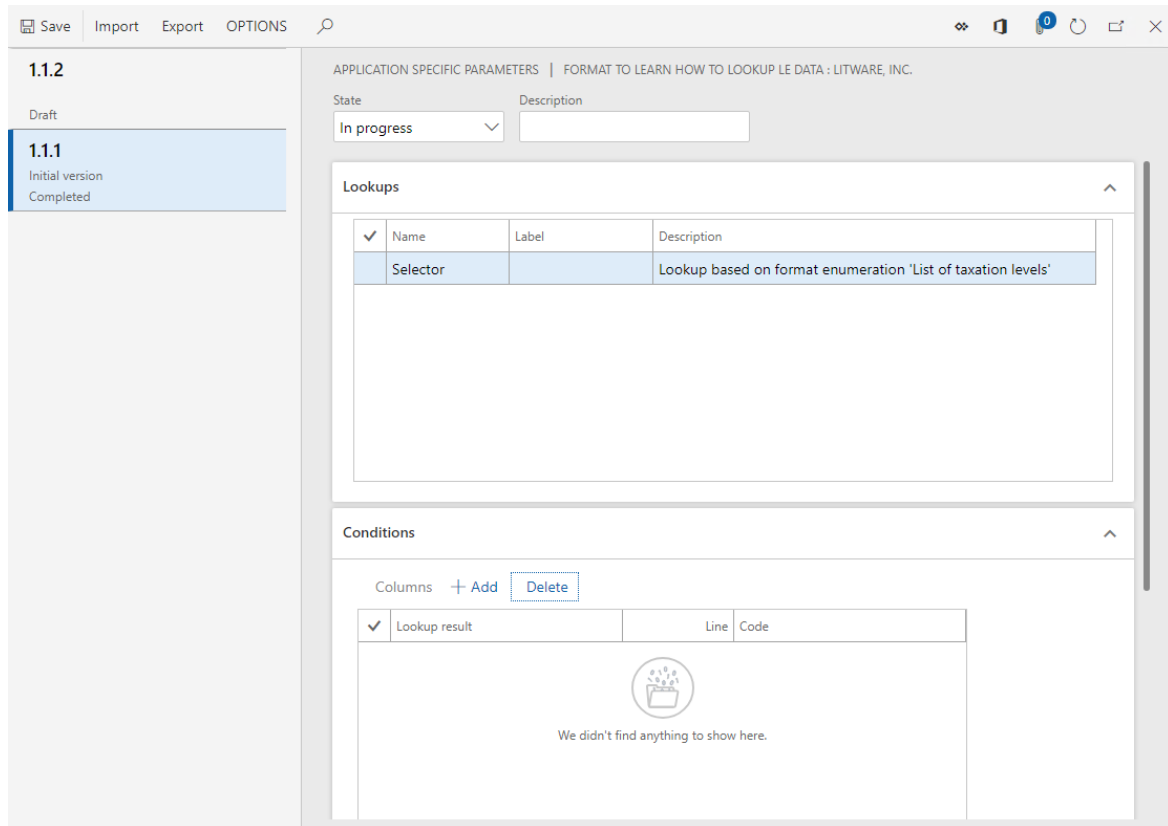
The following illustration shows the configurations that you must have when you've finished.



Set up parameters for the DEMF company

You can use the ER framework to set up application-specific parameters for an ER format.

1. Select the **DEMF** legal entity.
2. In the configurations tree, select the **Format to learn how to lookup LE data** format.
3. On the Action Pane, on the **Configurations** tab, in the **Application specific parameters** group, select **Setup**.



On the **Application specific parameters** page, you can configure the rules for the **Selector** data source of the **Format to learn how to lookup LE data** format.

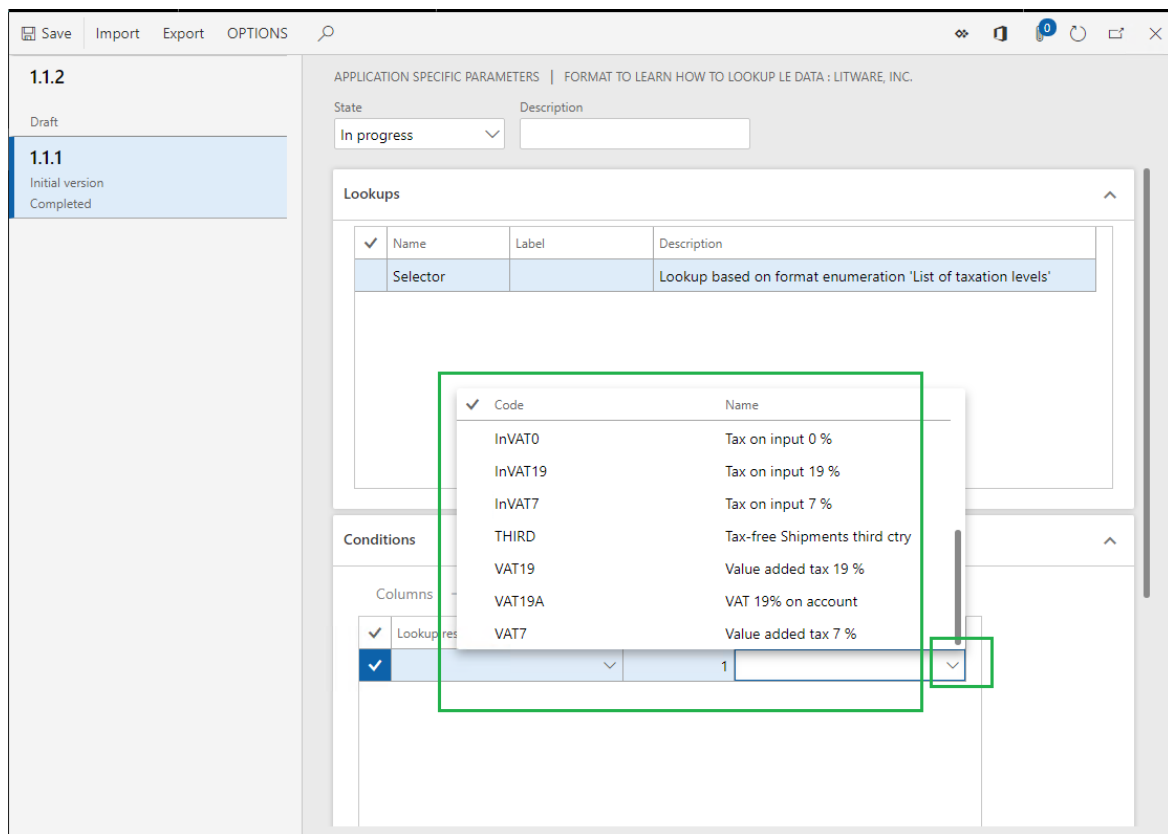
If the base ER format will contain several data sources of the **Lookup** type, you must select the desired data source on the **Lookups** FastTab before you can start to configure the set of rules for the data source.

For each data source, you can configure separate rules for each version of the selected ER format.

The whole set of rules for all lookup data sources that are available in the selected version of the base ER format makes up the application-specific parameters for the ER format.

4. Select version **1.1.1** of the ER format.
5. On the **Conditions** FastTab, select **Add**.
6. In the **Code** field of the new record, select the drop-down arrow to open the lookup.

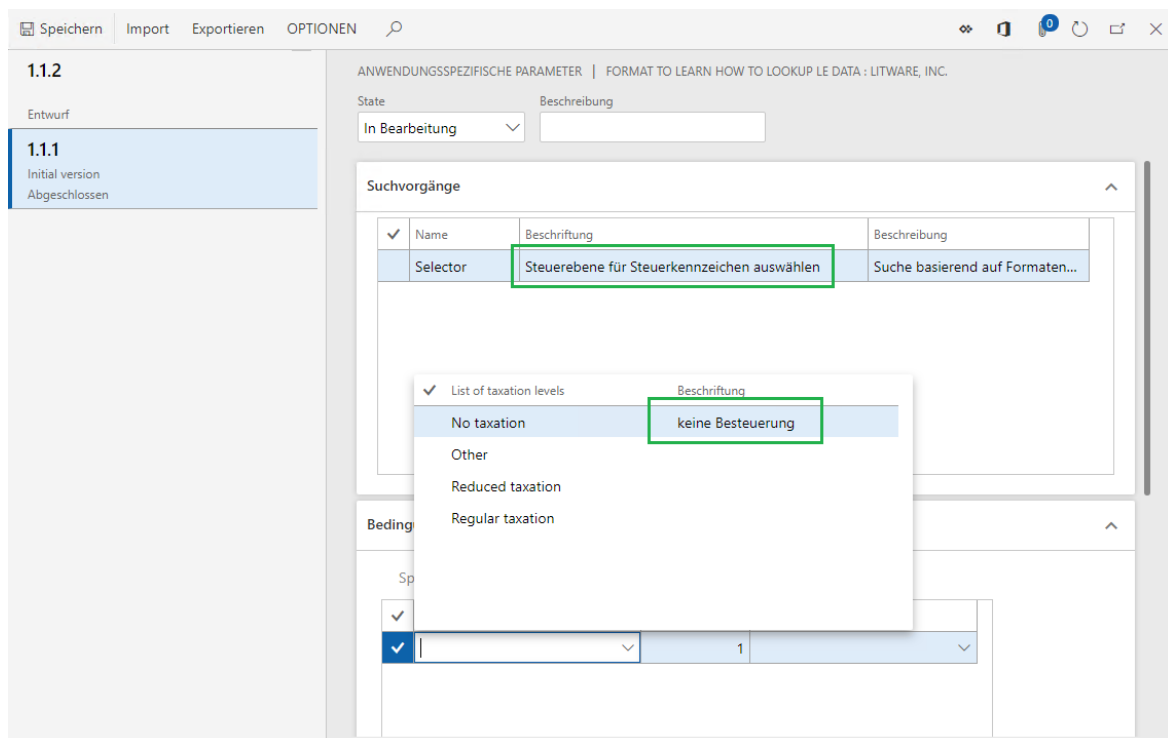
The lookup presents the list of tax codes for selection. This list is returned by the **Model.Data.Tax** data source that has been configured in the base ER format. Because this data source contains the **Name** field, the name of each tax code appears in the lookup.



7. Select the **VAT19** tax code.

8. In the **Lookup result** field of the new record, select the drop-down arrow to open the lookup. The lookup presents the list of values for the TaxationLevel format enumeration for selection.

Note that, if German is selected as the preferred language of the user that you're signed in as, the labels of the values in the lookup will be in German, provided that they have been translated in the base ER format. Additionally, if the label of a lookup data source has been translated, that label will appear in the user's preferred language on the **Lookups** tab.



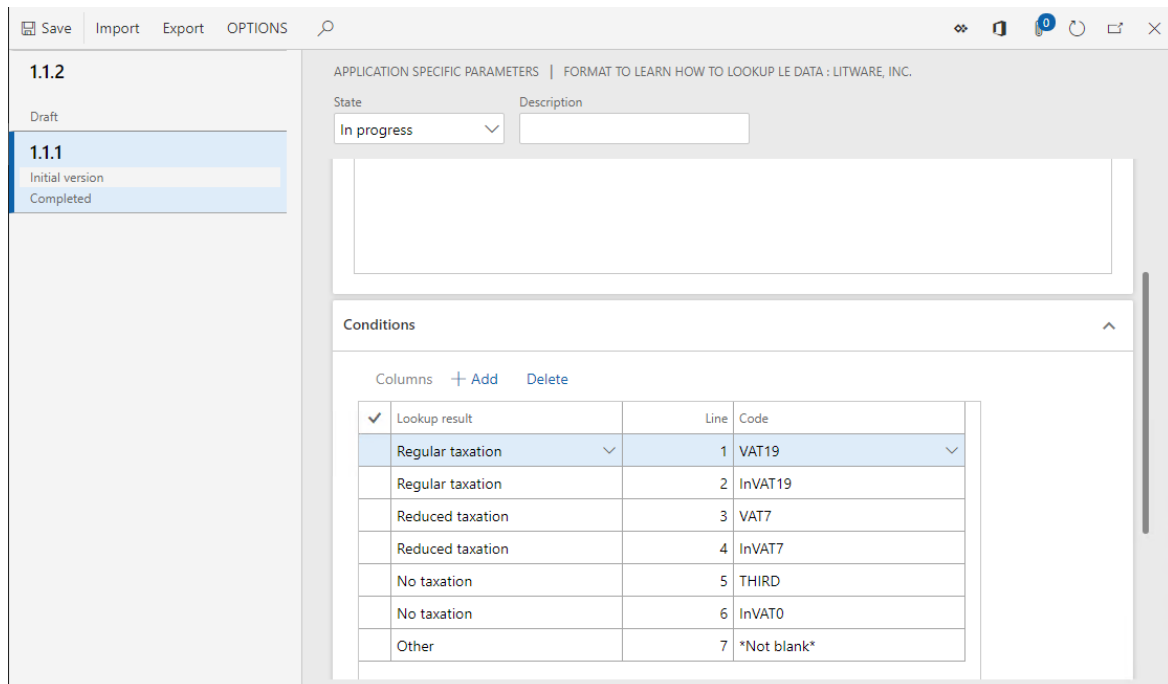
9. Select the **Regular taxation** value.

By adding this record, you define the following rule: Whenever the **Selector** lookup data source is

requested, and the **VAT19** tax code is passed as an argument, **Regular taxation** will be returned as the requested taxation level.

10. Select **Add**, and then follow these steps:
  - a. In the **Code** field, select the **InVAT19** tax code.
  - b. In the **Lookup result** field, select the **Regular taxation** value.
11. Select **Add** again, and then follow these steps:
  - a. In the **Code** field, select the **VAT7** tax code.
  - b. In the **Lookup result** field, select the **Reduced taxation** value.
12. Select **Add** again, and then follow these steps:
  - a. In the **Code** field, select the **InVAT7** tax code.
  - b. In the **Lookup result** field, select the **Reduced taxation** value.
13. Select **Add** again, and then follow these steps:
  - a. In the **Code** field, select the **THIRD** tax code.
  - b. In the **Lookup result** field, select the **No taxation** value.
14. Select **Add** again, and then follow these steps:
  - a. In the **Code** field, select the **InVAT0** tax code.
  - b. In the **Lookup result** field, select the **No taxation** value.
15. Select **Add** again, and then follow these steps:
  - a. In the **Code** field, select the **\*Not blank\*** option.
  - b. In the **Lookup result** field, select the **Other** value.

By adding this last record, you define the following rule: Whenever the tax code that is passed as an argument doesn't satisfy any of the previous rules, the lookup data source will return **Other** as the requested taxation level.



16. In the **State** field, select **Completed**.

When you run an ER format version that has a status of either **Completed** or **Shared**, this set of rules must be in the **Completed** state. Otherwise, execution of the base ER format will be interrupted when the format tries to load data from this set of rules while the **Selector** lookup data source is being run.

When you run an ER format version that has a status of **Draft**, the base ER format can access this set of rules, regardless of its state.

17. Select **Save**.
18. Close the **Application specific parameters** page.

## Run the ER format in the DEMF company

1. In the configurations tree, select the **Format to learn how to lookup LE data** format.
2. On the Action Pane, select **Run**.
3. In the dialog box that appears, select **OK**.
4. Download the statement that is generated and store it locally.

In the generated statement, notice that the summary of the **InVAT7** tax code has been put on the **Reduced** level, and the summaries of the **VAT19** and **InVA19** tax codes have been put on the **Regular** level. This behavior is determined by the configuration in the legal entity–dependent set of rules.

5. Go to **Tax > Indirect taxes > Sales tax > Sales tax codes**.
6. Select the **InVAT7** tax code.
7. On the Action Pane, on the **Sales tax code** tab, in the **Inquiries** group, select **Posted sales tax** to view information about the tax value and applied tax rate per tax code.

View accounting OPTIONS

IN VAT7 : TAX ON INPUT 7 %

**Posted sales tax**

Overview General Amount History

✓ Voucher ↑	Date	Source	Sales tax code	Sales tax direction	Transaction
PIV-110000004	1/4/2015	Purchase order	InVAT7	Sales tax receivable	EUR
PIV-110000018	3/1/2016	Purchase order	InVAT7	Sales tax receivable	EUR
PIV-110000027	12/3/2016	Purchase order	InVAT7	Sales tax receivable	EUR

```
<?xml version="1.0" encoding="UTF-8"?>
- <Statement>
- <Taxation>
  <Regular Value="-5073163.93" Rate="19" Base="-26700862.86"/>
  <Reduced Value="3623.2" Rate="7" Base="51760.0"/>
  <None Base="0"/>
</Taxation>
</Statement>
```

Total actual sales tax amount

3,623.20

8. Close the **Posted sales tax** page.

## Set up parameters for the USMF company

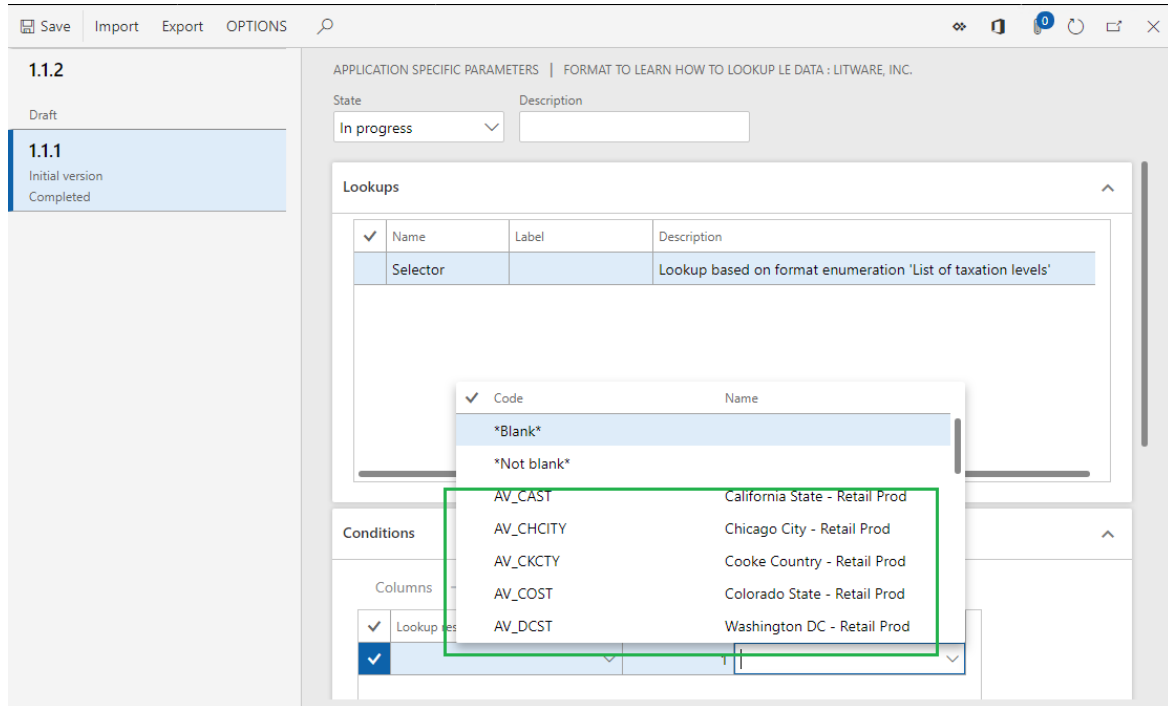
1. Select the **USMF** legal entity.
2. Go to **Organization administration > Electronic reporting > Configurations**.
3. In the configurations tree, expand the **Model to learn parameterized calls** item, expand the **Format to learn parameterized calls** item, and select the **Format to learn how to lookup LE data** format.
4. On the Action Pane, on the **Configurations** tab, in the **Application specific parameters** group, select



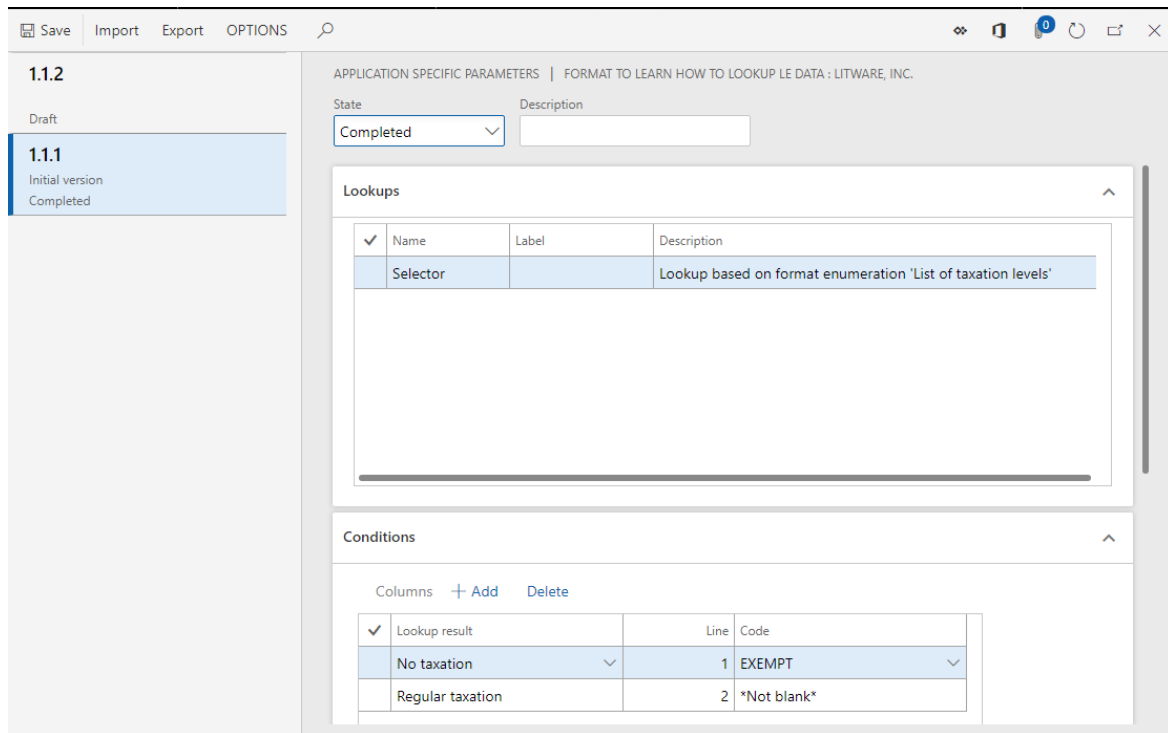
## Setup.

5. Select version **1.1.1** of the selected ER format.
6. On the **Conditions** FastTab, select **Add**.
7. In the **Code** field of the new record, select the drop-down arrow to open the lookup.

The lookup now presents the list of tax codes for the **USMF** company tax for selection.



8. Select the **EXEMPT** tax code.
9. In the **Lookup result** field of the new record, select the **No taxation** value.
10. Select **Add** again.
11. In the **Code** field of the new record, select the **\*Not blank\*** option.
12. In the **Lookup result** field of the new record, select the **Regular taxation** value.
13. In the **State** field, select **Completed**.
14. Select **Save**.



15. Close the **Application specific parameters** page.

## Run the ER format in the USMF company

1. In the configurations tree, select the **Format to learn how to lookup LE data** format.
2. On the Action Pane, select **Run**.
3. In the dialog box that appears, select **OK**.
4. Download the statement that is generated and store it locally.

In the generated statement, notice that you've now reused the same ER format for a different legal entity, but without making any adjustments to the ER format.

## Reuse legal entity–dependent parameters

### Export parameters

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Select **Reporting configurations**.
3. In the configurations tree, select the **Format to learn how to lookup LE data** format.
4. On the Action Pane, on the **Configurations** tab, in the **Application specific parameters** group, select **Setup**.
5. Select version **1.1.1** of the ER format.
6. On the Action Pane, select **Export**.
7. Download the file that is generated and store it locally.

The configured set of application-specific parameters has now been exported as an XML file.

### Import parameters

1. Select version **1.1.2** of the ER format.
2. On the Action Pane, select **Import**.

3. Select **Yes** to confirm that you want to override the existing application-specific parameters for this format version.
4. Select **Browse** to find the file that contains the exported application-specific parameters for version 1.1.1.
5. Select **OK**.

Version 1.1.2 of the ER format now has the same application-specific parameters that you originally configured for version 1.1.1.

Note that the application-specific parameters of an ER format are legal entity–dependent. To reuse the application-specific parameters that were configured for one legal entity in a different legal entity, you must export them while you're signed in to the first legal entity and then import them after you sign in to the other legal entity.

You can also use this approach to transfer an ER format related application-specific parameters that were originally configured in one instance of Finance to another instance of Finance.

Be aware that if you configure application-specific parameters for one version of an ER format and import a higher version of the same format into the current Finance instance, the existing application-specific parameters won't be applied for the imported version.

Also be aware that, when you select a file for import, the structure of the application-specific parameters in that file is compared with the structure of the corresponding data source of the **Lookup** type in the ER format that is selected for import. The import is done when the structure of each application-specific parameter matches the structure of the corresponding data source in the ER format that is selected for import. If the structures don't match, you receive a warning message that states that the import can't be done. If you force the import to be done, the existing application-specific parameters for the selected ER format will be cleaned up, and you must set them up from the beginning.

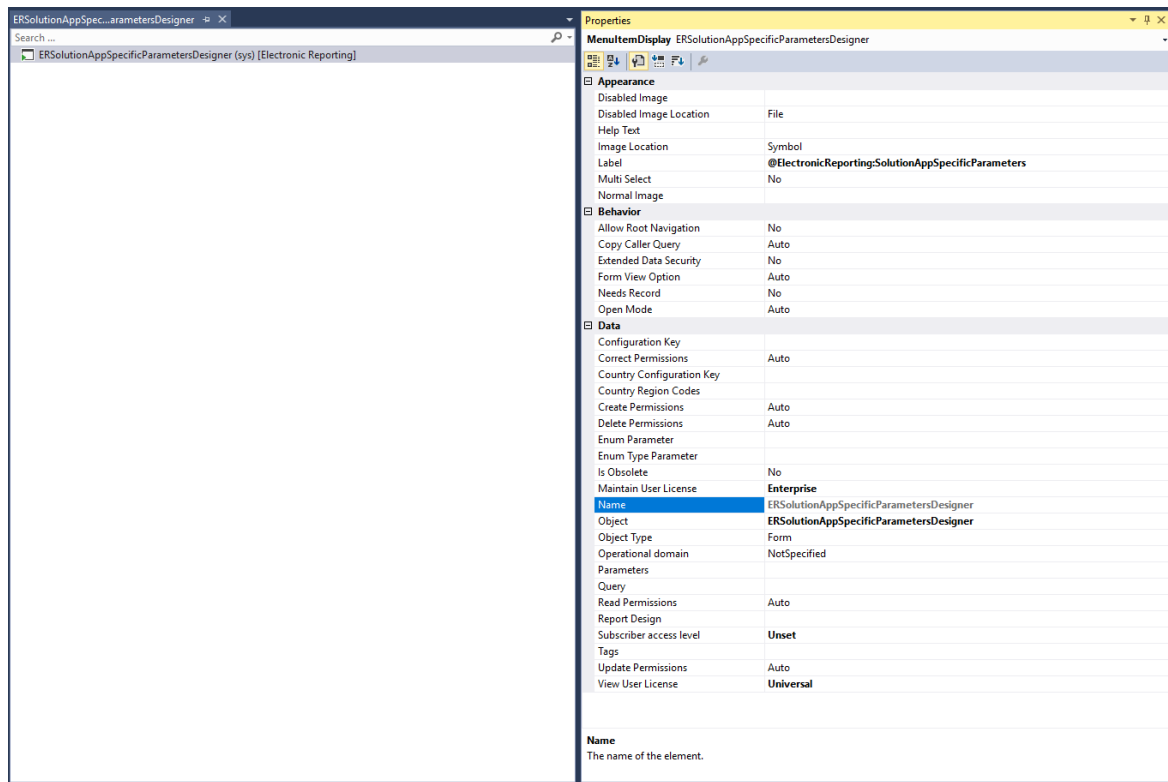
## Relationship between application-specific parameters and an ER format

The relationship between an ER format and its application-specific parameters is established by the ER format's instance-independent unique identification code. Therefore, when you remove an ER format from Finance, the application-specific parameters that are configured for the ER format are kept in the current instance of Finance. They can be accessed whenever the base ER format is reimported into this instance of Finance.

## Access application-specific parameters by using the ER framework

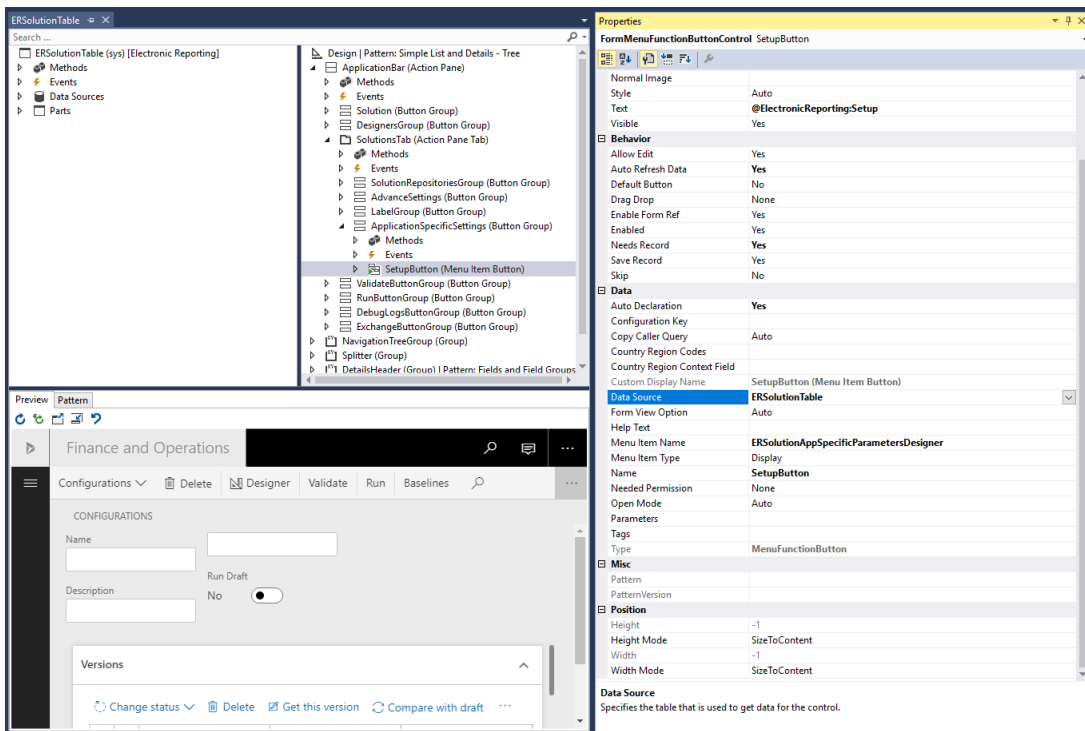
In the preceding example, you have accessed application-specific parameters of an ER format by using the ER framework. This approach doesn't let you restrict access to the application-specific parameters of a specific ER format. If you must apply such restrictions, follow these steps.

1. Either reuse an existing **ERSolutionAppSpecificParametersDesigner** menu item, or implement your own **ERSolutionAppSpecificParametersDesigner** menu item.



2. Follow one of these steps:

- a. Create a new menu item button, and link it to the corresponding record from the ERSolutionTable table by setting its Data Source property to ERSolutionTable.



- b. Create a simple button, and override the Clicked method as shown in the following example.

By using this approach, you can specify a unique solution ID (defined via the GUID value) to allow access to the application-specific parameters of only a specific ER format and descendant copies that have been derived from it.

```
public void clicked()
{
    super();

    ERSolutionTable solutionTableRecord = ERSolutionTable::findByGUID(str2Guid('ADACCB2F-
EFD1-4C90-877D-7E1E5D1AEE92'));

    Args args = new Args();
    args.record(solutionTableRecord);
    args.caller(this);

    new MenuFunction(menuItemDisplayStr(ERSolutionAppSpecificParametersDesigner),
MenuItemType::Display)
        .run(args);
}
```

## Additional resources

[Formula designer in Electronic reporting](#)

[Configure ER formats to use parameters that are specified per legal entity](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Defer the execution of sequence elements in ER formats

2/18/2021 • 11 minutes to read • [Edit Online](#)

## Overview

You can use the Operations designer of the [Electronic reporting \(ER\)](#) framework to [configure](#) the [format component](#) of an ER solution that is used to generate outbound documents in a text format. The hierarchical structure of the configured format component consists of format elements of various types. These format elements are used to fill generated documents with the required information at runtime. By default, when you run an ER format, the format elements are run in the same order as they are presented in the format hierarchy: one by one, from top to bottom. However, at design time, you can change the execution order for any sequence elements of the configured format component.

By turning on the **Deferred execution** option for a sequence format element in the configured format, you can defer (postpone) the execution of that element. In this case, the element isn't run until all other elements of its parent have been run.

To learn more about this feature, complete the example in this topic.

## Limitations

The **Deferred execution** option is supported only for sequence elements that are configured for an ER format that is used to generate **outbound** documents in text format.

The **Deferred execution** option isn't applicable to sequences that have been configured as trimmed sequences where the maximum length is limited.

## Example: Defer the execution of a sequence element in an ER format

The following steps explain how a user in the System administrator or Electronic reporting functional consultant [role](#) can configure an ER format that contains a sequence element where order of execution differs from the order in the format hierarchy.

These steps can be performed in the **USMF** company in Microsoft Dynamics 365 Finance.

### Prerequisites

To complete this example, you must have access to the **USMF** company in Finance for one of the following roles:

- Electronic reporting functional consultant
- System administrator

If you haven't yet completed the example in the [Defer the execution of XML elements in ER formats](#) topic, download the following [configurations](#) of the sample ER solution.

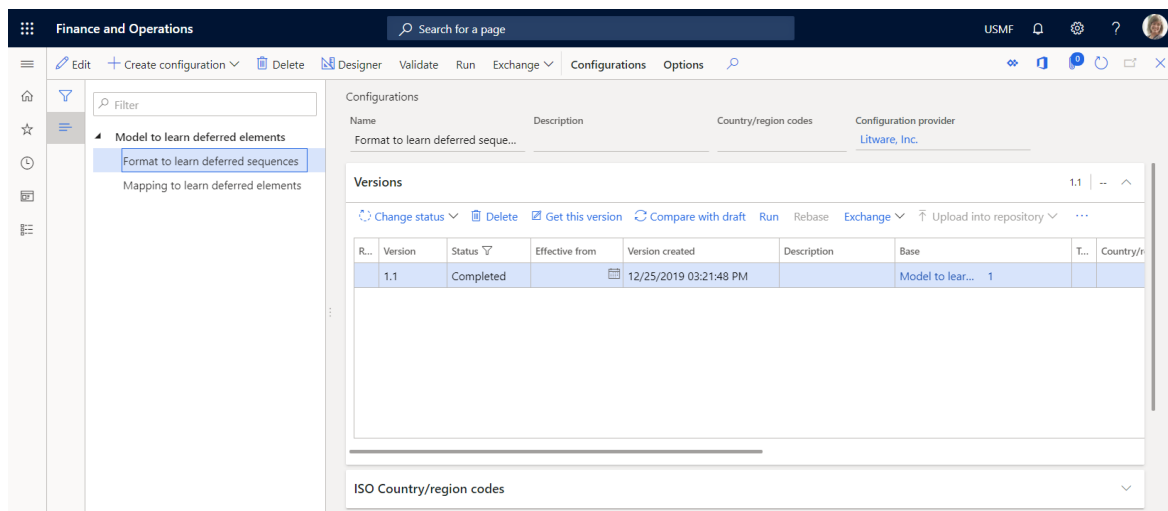
CONTENT DESCRIPTION	FILE NAME
ER data model configuration	<a href="#">Model to learn deferred elements.version.1.xml</a>
ER model mapping configuration	<a href="#">Mapping to learn deferred elements.version.1.1.xml</a>

Before you begin, you must also download and save the following configuration of the sample ER solution.

CONTENT DESCRIPTION	FILE NAME
ER format configuration	<a href="#">Format to learn deferred sequences.version.1.1.xml</a>

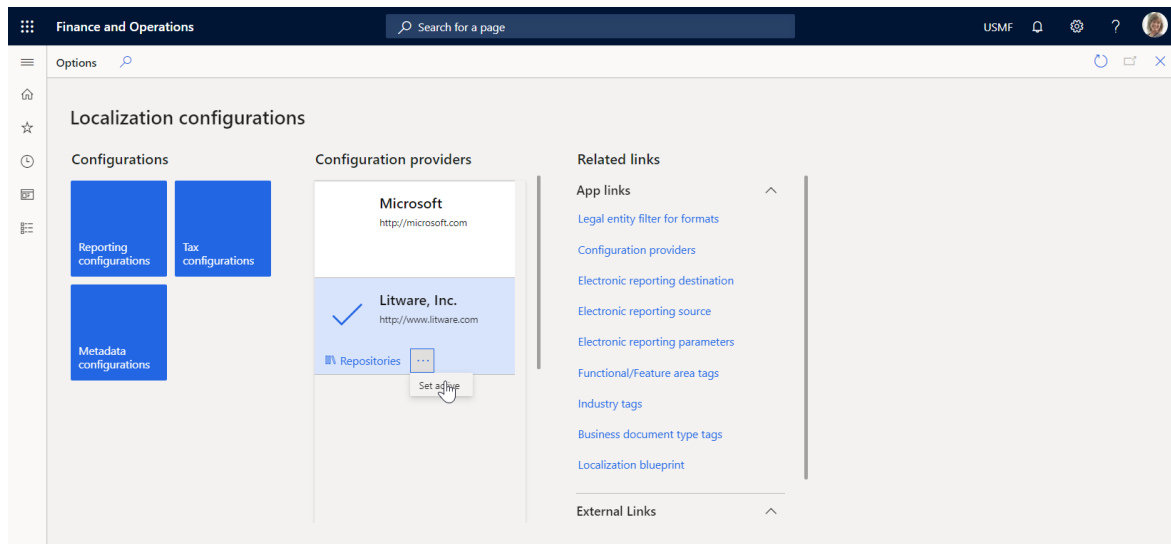
### Import the sample ER configurations

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Select **Reporting configurations**.
3. On the **Configurations** page, if the **Model to learn deferred elements** configuration isn't available in the configuration tree, import the ER data model configuration:
  - a. Select **Exchange**, and then select **Load from XML file**.
  - b. Select **Browse**, find and select the **Model to learn deferred elements.1.xml** file, and then select **OK**.
4. If the **Mapping to learn deferred elements** configuration isn't available in the configuration tree, import the ER model mapping configuration:
  - a. Select **Exchange**, and then select **Load from XML file**.
  - b. Select **Browse**, find and select the **Mapping to learn deferred elements.1.1.xml** file, and then select **OK**.
5. Import the ER format configuration:
  - a. Select **Exchange**, and then select **Load from XML file**.
  - b. Select **Browse**, find and select the **Format to learn deferred sequences.1.1.xml** file, and then select **OK**.
6. In the configuration tree, expand **Model to learn deferred elements**.
7. Review the list of imported ER configurations in the configuration tree.



### Activate a configurations provider

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. On the **Localization configurations** page, in the **Configuration providers** section, make sure that the **configuration provider** for the Litware, Inc. (<http://www.litware.com>) sample company is listed, and that it's marked as active. If this configuration provider isn't listed, or if it isn't marked as active, follow the steps in the [Create a configuration provider and mark it as active](#) topic.

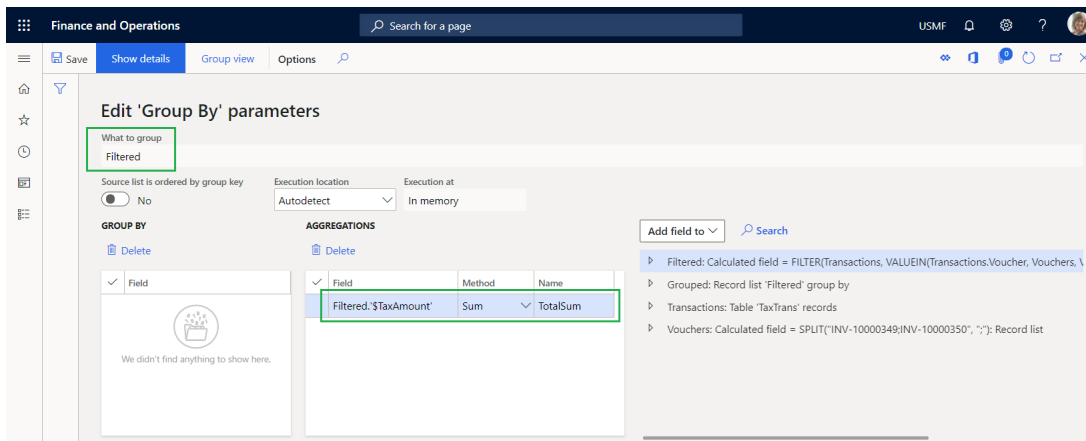


## Review the imported model mapping

Review the settings of the ER model mapping component that is configured to access tax transactions and expose accessed data on request.

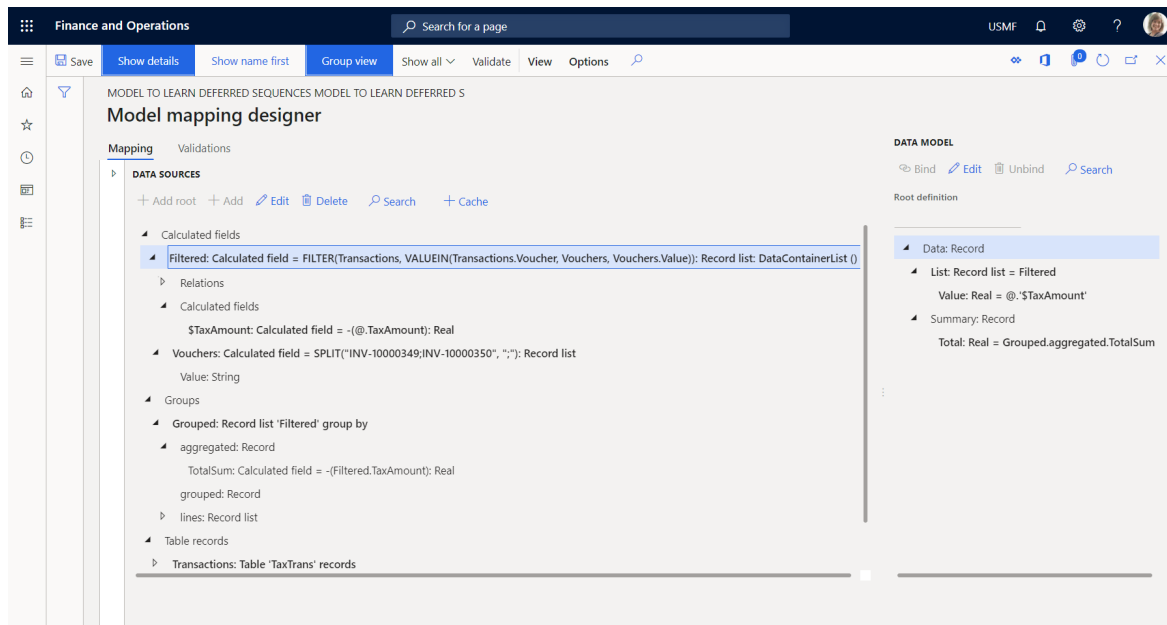
1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Select **Reporting configurations**.
3. On the **Configurations** page, in the configuration tree, expand **Model to learn deferred elements**.
4. Select the **Mapping to learn deferred elements** configuration.
5. Select **Designer** to open the list of mappings.
6. Select **Designer** to review the mapping details.
7. Select **Show details**.
8. Review the data sources that are configured to access tax transactions:
  - The **Transactions** data source of the *Table record* type is configured to access records of the **TaxTrans** application table.
  - The **Vouchers** data source of the *Calculated field* type is configured to return the required voucher codes (**INV-10000349** and **INV-10000350**) as a list of records.
  - The **Filtered** data source of the *Calculated field* type is configured to select, from the **Transactions** data source, only tax transactions of the required vouchers.
  - The **\$TaxAmount** field of the *Calculated field* type is added for the **Filtered** data source to expose the tax value that has the opposite sign.
  - The **Grouped** data source of the *Group By* type is configured to group filtered tax transactions of the **Filtered** data source.
  - The **TotalSum** aggregation field of the **Grouped** data source is configured to summarize values of the **\$TaxAmount** field of the **Filtered** data source for all filtered tax transactions of that data source.





9. Review how the configured data sources are bound to the data model, and how they expose accessed data to make it available in an ER format:

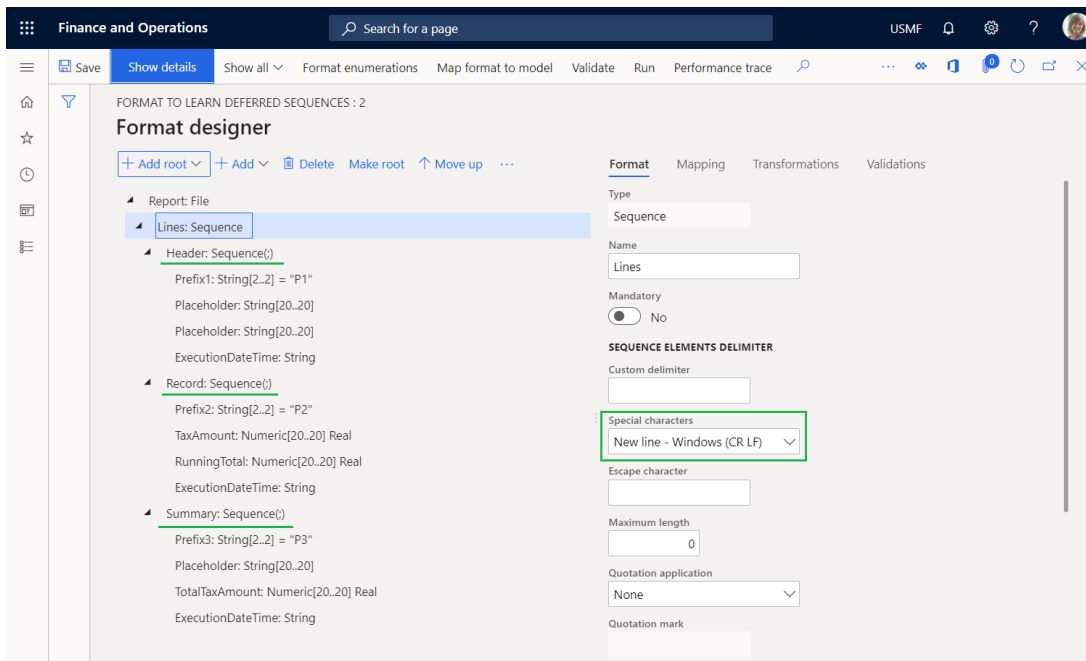
- The **Filtered** data source is bound to the **Data.List** field of the data model.
- The **\$TaxAmount** field of the **Filtered** data source is bound to the **Data.List.Value** field of the data model.
- The **TotalSum** field of the **Grouped** data source is bound to the **Data.Summary.Total** field of the data model.



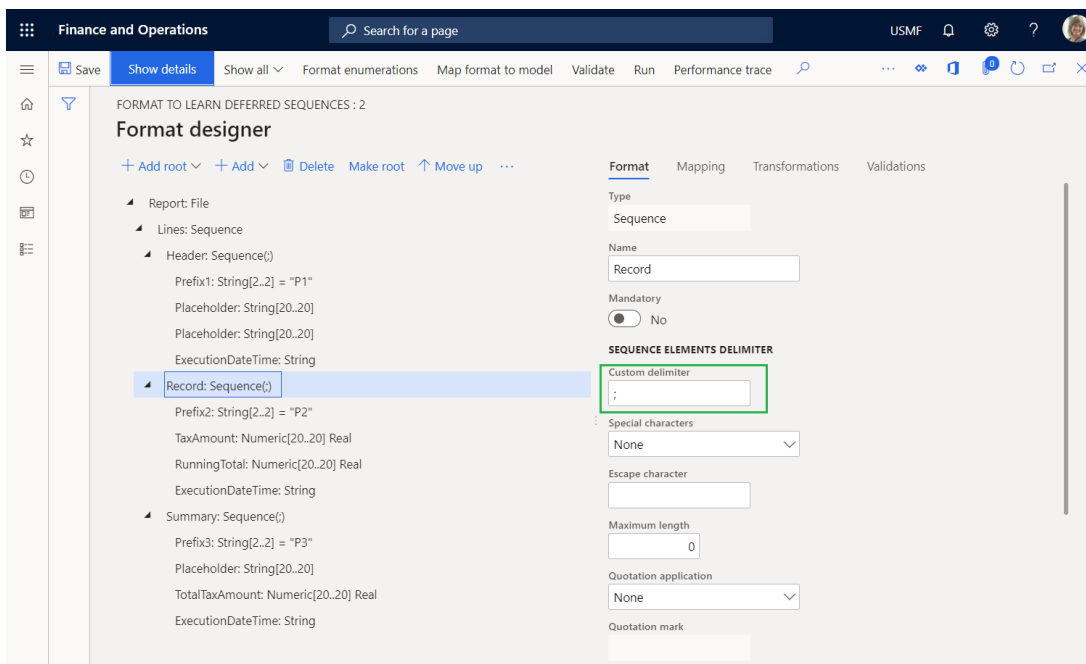
10. Close the **Model mapping designer** and **Model mappings** pages.

### Review the imported format

1. On the **Configurations** page, in the configuration tree, select the **Format to learn deferred sequences** configuration.
2. Select **Designer** to review the format details.
3. Select **Show details**.
4. Review the settings of the ER format components that are configured to generate an outbound document in text format that includes details of the tax transactions:
  - The **Report\Lines** sequence format element is configured to fill the outbound document with a single line that is generated from the nested sequence elements (**Header**, **Record**, and **Summary**).



- The **Report\Lines\Header** sequence format element is configured to fill the outbound document with a single header line that shows the date and time when the processing starts.
- The **Report \Lines\Record** sequence format element is configured to fill the outbound document with a single line that shows the details of individual tax transactions. These tax transactions are separated by a semicolon.

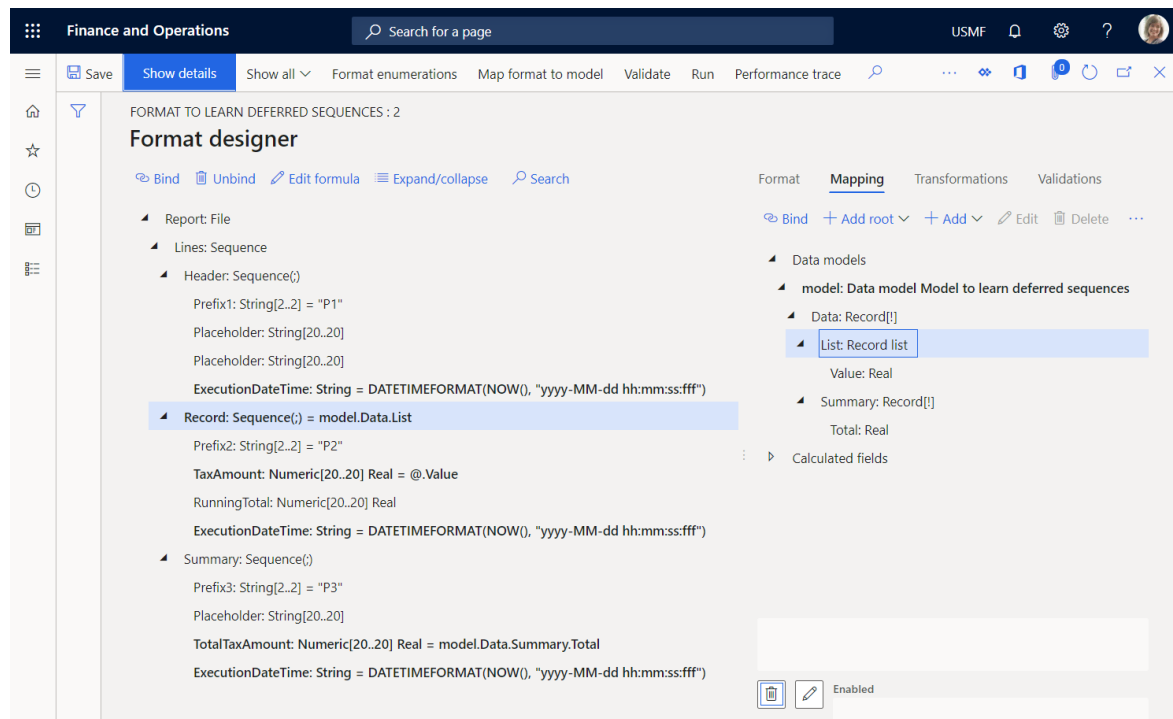


- The **Report\Lines\Summary** sequence format element is configured to fill the outbound document with a single summary line that includes the sum of the tax values from the processed tax transactions.

5. On the **Mapping** tab, review the following details:

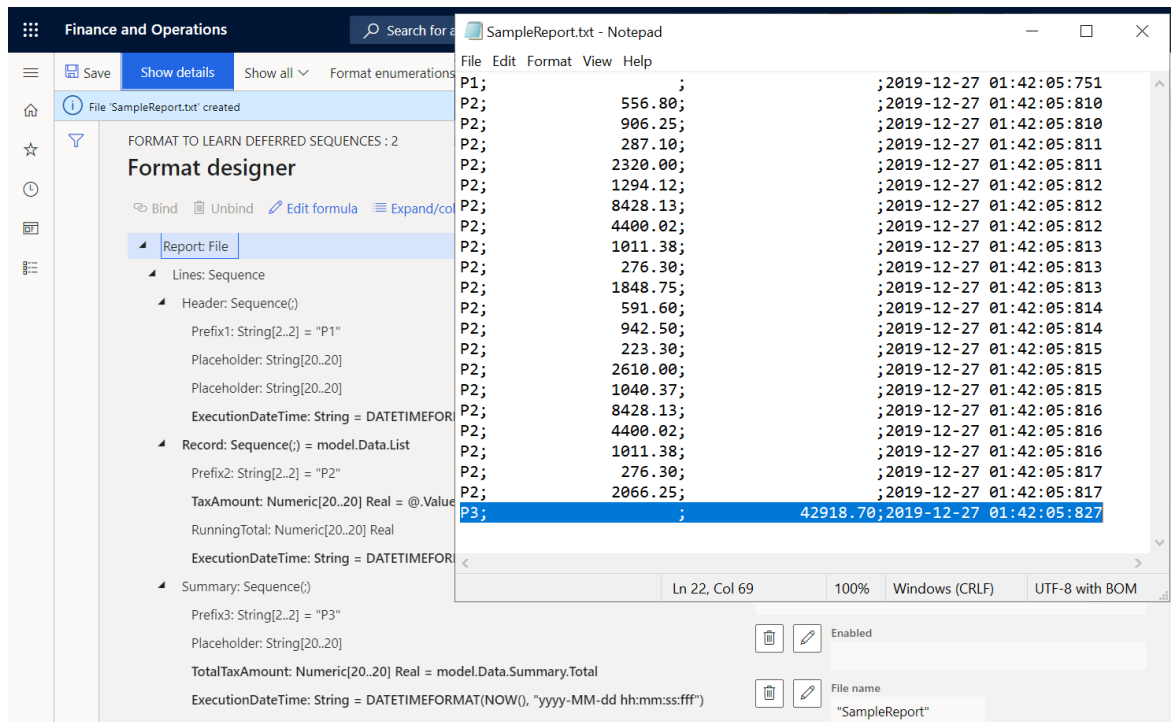
- The **Report\Lines\Header** element doesn't have to be bound to a data source to generate a single line in an outbound document.
- The **Prefix1** element generates **P1** symbols to indicate that the line that is added is the report header line.
- The **ExecutionDateTime** element generates the date and time (including milliseconds) when the header line is added.

- The **Report\Lines\Record** element is bound to the **model.Data.List** list to generate a single line for every record from the bound list.
- The **Prefix2** element generates **P2** symbols to indicate that the line that is added is for the tax transaction details.
- The **TaxAmount** element is bound to **model.Data.List.Value** (which is shown as **@.Value** in the relative path view) to generate the tax value of the current tax transaction.
- The **RunningTotal** element is a placeholder for the running total of the tax values. Currently, this element has no output, because neither a binding nor a default value is configured for it.
- The **ExecutionDateTime** element generates the date and time (including milliseconds) when the current transaction is processed in this report.
- The **Report\Lines\Summary** element doesn't have to be bound to a data source to generate a single line in an outbound document.
- The **Prefix3** element generates **P3** symbols to indicate that the line that is added contains the total tax value.
- The **TotalTaxAmount** element is bound to **model.Data.Summary.Total** to generate the sum of the tax values of the processed tax transactions.
- The **ExecutionDateTime** element generates the date and time (including milliseconds) when the summary line is added.



## Run the imported format

1. On the **Format designer** page, select **Run**.
2. Download the file that the web browser offers, and open it for review.

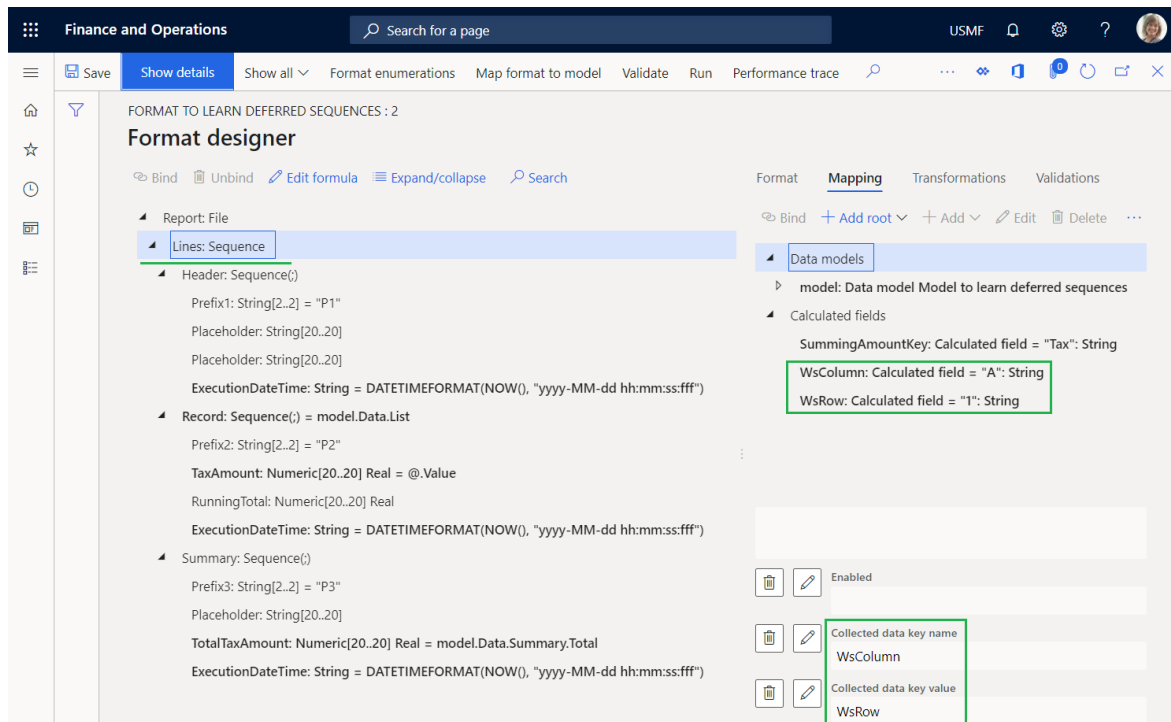


Notice that summary line 22 presents the sum of the tax values for the processed transactions. Because the format is configured to use the `model.Data.Summary.Total` binding to return this sum, the sum is calculated by calling the `TotalSum` aggregation of the `Grouped` data source of the `GroupBy` type that uses the model mapping. To calculate this aggregation, model mapping iterates over all transactions that have been selected in the `Filtered` data source. By comparing the execution times of lines 21 and 22, you can determine that calculation of the sum took 10 milliseconds (ms). By comparing the execution times of lines 2 and 21, you can determine that generation of all transactional lines took 7 ms. Therefore, a total of 17 ms was required.

### Modify the format so that the summing is based on generated output

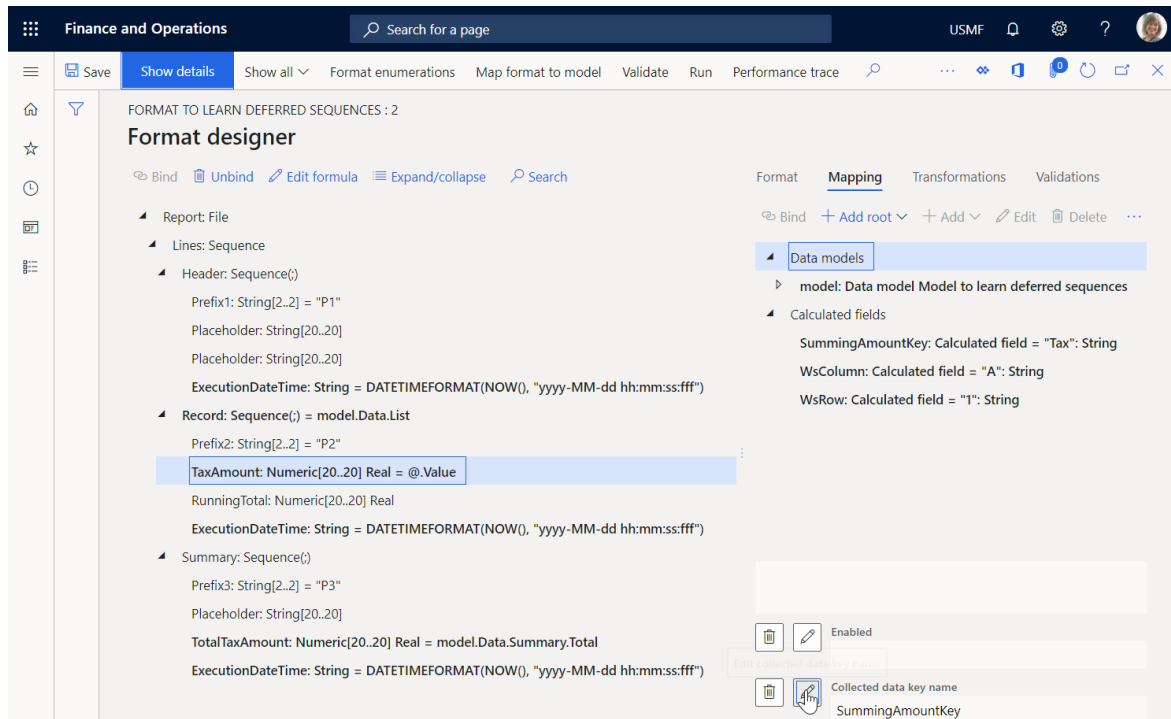
If the volume of transactions is much larger than the volume in the current example, the summing time might increase and cause performance issues. By changing the setting of the format, you can help prevent these performance issues. Because you access tax values to include them in the generated report, you can reuse this information to sum tax values. For more information, see [Configure format to do counting and summing](#).

1. On the **Format designer** page, on the **Format** tab, select the **Report** file element in the format tree.
2. Set the **Collect output details** option to **Yes**. You can now configure this format by using the content of an existing report as a data source that can be accessed by using the built-in ER functions in the **Data collection** category.
3. On the **Mapping** tab, select the **Report\Lines** sequence element.
4. Configure the **Collected data key name** expression as `WsColumn`.
5. Configure the **Collected data key value** expression as `WsRow`.



6. Select the **Report\Lines\Record\TaxAmount** numeric element.

7. Configure the **Collected data key name** expression as `SummingAmountKey`.

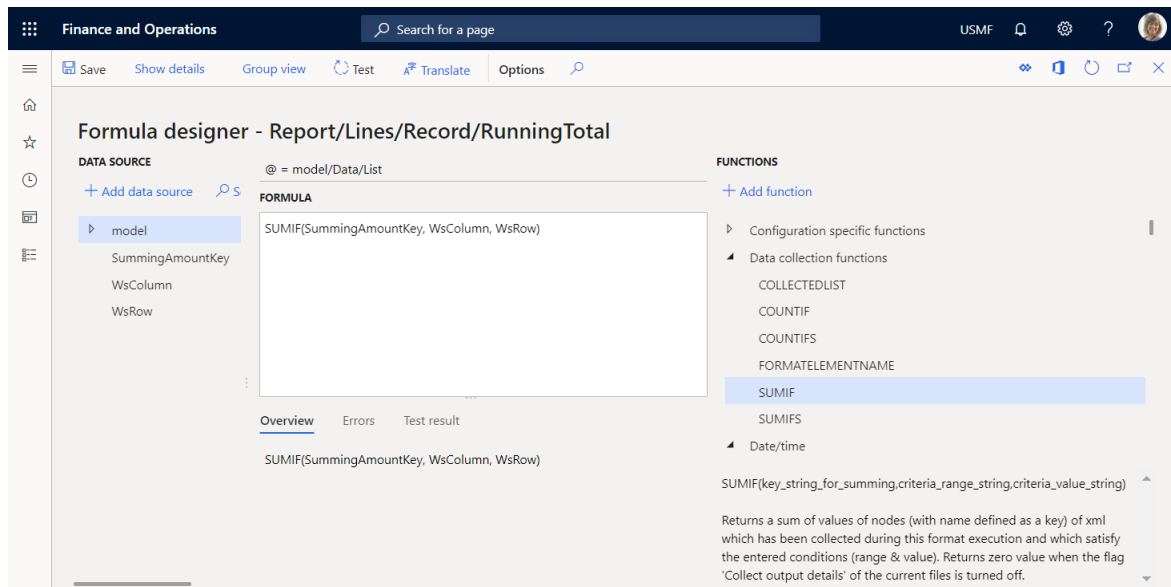


You can consider this setting the fulfillment of a virtual worksheet, where the value of cell A1 is appended with the value of the tax amount from every processed tax transaction.

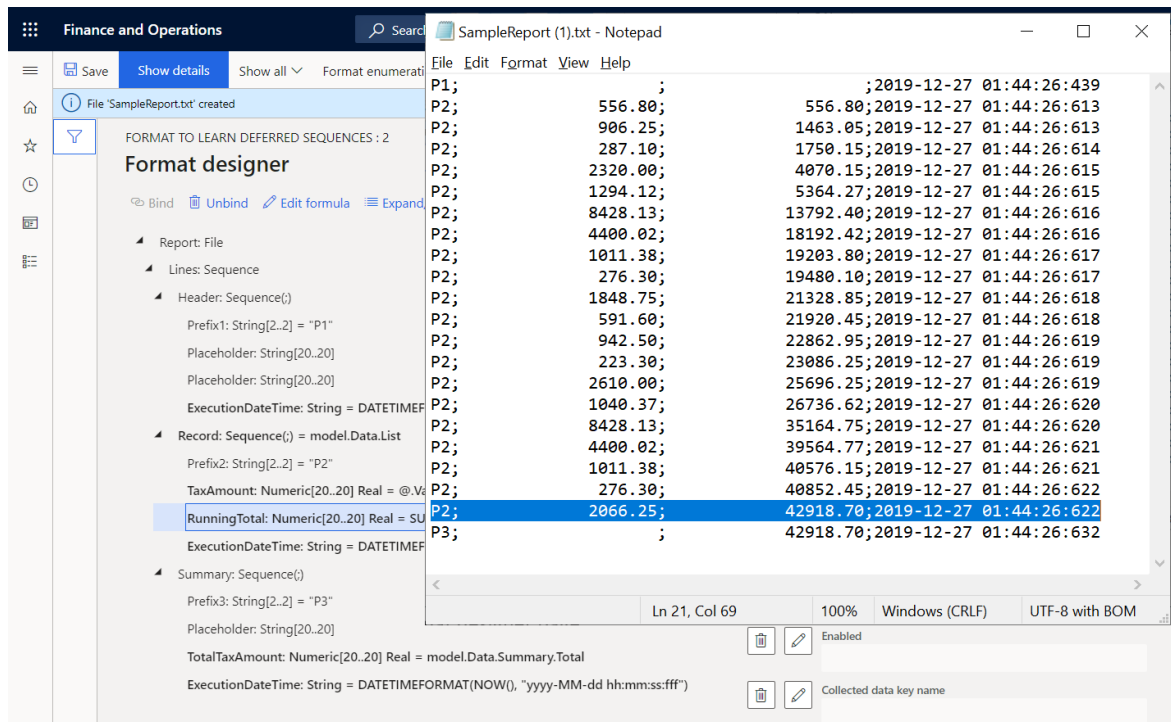
8. Select the **Report\Lines\Record\RunningTotal** numeric element, and then select **Edit formula**.

9. Configure the `SUMIF(SummingAmountKey, WsColumn, WsRow)` expression by using the built-in **SUMIF** ER function.

10. Select **Save**.



11. Close the Formula designer page.
12. Select **Save**, and then select **Run**.
13. Download and review the file that the web browser offers.

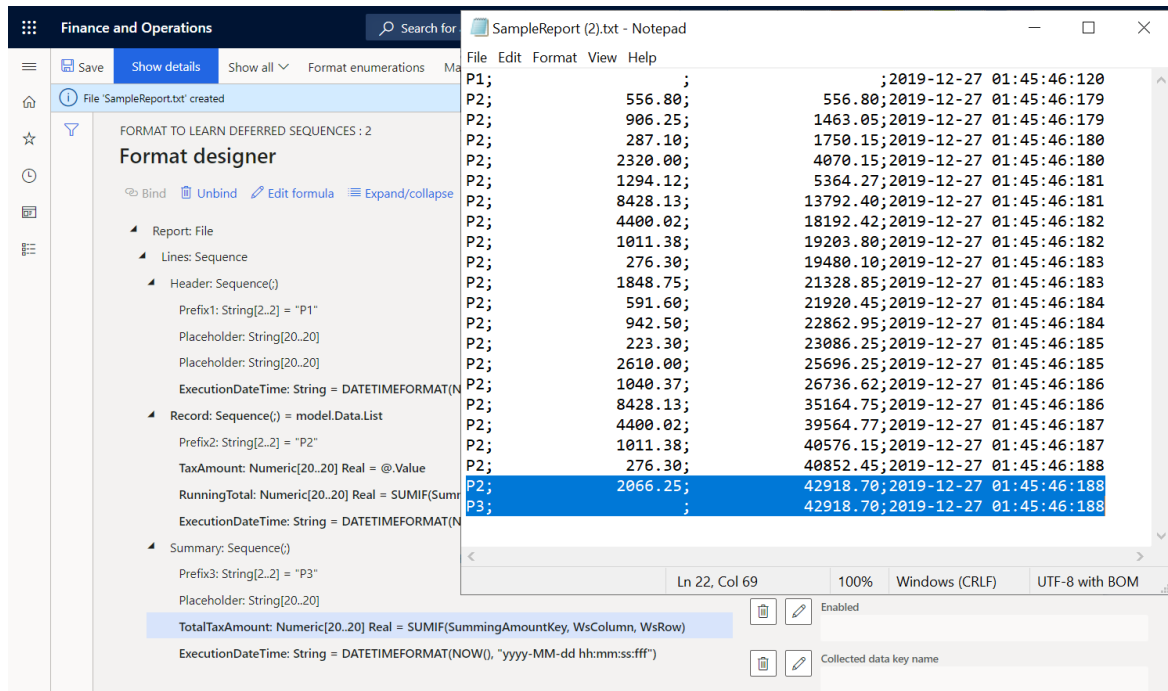


Line 21 contains the running total of tax values that is calculated for all processed transactions by using the generated output as a data source. This data source starts from the beginning of the report and continues through the last tax transaction. Line 22 contains the sum of the tax values for all processed transactions that are calculated in the model mapping by using the data source of the **GroupBy** type. Notice that these values are equal. Therefore, the output-based summing can be used instead of **GroupBy**. By comparing the execution times of lines 2 and 21, you can determine that generation of all the transactional lines and summing took 9 ms. Therefore, as far as the generation of detailed lines and the summing of tax values are concerned, the modified format is approximately two times faster than the original format.

14. Select the **Report\Lines\Summary>TotalTaxAmount** numeric element, and then select **Edit formula**.
15. Enter the `SUMIF(SummingAmountKey, WsColumn, WsRow)` expression instead of the existing expression.

16. Select **Save**, and then select **Run**.

17. Download and review the file that the web browser offers.

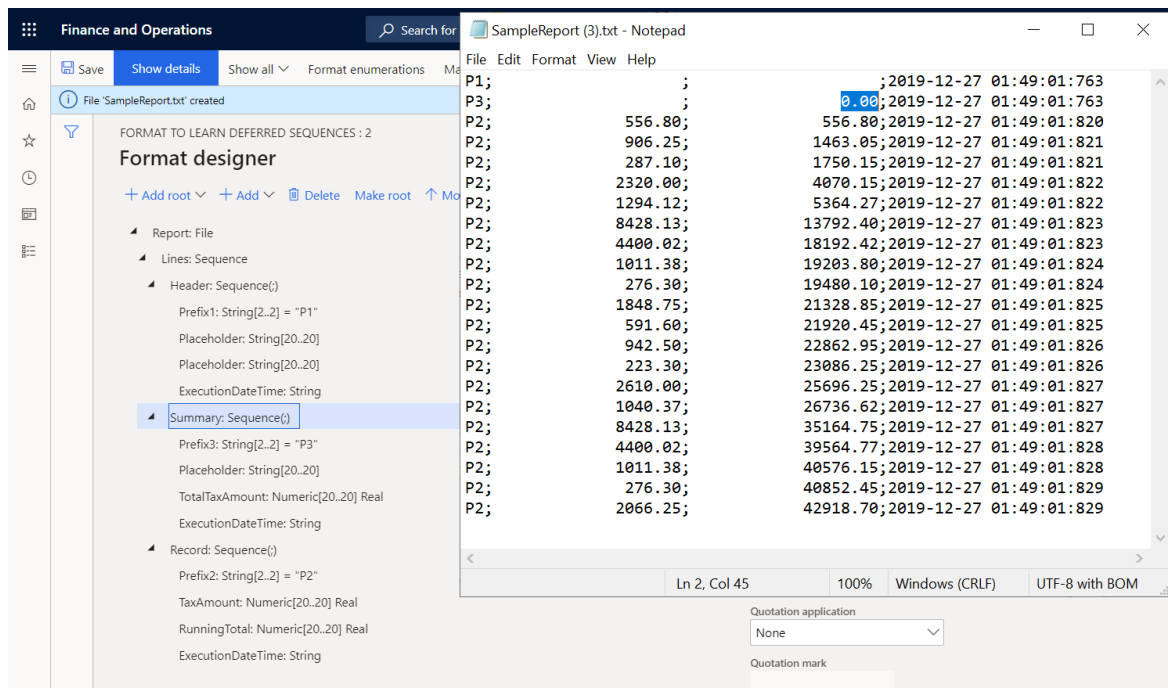


Notice that the running total of tax values on the last transaction details line now equals the sum on the summary line.

### Put values of output-based summing in the report header

If, for example, you must present the sum of tax values in the header of your report, you can modify your format.

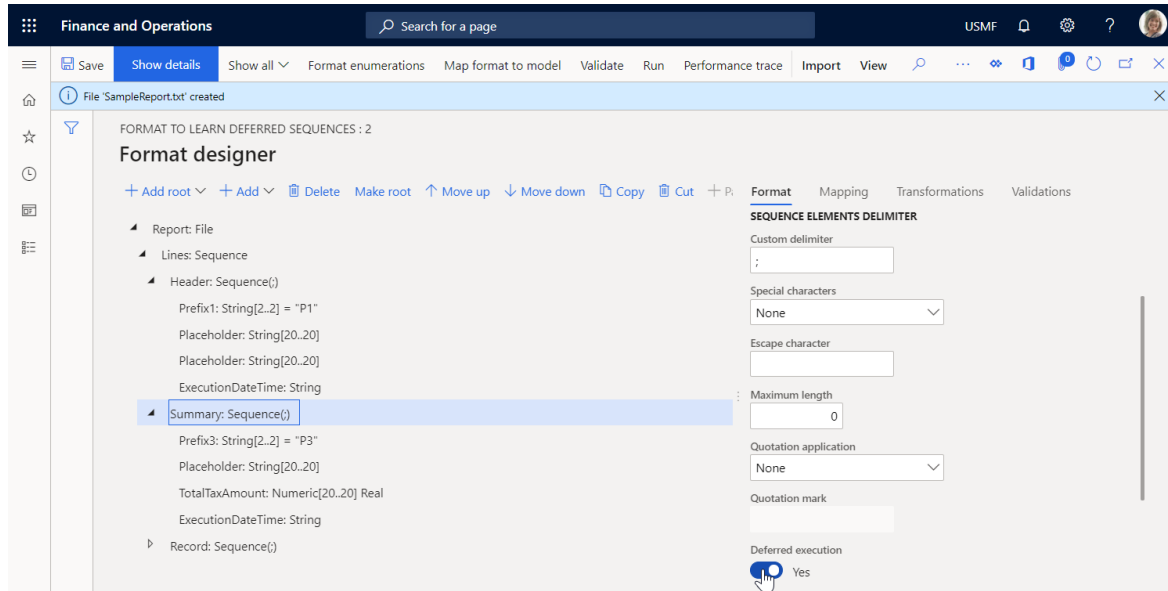
1. On the **Format designer** page, on the **Format** tab, select the **Report\Lines\Summary** sequence element.
2. Select **Move up**.
3. Select **Save**, and then select **Run**.
4. Download and review the file that the web browser offers.



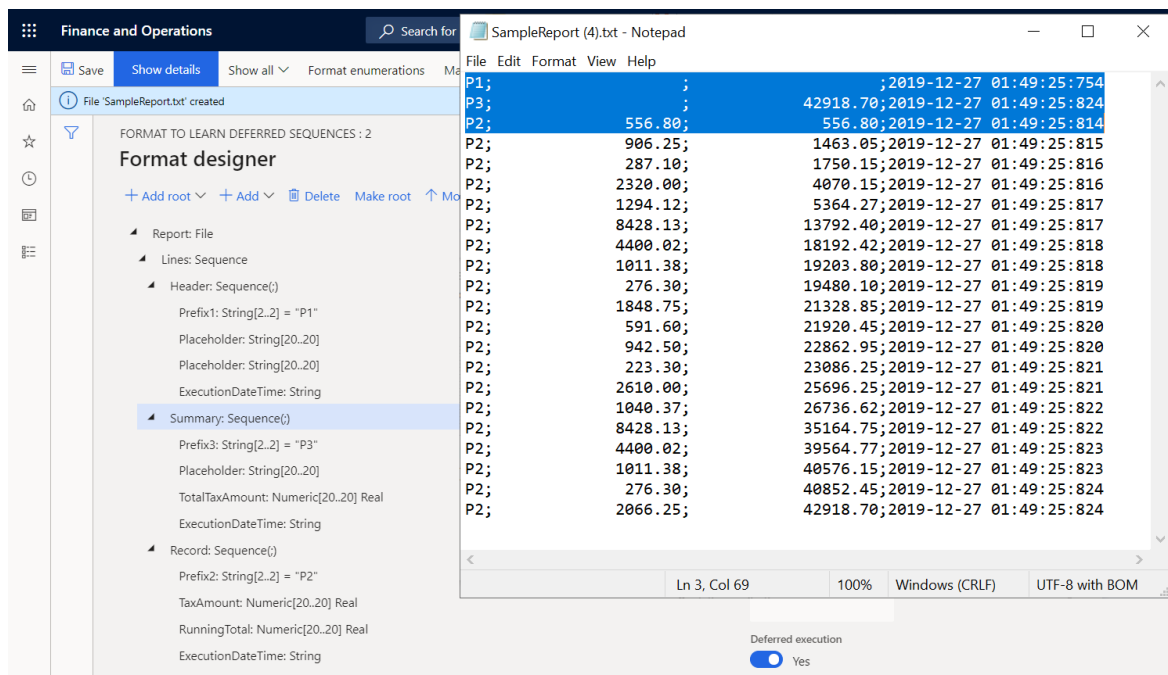
Notice that the sum of tax values on summary line 2 now equals 0 (zero), because this sum is now calculated based on the generated output. When line 2 is generated, the generated output doesn't yet contain lines that have transaction details. You can configure this format to defer the execution of the **Report\Lines\Summary** sequence element until the **Report\Lines\Record** sequence element has been run for all tax transactions.

### Defer the execution of the summary sequence so that the calculated total is used

1. On the **Format designer** page, on the **Format** tab, select the **Report\Lines\Summary** sequence element.
2. Set the **Deferred execution** option to **Yes**.



3. Select **Save**, and then select **Run**.
4. Download and review the file that the web browser offers.



The **Report\Lines\Summary** sequence element is now run only after all other items that are nested under its parent element, **Report\Lines**, have been run. Therefore, it's run after the **Report\Lines\Record** sequence element has been run for all tax transactions of the model.Data.List data source. The execution times of lines 1, 2, and 3, and of the last line, 22, reveal this fact.



## Additional resources

- [Configure format to do counting and summing](#)
- [Trace execution of ER format to troubleshoot performance issues](#)
- [Defer the execution of XML elements in ER formats](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Defer the execution of XML elements in ER formats

2/18/2021 • 11 minutes to read • [Edit Online](#)

## Overview

You can use the Operations designer of the [Electronic reporting \(ER\)](#) framework to [configure](#) the [format component](#) of an ER solution that is used to generate outbound documents in XML format. The hierarchical structure of the configured format component consists of format elements of various types. These format elements are used to fill generated documents with the required information at runtime. By default, when you run an ER format, the format elements are run in the same order as they are presented in the format hierarchy: one by one, from top to bottom. However, at design time, you can change the execution order for any XML elements of the configured format component.

By turning on the **Deferred execution** option for an XML element in the configured format, you can defer (postpone) the execution of that element. In this case, the element isn't run until all other elements of its parent have been run.

To learn more about this feature, complete the example in this topic.

## Limitations

The **Deferred execution** option is supported only for XML elements that are configured for an ER format that is used to generate **outbound** documents in XML format.

The **Deferred execution** option is supported only for XML elements that reside in only one other XML element. Therefore, it isn't applicable to XML elements that reside in other types of format elements (for example, in an **XML sequence** element).

The **Deferred execution** option isn't supported for XML elements that reside in the **Common\File** format element when the **Split file** option is set to **Yes**. For more information about how to split XML files, see [Split generated XML files based on file size and content quantity](#).

## Example: Defer the execution of an XML element in an ER format

The following steps explain how a user in the System administrator or Electronic reporting functional consultant [role](#) can configure an ER format that contains an XML element where the order of execution differs from the order in the format hierarchy.

These steps can be performed in the **USMF** company in Microsoft Dynamics 365 Finance.

### Prerequisites

To complete this example, you must have access to the **USMF** company in Finance for one of the following roles:

- Electronic reporting functional consultant
- System administrator

If you haven't yet completed the example in the [Defer the execution of sequence elements in ER formats](#) topic, download the following [configurations](#) of the sample ER solution.

CONTENT DESCRIPTION	FILE NAME
ER data model configuration	<a href="#">Model to learn deferred elements.version.1.xml</a>

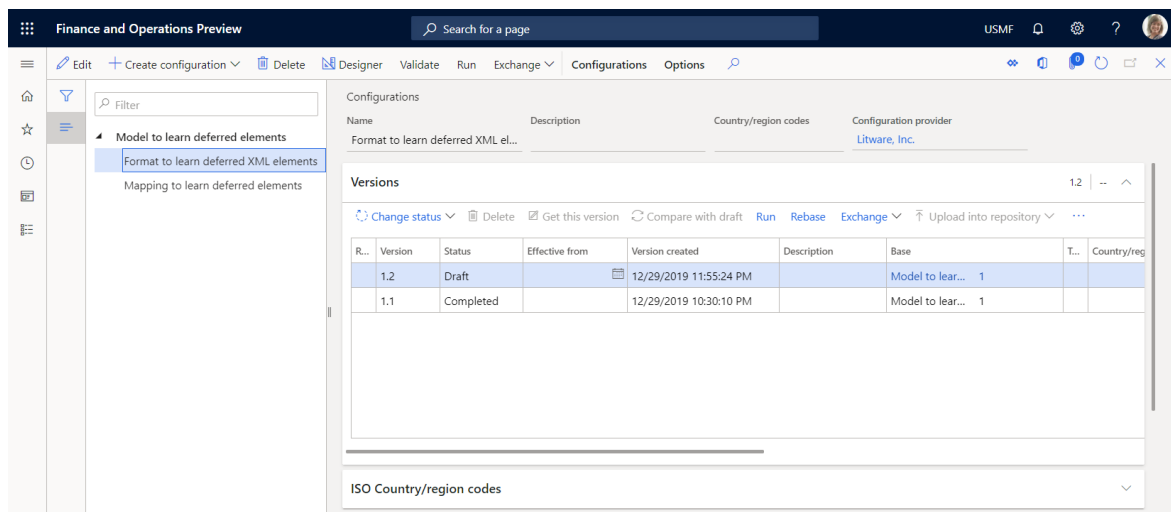
CONTENT DESCRIPTION	FILE NAME
ER model mapping configuration	<a href="#">Mapping to learn deferred elements.version.1.1.xml</a>

Before you begin, you must also download and save the following configuration of the sample ER solution to your local computer.

CONTENT DESCRIPTION	FILE NAME
ER format configuration	<a href="#">Format to learn deferred XML elements.version.1.1.xml</a>

### Import the sample ER configurations

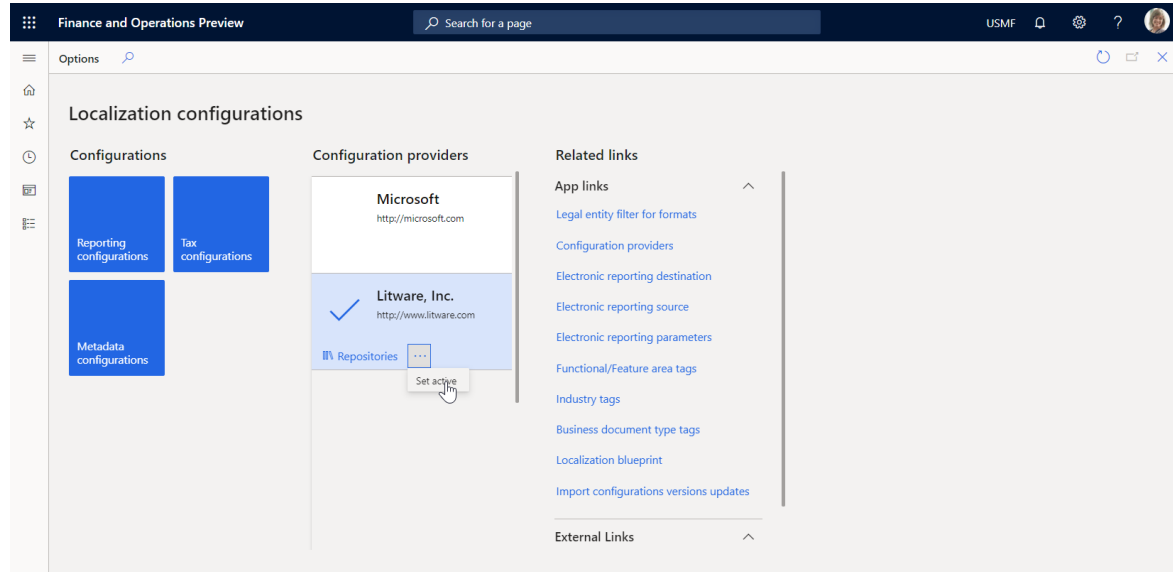
1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Select **Reporting configurations**.
3. On the **Configurations** page, if the **Model to learn deferred elements** configuration isn't available in the configuration tree, import the ER data model configuration:
  - a. Select **Exchange**, and then select **Load from XML file**.
  - b. Select **Browse**, find and select the **Model to learn deferred elements.1.xml** file, and then select **OK**.
4. If the **Mapping to learn deferred elements** configuration isn't available in the configuration tree, import the ER model mapping configuration:
  - a. Select **Exchange**, and then select **Load from XML file**.
  - b. Select **Browse**, find and select the **Mapping to learn deferred elements.1.1.xml** file, and then select **OK**.
5. Import the ER format configuration:
  - a. Select **Exchange**, and then select **Load from XML file**.
  - b. Select **Browse**, find and select the **Format to learn deferred XML elements.1.1.xml** file, and then select **OK**.
6. In the configuration tree, expand **Model to learn deferred elements**.
7. Review the list of imported ER configurations in the configuration tree.



### Activate a configuration provider

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. On the **Localization configurations** page, in the **Configuration providers** section, make sure that

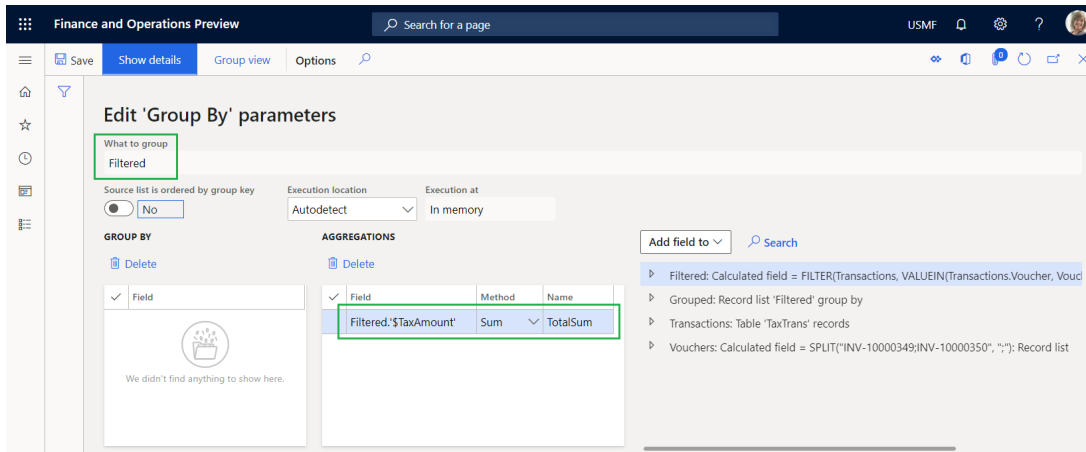
the [configuration provider](#) for the Litware, Inc. (<http://www.litware.com>) sample company is listed, and that it's marked as active. If this configuration provider isn't listed, or if it isn't marked as active, follow the steps in the [Create a configuration provider and mark it as active](#) topic.



### Review the imported model mapping

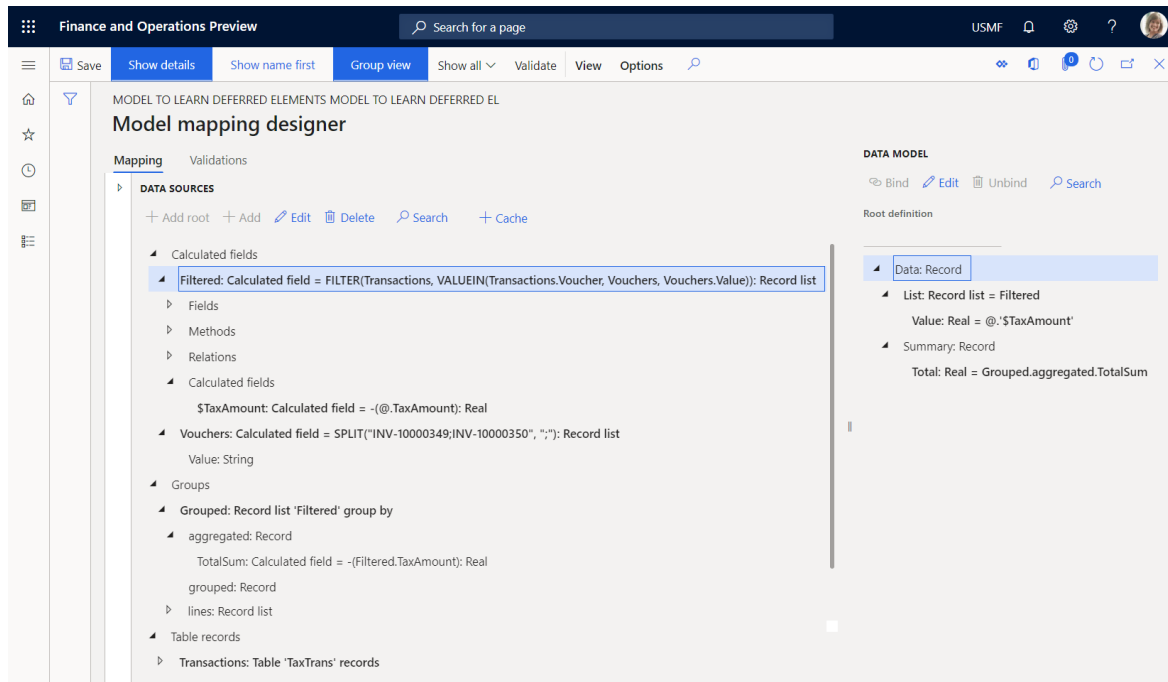
Review the settings of the ER model mapping component that is configured to access tax transactions and expose accessed data on request.

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. Select **Reporting configurations**.
3. On the **Configurations** page, in the configuration tree, expand **Model to learn deferred elements**.
4. Select the **Mapping to learn deferred elements** configuration.
5. Select **Designer** to open the list of mappings.
6. Select **Designer** to review the mapping details.
7. Select **Show details**.
8. Review the data sources that are configured to access tax transactions:
  - The **Transactions** data source of the *Table record* type is configured to access records of the **TaxTrans** application table.
  - The **Vouchers** data source of the *Calculated field* type is configured to return the required voucher codes (**INV-10000349** and **INV-10000350**) as a list of records.
  - The **Filtered** data source of the *Calculated field* type is configured to select, from the **Transactions** data source, only tax transactions of the required vouchers.
  - The **\$TaxAmount** field of the *Calculated field* type is added for the **Filtered** data source to expose the tax value that has the opposite sign.
  - The **Grouped** data source of the *Group By* type is configured to group filtered tax transactions of the **Filtered** data source.
  - The **TotalSum** aggregation field of the **Grouped** data source is configured to summarize values of the **\$TaxAmount** field of the **Filtered** data source for all filtered tax transactions of that data source.



9. Review how the configured data sources are bound to the data model, and how they expose accessed data to make it available in an ER format:

- The **Filtered** data source is bound to the **Data.List** field of the data model.
- The **\$TaxAmount** field of the **Filtered** data source is bound to the **Data.List.Value** field of the data model.
- The **TotalSum** field of the **Grouped** data source is bound to the **Data.Summary.Total** field of the data model.



10. Close the **Model mapping designer** and **Model mappings** pages.

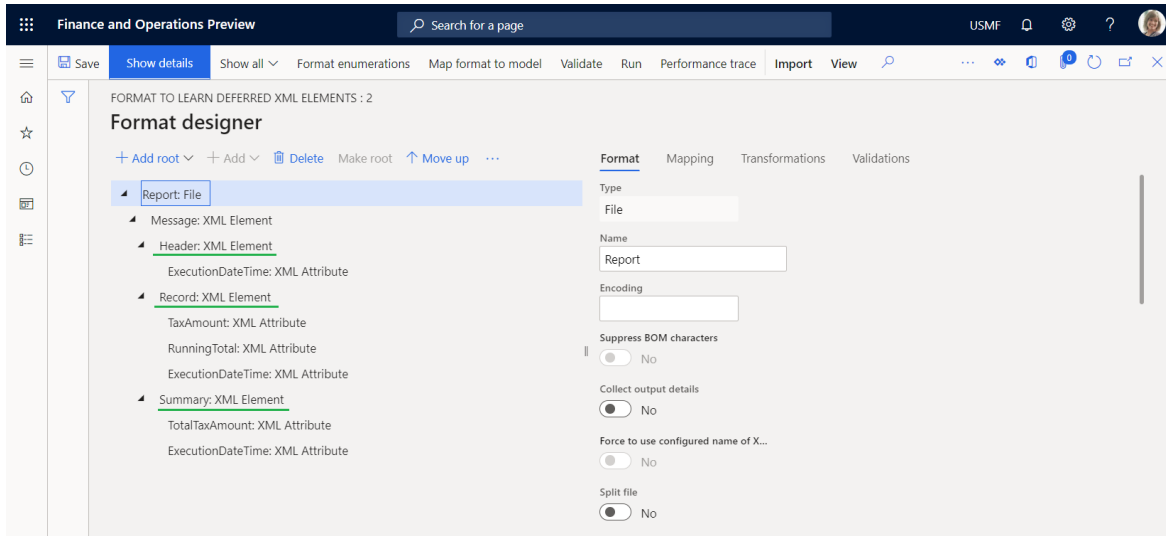
### Review the imported format

1. On the **Configurations** page, in the configuration tree, select the **Format to learn deferred XML elements** configuration.
2. Select **Designer** to review the format details.
3. Select **Show details**.
4. Review the settings of the ER format components that are configured to generate an outbound document in XML format that includes details of the tax transactions:

- The **Report\Message** XML element is configured to fill the outbound document with a single node that includes the nested XML elements (**Header**, **Record**, and **Summary**).
- The **Report\Message\Header** XML element is configured to fill the outbound document with a

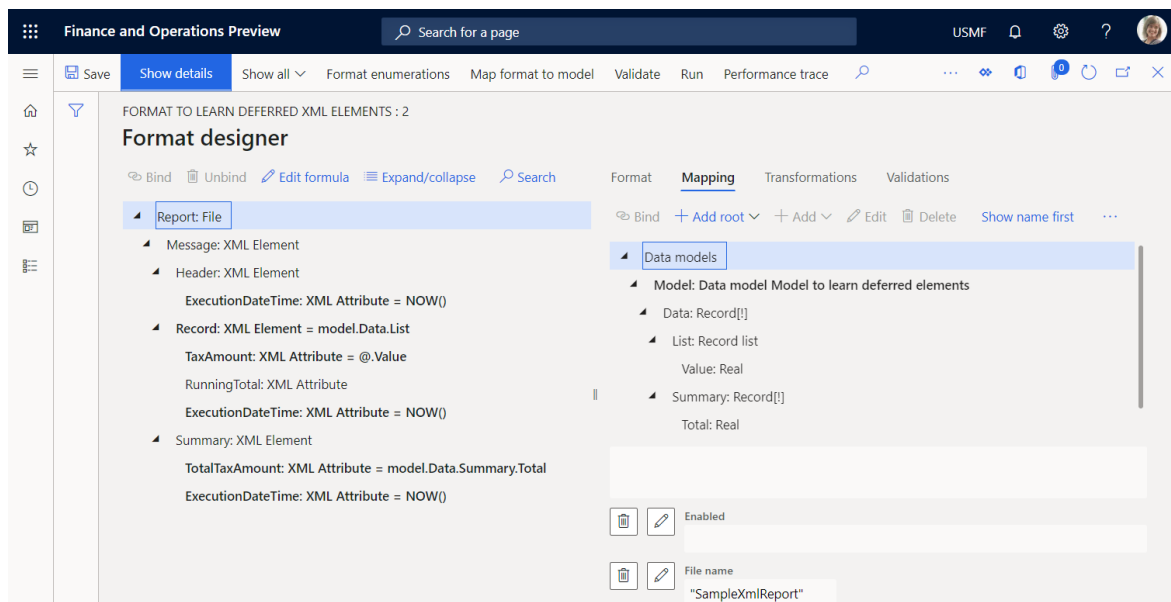
single header node that shows the date and time when the processing starts.

- The **Report \Message \Record** XML element is configured to fill the outbound document with a single record node that shows the details of a single tax transaction.
- The **Report \Message \Summary** XML element is configured to fill the outbound document with a single summary node that includes the sum of the tax values from the processed tax transactions.



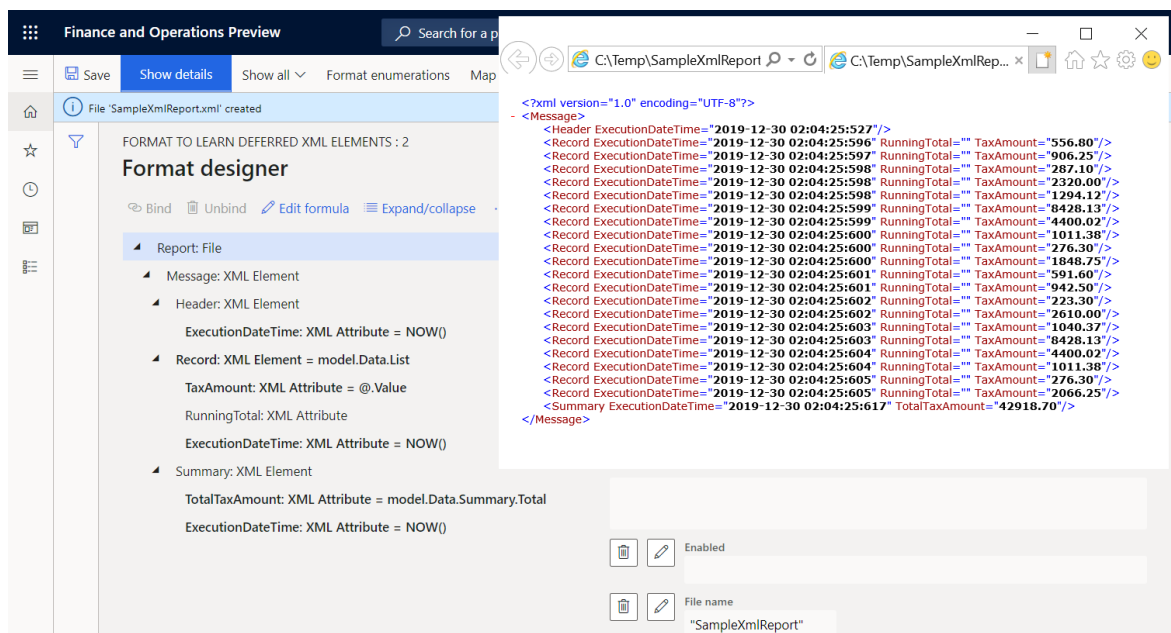
5. On the **Mapping** tab, review the following details:

- The **Report \Message \Header** element doesn't have to be bound to a source to generate a single node in an outbound document.
- The **ExecutionDate** attribute generates the date and time (including milliseconds) when the header node is added.
- The **Report \Message \Record** element is bound to the **model.Data.List** list to generate a single record node for every record from the bound list.
- The **TaxAmount** attribute is bound to **model.Data.List.Value** (which is shown as **@.Value** in the relative path view) to generate the tax value of the current tax transaction.
- The **RunningTotal** attribute is a placeholder for the running total of the tax values. Currently, this attribute has no output, because neither a binding nor a default value is configured for it.
- The **ExecutionDate** attribute generates the date and time (including milliseconds) when the current transaction is processed in this report.
- The **Report \Message \Summary** element doesn't have to be bound to a data source to generate a single node in an outbound document.
- The **TotalTaxAmount** attribute is bound to **model.Data.Summary.Total** to generate the sum of the tax values of the processed tax transactions.
- The **ExecutionDate** attribute generates the date and time (including milliseconds) when the summary node is added.



## Run the imported format

1. On the Format designer page, select Run.
2. Download the file that the web browser offers, and open it for review.



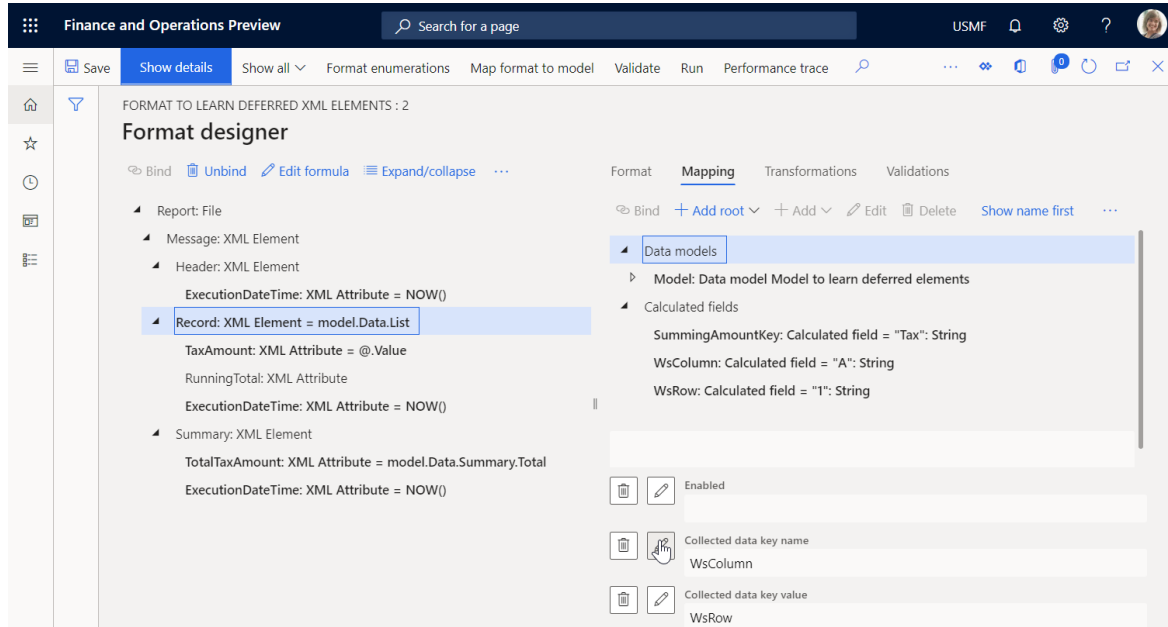
Notice that the summary node presents the sum of the tax values for the processed transactions. Because the format is configured to use the `model.Data.Summary.Total` binding to return this sum, the sum is calculated by calling the `TotalSum` aggregation of the `Grouped` data source of the `GroupBy` type in the model mapping. To calculate this aggregation, the model mapping iterates over all transactions that have been selected in the `Filtered` data source. By comparing the execution times of the summary node and the last record node, you can determine that calculation of the sum took 12 milliseconds (ms). By comparing the execution times of the first and last record nodes, you can determine that generation of all record nodes took 9 ms. Therefore, a total of 21 ms was required.

## Modify the format so that the calculation is based on generated output

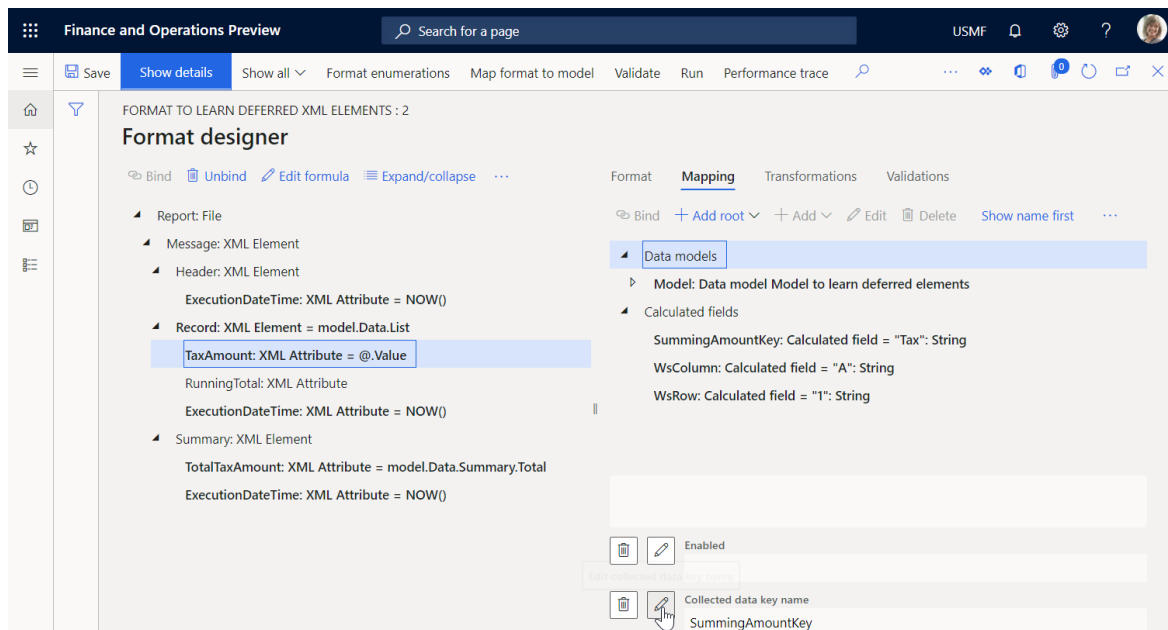
If the volume of transaction is much larger than the volume in the current example, the calculation time might increase and cause performance issues. By changing the setting of the format, you can help prevent these performance issues. Because you access tax values to include them in the generated report, you can reuse this information to calculate tax values. For more information, see [Configure format to do counting and summing](#).

1. On the Format designer page, on the Format tab, select the Report file element in the format tree.

- Set the **Collect output details** option to **Yes**. You can now configure this format by using the content of a generated report as a data source that can be accessed by using the built-in ER functions in the **Data collection** category.
- On the **Mapping** tab, select the **Report\Message\Record** XML element.
- Configure the **Collected data key name** expression as `WsColumn`.
- Configure the **Collected data key value** expression as `WsRow`.



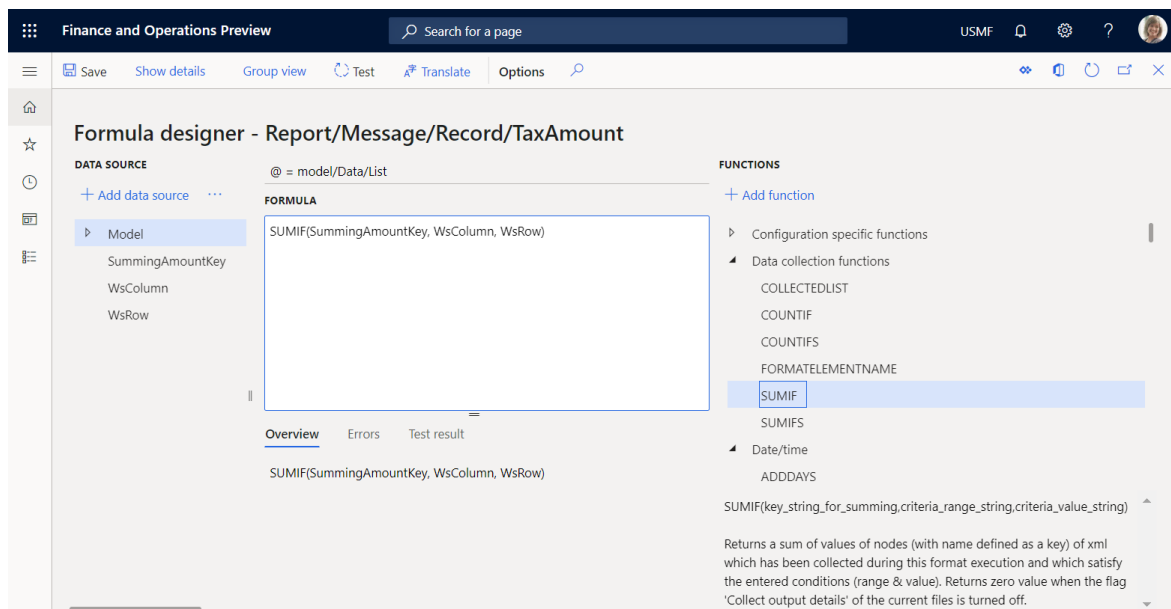
- Select the **Report\Message\Record\TaxAmount** attribute.
- Configure the **Collected data key name** expression as `SummingAmountKey`.



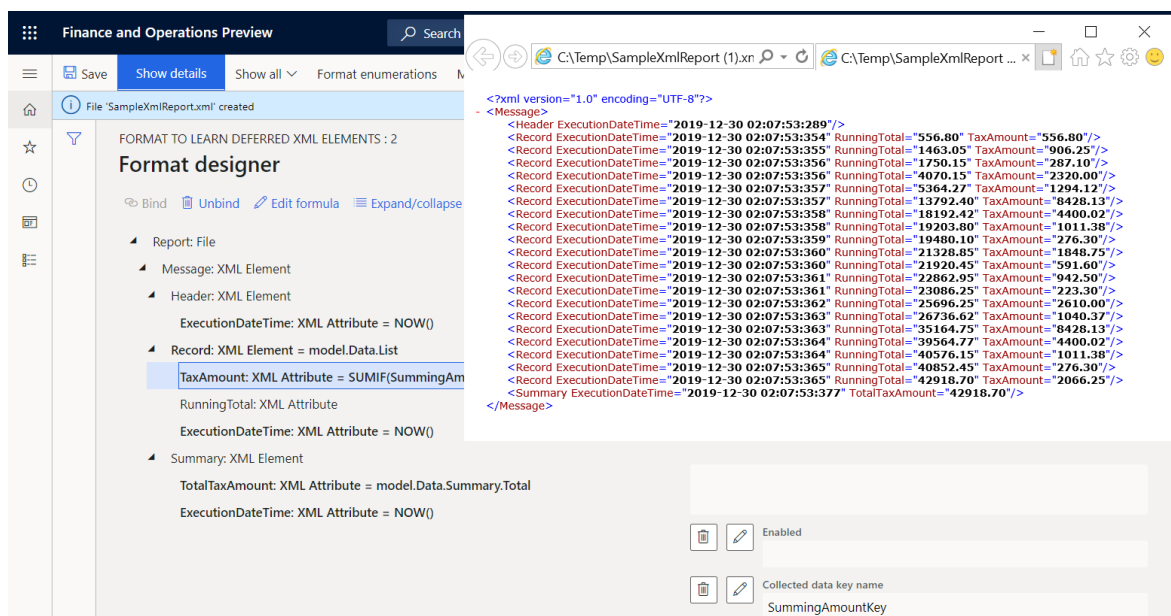
You can consider this setting the fulfillment of a virtual worksheet, where the value of cell A1 is appended with the value of the tax amount from every processed tax transaction.

- Select the **Report\Message\Record\RunningTotal** attribute, and then select **Edit formula**.
- Configure the `SUMIF(SummingAmountKey, WsColumn, WsRow)` expression by using the built-in **SUMIF** ER function, and then select **Save**.



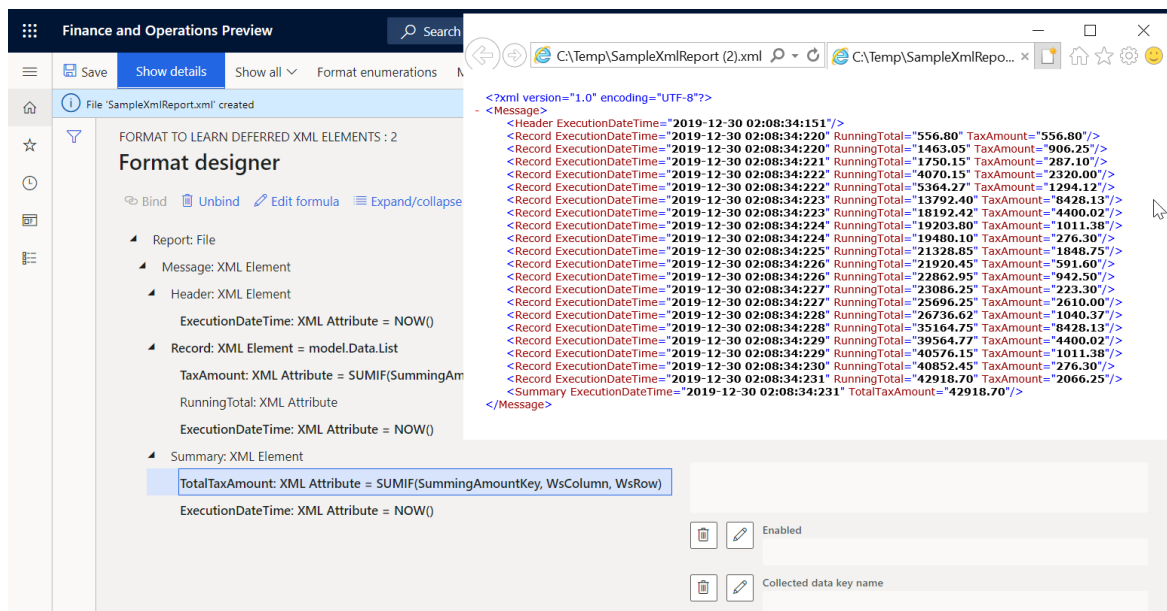


10. Close the Formula designer page.
11. Select **Save**, and then select **Run**.
12. Download and review the file that the web browser offers.



The last record node contains the running total of tax values that is calculated for all processed transactions by using the generated output as a data source. This data source starts from the beginning of the report and continues through the last tax transaction. The summary node contains the sum of the tax values for all processed transactions that are calculated in the model mapping by using the data source of the *GroupBy* type. Notice that these values are equal. Therefore, the output-based summing can be used instead of **GroupBy**. By comparing the execution times of the first record node and the summary node, you can determine that generation of all the record nodes and summing took 11 ms. Therefore, as far as the generation of record nodes and the summing of tax values are concerned, the modified format is approximately two times faster than the original format.

13. Select the **Report\Message\Summary\TotalTaxAmount** attribute, and then select **Edit formula**.
14. Enter the `SUMIF(SummingAmountKey, WsColumn, WsRow)` expression instead of the existing expression.
15. Select **Save**, and then select **Run**.
16. Download and review the file that the web browser offers.

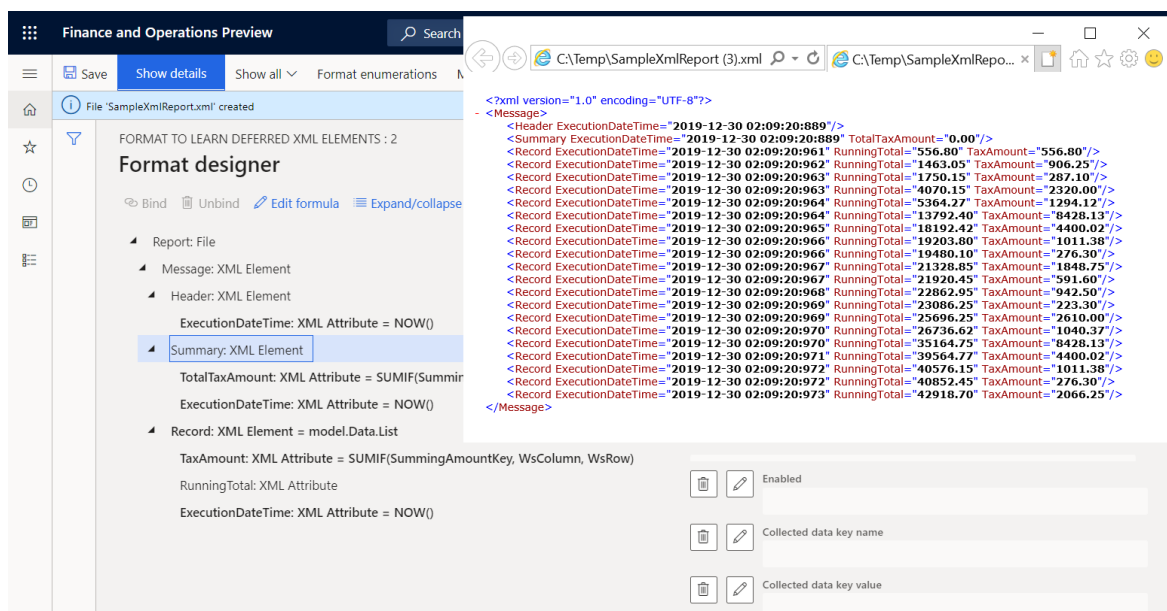


Notice that the running total of tax values in the last record node now equals the sum in the summary node.

### Put values of output-based summing in the report header

If, for example, you must present the sum of tax values in the header of your report, you can modify your format.

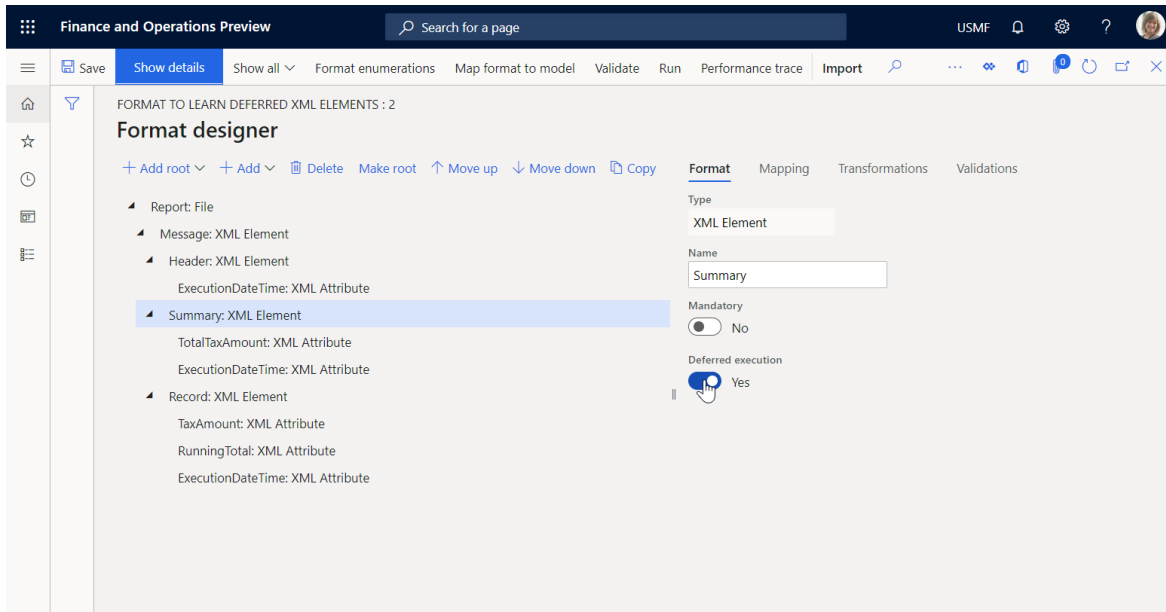
1. On the Format designer page, on the Format tab, select the Report\Message\Summary XML element.
2. Select **Move up**.
3. Select **Save**, and then select **Run**.
4. Download and review the file that the web browser offers.



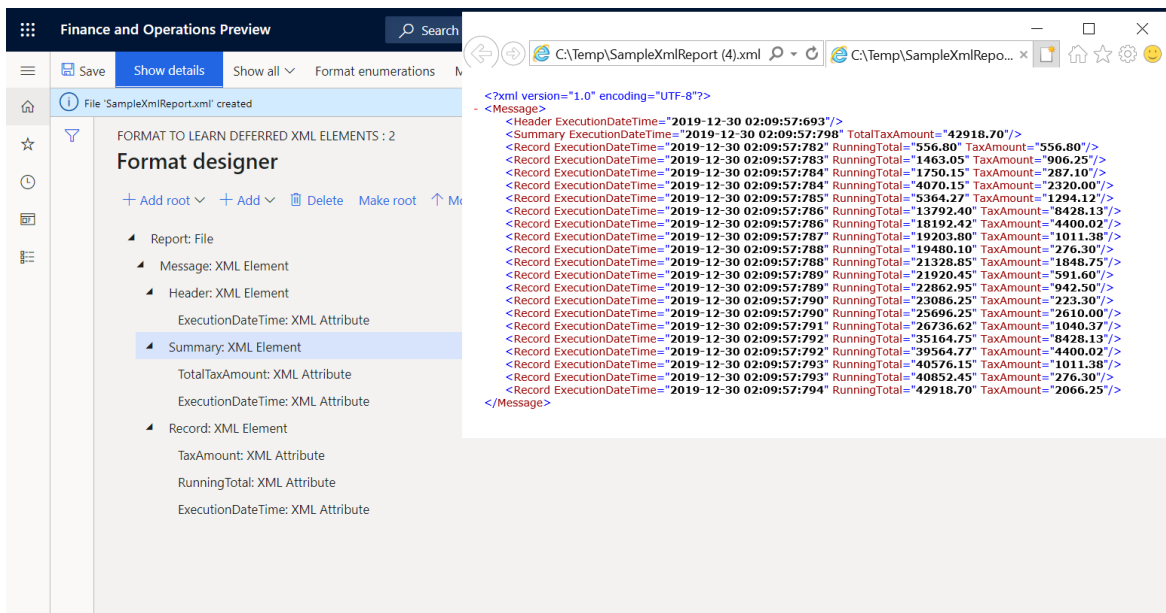
Notice that the sum of tax values in the summary node now equals 0 (zero), because this sum is now calculated based on the generated output. When the first record node is generated, the generated output doesn't yet contain record nodes that have transaction details. You can configure this format to defer the execution of the Report\Message\Summary element until the Report\Message\Record element has been run for all tax transactions.

### Defer the execution of the summary XML element so that the calculated total is used

1. On the **Format designer** page, on the **Format** tab, select the **Report\Message\Summary** XML element.
2. Set the **Deferred execution** option to **Yes**.



3. Select **Save**, and then select **Run**.
4. Download and review the file that the web browser offers.



The **Report\Message\Summary** element is now run only after all other items that are nested under its parent element, **Report\Message**, have been run. Therefore, it's run after the **Report\Message\Record** element has been run for all tax transactions of the **model.Data.List** data source. The execution times of the first and last record nodes, and of the header and summary nodes, reveal this fact.

## Additional resources

- [Configure format to do counting and summing](#)
- [Trace execution of ER format to troubleshoot performance issues](#)
- [Defer the execution of sequence elements in ER formats](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure the Electronic reporting (ER) framework

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic explains how to set up the basic functionality for Electronic reporting (ER). It also describes the steps that you must complete before you can set up ER.

## Prerequisites for ER setup

Before you can set up ER, you must set up the required [document types](#) in Document management:

- A document type for Microsoft Office documents that are used as templates for ER reports.
- A document type that is used to store the output of ER reports in the jobs archive.
- A document type that is used to store the output of ER reports so that they can be viewed in other programs.
- A document type that is used to keep baselines of outputs of ER configurations.
- A document type that is used to handle files in the ER framework for all other purposes.

For each document type, the following attribute values can be selected.

ATTRIBUTE NAME	ATTRIBUTE VALUE
Class	Attach file
Group	File
Location	Azure storage or SharePoint

## Set up ER

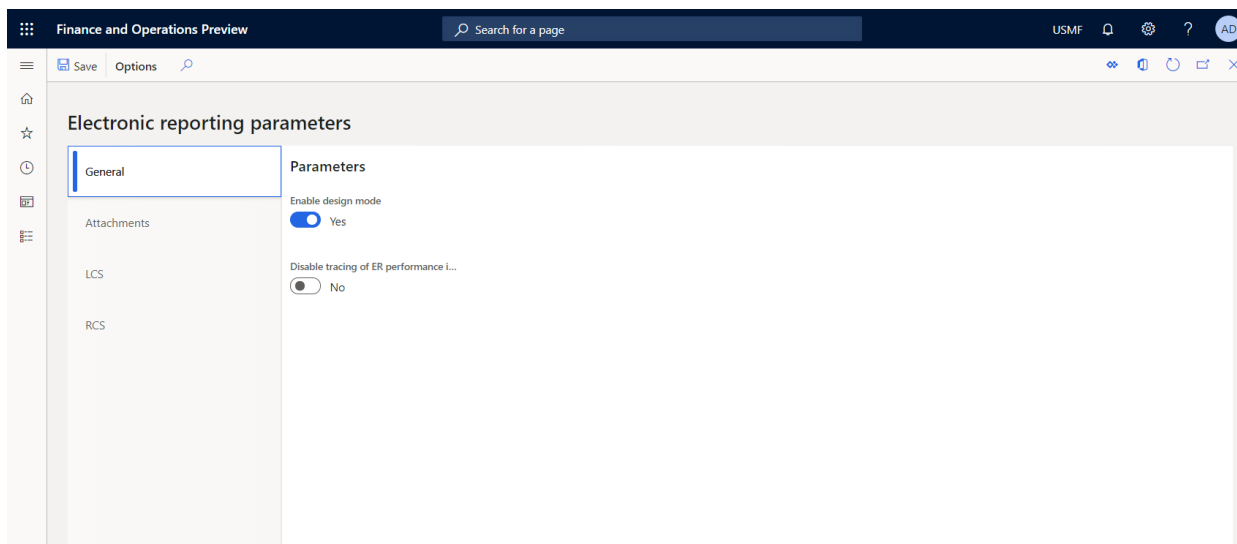
Use the following procedure to set up the basic functionality of ER for all legal entities.

1. Open the **Electronic reporting** workspace page.
2. Click **Electronic reporting parameters**.

## Main parameters

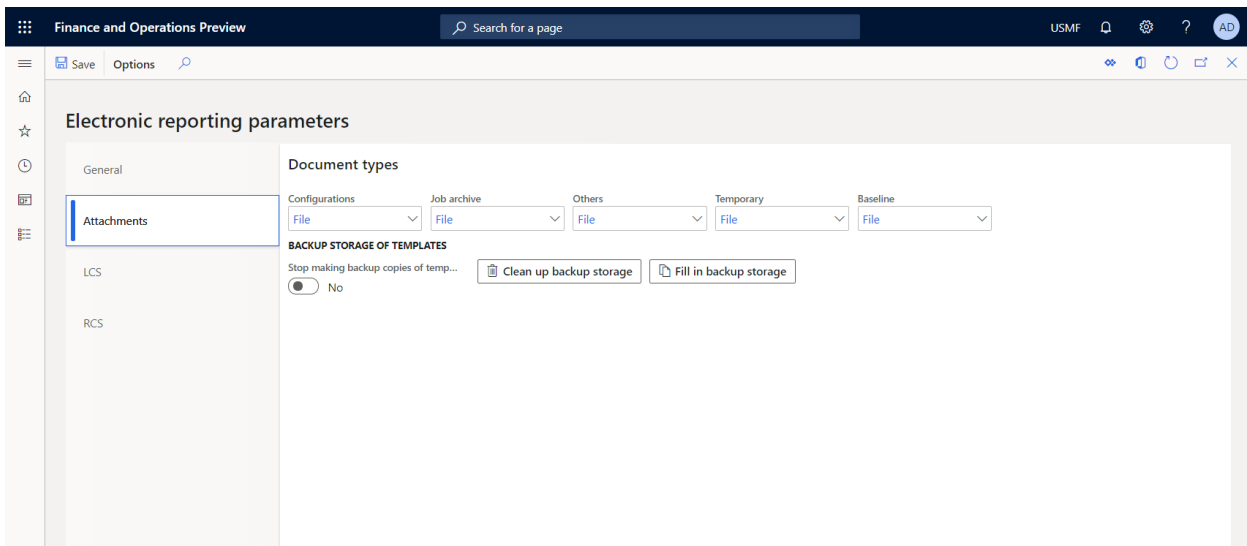
On the **General** tab of the **Electronic reporting parameters** page, set the following ER parameters:

- **Enable design mode**
  - Set this option to **Yes** to enable embedded ER designers, so that users can create their own ER configurations.
  - Set this option to **No** to require that users access the functionality of ER designers by signing up for [Configuration service](#).
- **Disable tracing of ER performance in data handling**
  - Set this option to **No** to allow Microsoft Telemetry to collect information about the average time that is required to process a single incoming or outgoing record as an ER configuration. This information is tracked as a specific health metric of the environment, and it will help Microsoft quickly identify and address issues that affect customers who use the ER framework.
  - Set this option to **Yes** to stop collecting telemetry information.



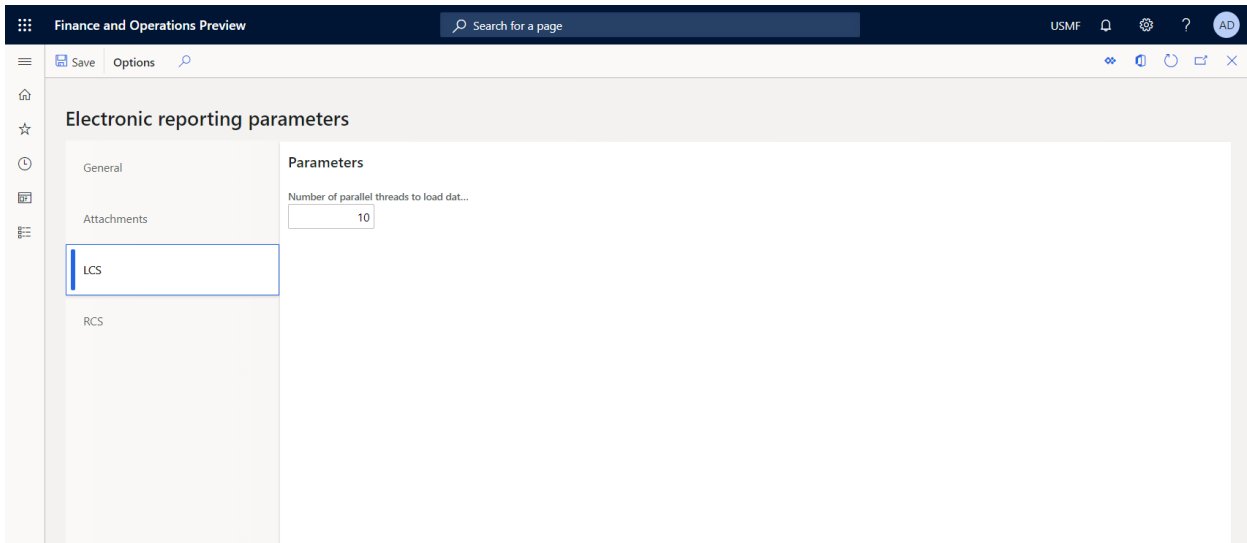
On the **Attachments** tab of the **Electronic reporting parameters** page, set the following ER parameters:

- **Configurations** – Select a document type to specify the storage of templates of ER formats. This document type is selected within the scope of a specific company. This document type will be used, regardless of the company that a user is signed in to while an ER format is being used.
  - **Job archive** – Select a document type to specify the storage of generated documents that are attached to the records of the ER jobs [archive](#). This document type is selected as a company-specific document type. You must make sure that the selected document type is configured for every company that you plan to run ER formats for, and then store the results in the ER jobs archive.
  - **Temporary** – Select a document type to specify the storage of generated documents that are used for other purposes. For example, documents might be generated for [preview](#) by other services. This document type is selected as a company-specific document type. You must make sure that the selected document type is configured for every company that you plan to run ER formats for, and then store the results in the ER jobs archive.
  - **Baseline** – Select a document type to specify the storage of documents that are used as [baselines](#) during the automated testing of ER configurations. This document type is selected as a company-specific document type. You must make sure that the selected document type is configured for every company that you plan to run ER formats for, and then store the results in the ER jobs archive.
  - **Others** – Select a document type to specify the storage of generated documents that are used for all other purposes. This document type is selected as a company-specific document type. You must make sure that the selected document type is configured for every company that you plan to run ER formats for, and then store the results in the ER jobs archive.
  - **Stop making backup copies of templates**
    - Set this option to **No** to automatically create a backup copy of any ER format configuration template and store the copy in the database storage.
    - Set this option to **Yes** to stop making backup copies of ER formation configuration templates.
- For more information, see [Backup storage of ER templates](#).



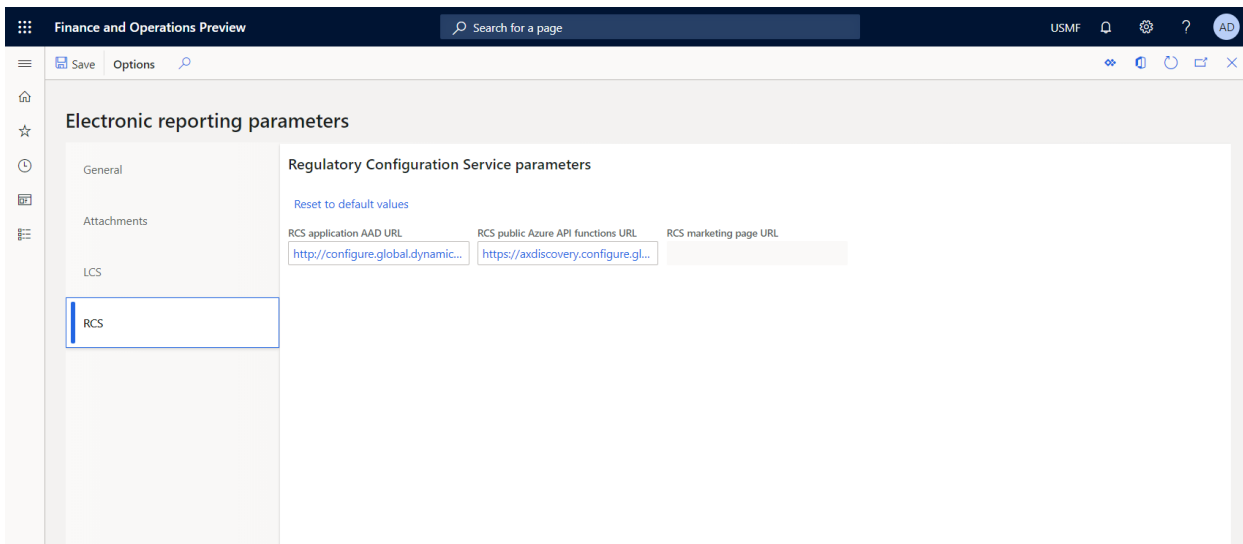
## LCS parameters

On the LCS tab of the **Electronic reporting parameters** page, define the number of parallel threads that should be used to load an ER configuration from repositories in Microsoft Dynamics Lifecycle Services (LCS), so that the configurations are loaded in the most efficient manner. The value can vary from 1 through 15, depending on the available resources of the current program. Based on this setting and the number of other tasks and their priorities, the real number of threads will be defined automatically.



## RCS parameters

On the RCS tab of the **Electronic reporting parameters** page, sign up for [Configuration service](#).



## Active ER configurations provider

On the **Configuration provider** table page, create ER provider records. Each provider can be **marked** as **Active**. The active provider's name and internet address are stored in an ER configuration, as attributes of the configuration owner.

## Optional setup for ER

In addition to the basic functionality, ER has other functionality that you can set up.

- On the **Electronic reporting destination** page, define the ER output destinations for each file output of each ER format configuration. Use the **document types** of the Document management framework that you set up earlier. You can also use this page to set up the optional ER functionality for each legal entity. For more information, see **Electronic reporting (ER) destinations**.
- When you add new Application Object Tree (AOT) artifacts or update existing AOT artifacts that are used as data sources (tables, views, or data entities) in ER, use the **Rebuild table references** menu item (**Organization administration** > **Electronic reporting** > **Rebuild table references**) to bring your AOT changes into the ER metadata.

## Frequently asked questions

**Question:** What is the optimal number of parallel threads to use to load an ER configuration from LCS?

**Answer:** To calculate the optimal number of parallel threads, use the following empirical formula:  $\text{Cores} \div 2 + 1(2)$ . For example, if the program runs on a virtual machine (VM) that has two CPUs, and each CPU contains four cores, the optimal number is five or six parallel threads.

**Question:** I have added a custom table to the AOT. I created a new ER model mapping configuration for my ER data model. During the design of the model mapping, I tried to add a new data source type, **Table records**, that refers to my table. I could manually add my table name to the **Table** lookup, and the ER model mapping accepted it without errors or warnings. However, my table's name isn't included in the list of available choices that the **Table** lookup of this data source offers. How do I include the name of my table?

**Answer:** To include the name of your custom table in the **Table** lookup, use the **Rebuild table references** menu item as described in the "Optional setup for ER" section earlier in this topic.

**Question:** Why can't I mark the Microsoft provider as **Active** in the **Electronic reporting** workspace in my production environment?

**Answer:** The Microsoft provider is used to mark ER configurations that have been designed and maintained by



Microsoft. We expect that Microsoft will release new versions of the configurations in the future. We recommend that you not mark the Microsoft provider as **Active**. Otherwise, you can update the configurations. (For example, you can change the content and register new versions.) These updates will cause issues in the future, when Microsoft provides new versions of the configurations, and those new versions must be imported and adopted. Instead, register a new ER provider for your company, and use it for your ER configurations maintenance. To reuse a Microsoft configuration, select it as the base for your derived copy. To incorporate changes that are provided by Microsoft, rebase your configuration to a new version of the Microsoft configuration when it becomes available.

**Question:** I successfully ran an ER format in one company. However, when I ran the same ER format in another company and used the same settings, I received the following error message: "The provided document type is not a File type." Why?

**Answer:** Most likely, the second company doesn't contain document types that have been selected in the **Job archive, Temporary, Baseline, and Others ER parameters**. To fix this issue, configure these document types in the second company.

- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) destinations](#)
- [Configuration service of Regulatory Services](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Backup storage of ER templates

2/18/2021 • 5 minutes to read • [Edit Online](#)

The [Electronic reporting \(ER\) overview](#) lets business users configure formats for outbound documents according to the legal requirements of various countries and regions. Configured ER formats can use predefined templates to generate outbound documents in various formats, such as Microsoft Excel workbooks, Microsoft Word documents, or PDF documents. The templates are filled with data that the configured dataflow for generated documents requires.

Each configured format can be published as part of an ER solution. Each ER solution can be exported from one instance of Finance and Operations and imported into another instance.

The ER framework uses the [Configure document management](#) to keep the required templates for the current Finance and Operations instance. Depending on the settings of the ER framework, Microsoft Azure Blob storage or a Microsoft SharePoint folder can be selected as the physical primary storage location for templates. (For more information, see [Configure the Electronic reporting \(ER\) framework](#).) The DocuValue table holds an individual record for each template. In each record, the **AccessInformation** field stores the path of a template file that is located in the configured storage location.

When you manage your Finance and Operations instances, you might decide to migrate the current instance to another location. For example, you might migrate your production instance to a new sandbox environment. If you configured the ER framework to store templates in Blob storage, the DocuValue table in the new sandbox environment refers to the instance of Blob storage in the production environment. However, this instance can't be accessed from the sandbox environment, because the migration process doesn't support the migration of artifacts in Blob storage. Therefore, if you try to run an ER format that uses a template to generate business documents, an exception occurs, and you're notified about the missing template. You're also guided to use the ER cleanup tool to delete and then re-import the ER format configuration that contains the template. Because you might have several ER format configurations, this process can be time consuming.

The Backup storage of ER templates feature can help you make your templates so that they are always available to generate business documents.

## NOTE

This feature can be used only when Blob storage has been selected as the physical storage location for ER templates.

## Automated recovery and notification

For this feature, every template of a new ER format configuration in the current environment is automatically saved to the backup storage location for templates (the ERDocuDatabaseStorage database table) when the following events occur:

- You import a new ER format configuration that contains a template.
- You complete the draft version of an ER format configuration that contains a template.

Backup copies of templates are migrated to a new instance of Finance and Operations as part of the application database.

If a template of an ER format is required for generation of outbound documents, to process vendor payments including generation of payment advice and control reports, for example, but the required template isn't found in the primary storage location, the following events occur:

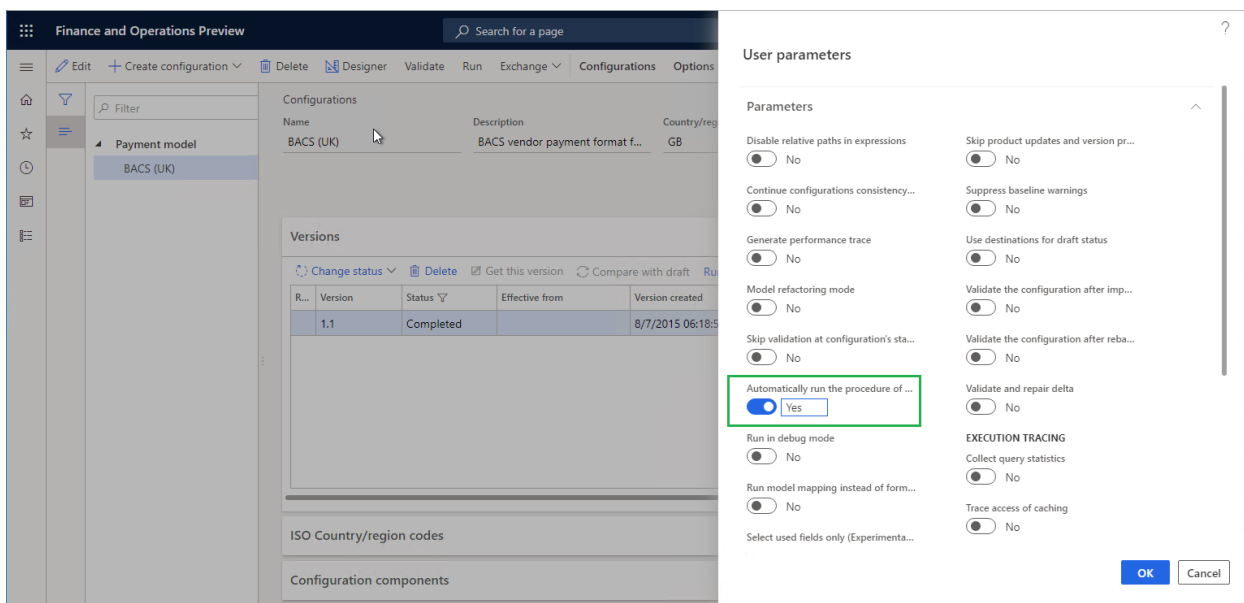
- If the template is available in the backup storage location, it is automatically taken from the backup storage location, restored to the primary storage location, and used for the current execution.
- Every user who is assigned to the **Electronic reporting developer** or **System administrator** role is notified about the missing template issue through the Action center. The message that appears depends on the value of the **Automatically run the procedure of restoring the broken templates in batch** parameter:
  - If this parameter is set to **Off**, the message recommends that you start the batch process to automatically fix similar issues for other ER format configuration templates. The message includes a link that you can use to start the batch process.
  - If this parameter is set to **On**, the message notifies you that a missing templates issue has been discovered, and that a new batch process, **Restore broken templates from internal database backup**, has been automatically scheduled. This batch process will automatically fix similar issues for other templates.

To set up the **Automatically run the procedure of restoring the broken templates in batch** parameter, complete the following steps:

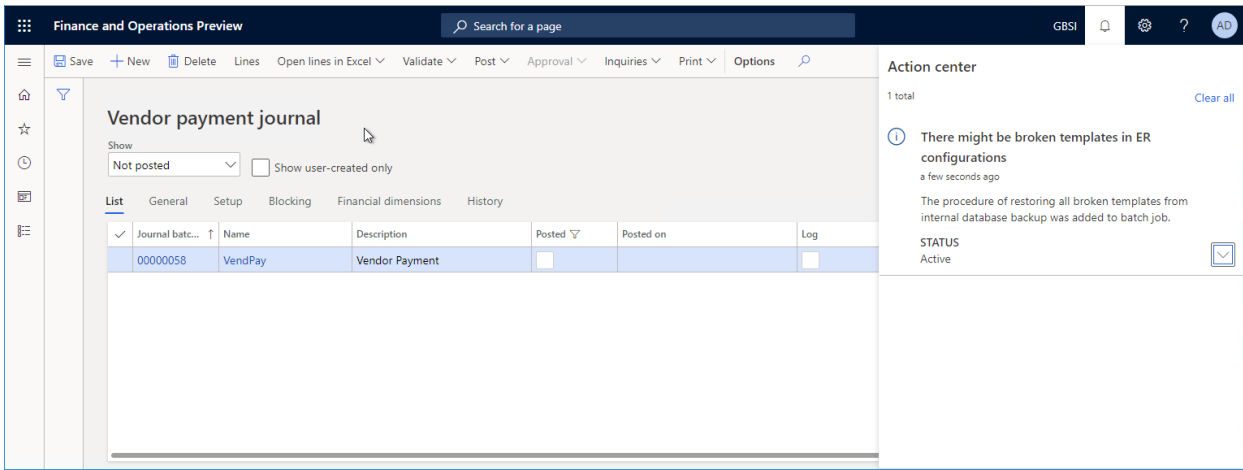
1. In Finance and Operations, open the **Organization administration > Electronic reporting > Configurations** page.
2. On the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
3. In the **User parameters** dialog box, set the required value for the **Automatically run the procedure of restoring the broken templates in batch** parameter.

#### NOTE

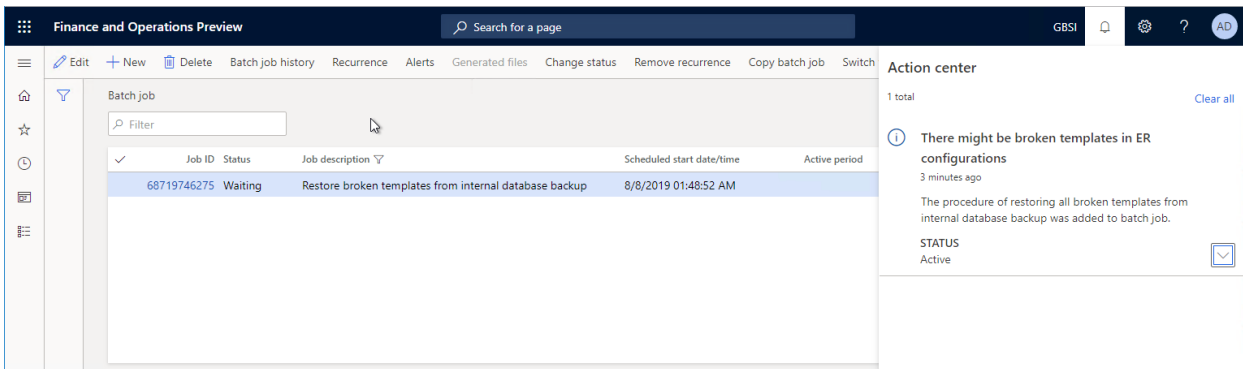
This parameter is defined as application user and logged company specific.



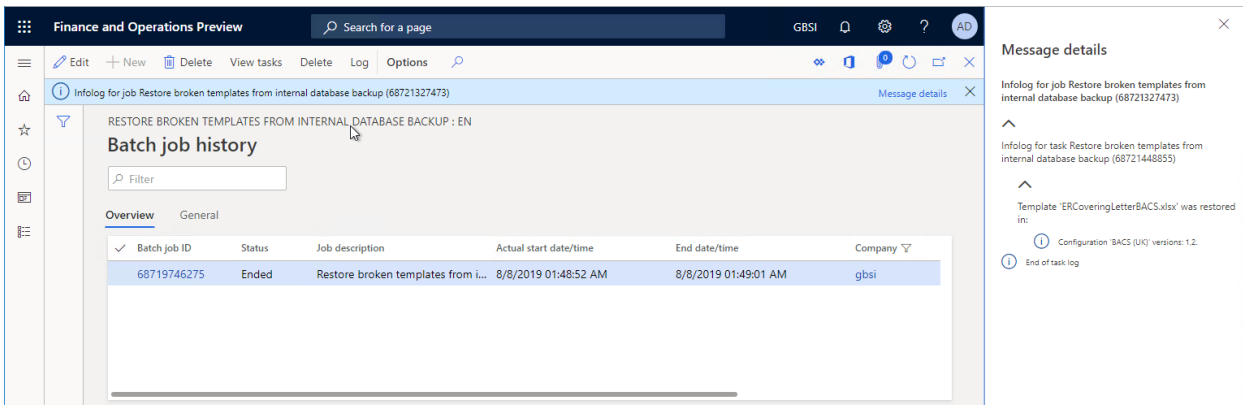
The following illustration shows an example of the message that appears when the **Automatically run the procedure of restoring the broken templates in batch** parameter is set to **On**.



The following illustration shows the **Restore broken templates from internal database backup** batch process on the **Batch job** page.



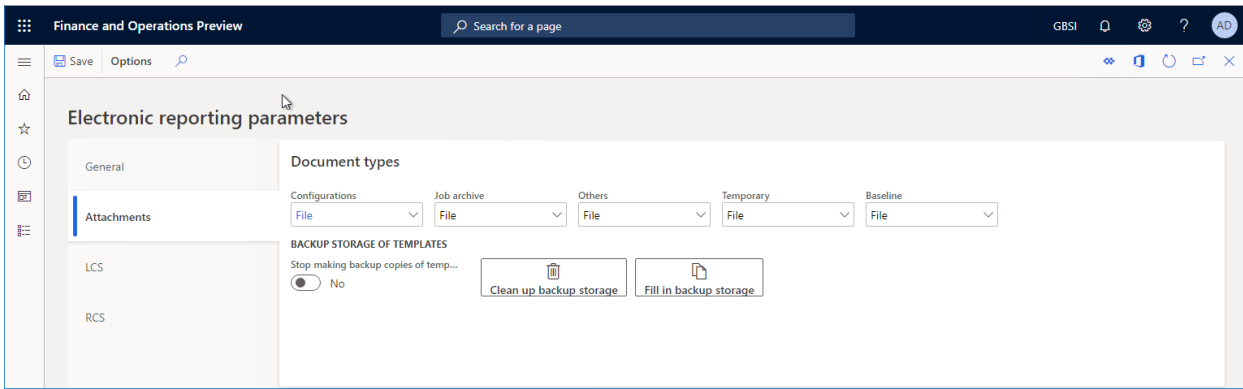
The execution log of the completed **Restore broken templates from internal database backup** batch process includes information about the templates that have been restored from the backup storage location to the primary storage location.



By default, the process of automatically creating backup copies of templates that reside in ER format configurations is turned on. To stop making backup copies of templates, set the **Stop making backup copies of template** option to **Yes** on the **Attachments** tab of the **Electronic reporting parameters** page. You can open this page from the **Electronic reporting** workspace.

If you set the **Stop making backup copies of templates** option to **Yes** and don't want to keep the backup copies that were previously made of templates, select **Clean up backup storage** on the **Electronic reporting parameters** page.

If you upgraded your environment to Finance and Operations version 10.0.5 (October 2019) and want to migrate to a new environment that includes ER format configurations that can be run, select **Fill in backup storage** on the **Electronic reporting parameters** page before the migration occurs. This button starts the process of making backup copies of all available templates, so that they can be stored in the ER backup storage location for templates.



## Manual recovery

Go to **Organization administration > Electronic reporting > Restore broken templates** to manually initiate the process of restoring ER templates from the backup storage location to the primary storage location. Before you start this process, on the **Restore broken templates** page you can specify whether it will be performed interactively, or the batch process will be scheduled for this.

## Supported deployments

In Finance and Operations version 10.0.5, the Backup storage of ER templates feature is available only in cloud deployments.

## Additional resources

[Electronic reporting \(ER\) overview](#)

[Configure the Electronic reporting \(ER\) framework](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create configuration providers and mark them as active

2/18/2021 • 2 minutes to read • [Edit Online](#)

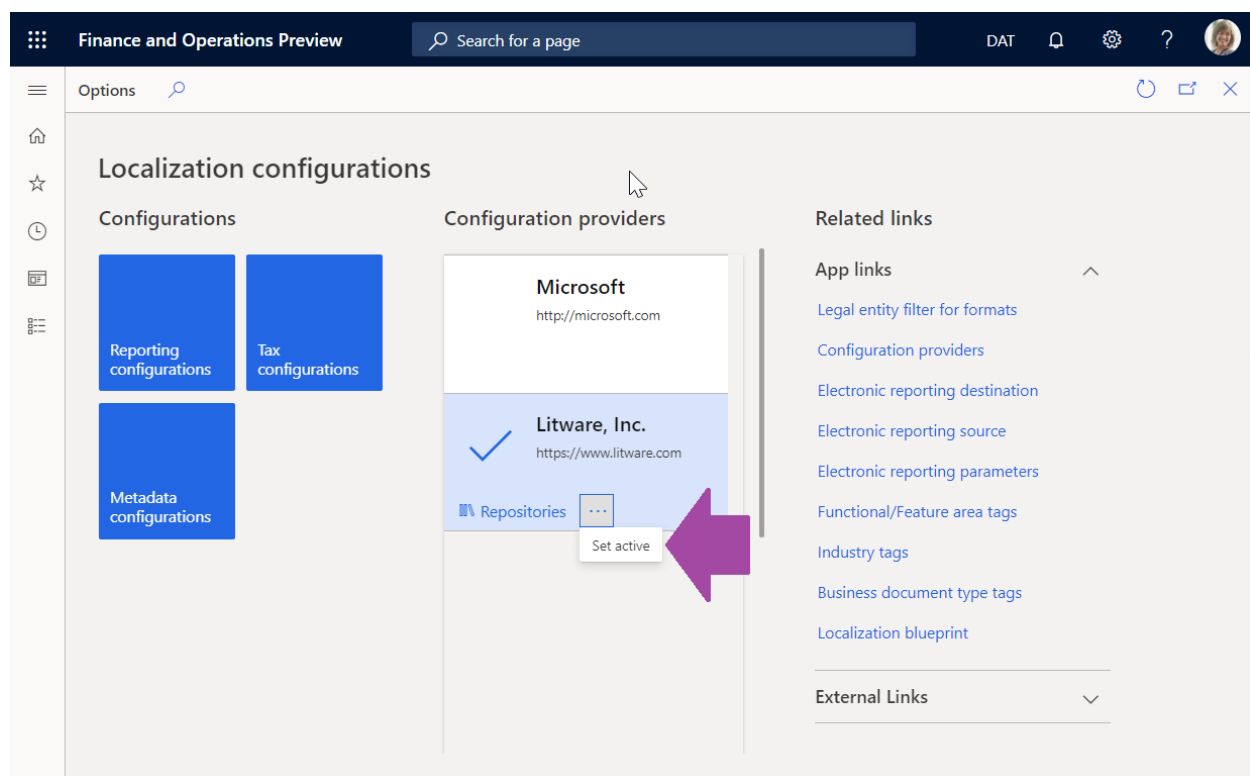
This topic explains how a user assigned to the System Administrator or Electronic Reporting Developer role can create a configuration provider for Electronic reporting (ER). Each ER configuration will refer to the provider as the author of the configuration. In this example, you will create a configuration provider for sample company, Litware, Inc. These steps can be performed in any company as ER configuration providers are shared among all companies.

## Create a provider

1. Go to the **navigation pane** in the upper left corner and select **Organization administration**.
2. Go to **Workspaces > Electronic reporting**.
3. Go to **Related links > Configuration providers**.
4. Select **New**.
  - A provider record has a unique name and URL. Review the content of this page and skip this procedure if a record for Litware, Inc. (<https://www.litware.com>) already exists.
5. In the Name field, type `Litware, Inc.`.
6. In the Internet address field, type `https://www.litware.com`.
7. Select **Save**.
8. Close the page.

## Select as an active provider

1. Select the Litware, Inc. provider.
2. Select **Set active**.



The screenshot shows the Dynamics 365 interface for configuration providers. The top navigation bar includes 'Finance and Operations Preview', a search bar, and user profile icons. The main content area is titled 'Localization configurations' and is divided into three sections: 'Configurations', 'Configuration providers', and 'Related links'. The 'Configuration providers' section lists two providers: 'Microsoft' (http://microsoft.com) and 'Litware, Inc.' (https://www.litware.com). The 'Litware, Inc.' provider is highlighted with a blue checkmark and a 'Set active' button. A purple arrow points to this button. The 'Related links' section on the right contains a list of links such as 'Legal entity filter for formats', 'Configuration providers', 'Electronic reporting destination', 'Electronic reporting source', 'Electronic reporting parameters', 'Functional/Feature area tags', 'Industry tags', 'Business document type tags', and 'Localization blueprint'.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Electronic reporting (ER) destinations

2/18/2021 • 12 minutes to read • [Edit Online](#)

You can configure a destination for each Electronic reporting (ER) format configuration and its output component (a folder or a file). Users who have appropriate access rights can also modify destination settings at runtime. This topic explains ER destination management, the types of destinations that are supported, and security considerations.

ER format configurations usually contain at least one output component: a file. Typically, configurations contain multiple file output components of different types (for example, XML, TXT, XLSX, DOCX, or PDF) that are grouped into either a single folder or multiple folders. ER destination management lets you preconfigure what occurs when each component is run. By default, when a configuration is run, a dialog box appears that lets you save or open the file. The same behavior also occurs when you import an ER configuration and don't configure any specific destinations for it. After a destination is created for a main output component, that destination overrides the default behavior, and the folder or file is sent according to the destination's settings.

## Availability and general prerequisites

The functionality for ER destinations isn't available in Microsoft Dynamics AX 7.0 (February 2016). Therefore, you must install Microsoft Dynamics 365 for Operations version 1611 (November 2016) or later to use the following destination types:

- [Email](#)
- [Archive](#)
- [File](#)
- [Screen](#)
- [Power BI](#)

Alternatively, you can install one of the following prerequisites. However, be aware that these alternatives provide a more limited ER destination experience.

- Microsoft Dynamics AX application version 7.0.1 (May 2016)
- [Electronic reporting destination management application hotfix](#)

There is also a [Print](#) destination type. To use it, you must install Microsoft Dynamics 365 Finance version 10.0.9 (April 2020).

## Overview

You can set up destinations only for ER configurations that have been [imported](#) into the current Finance instance, and for the formats that are available on the [Electronic reporting configurations](#) page. The functionality for ER destination management is available at [Organization administration > Electronic reporting > Electronic reporting destination](#).

### Default behavior

The default behavior for an ER format configuration depends on the execution type that you specify when an ER format starts.

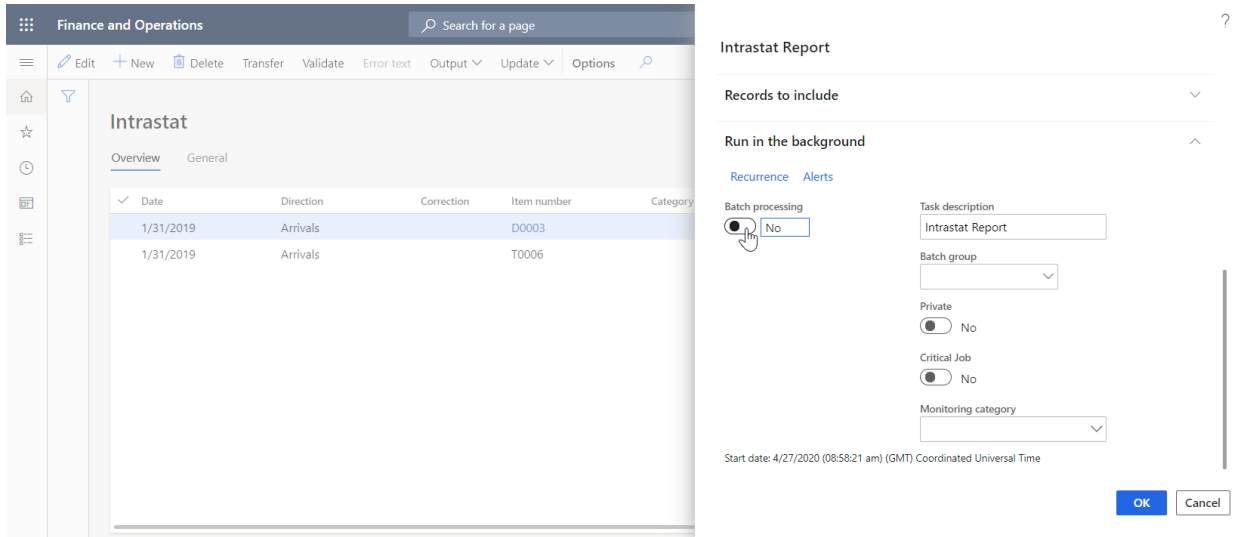
In the [Intrastat Report](#) dialog box, on the [Run in the background](#) FastTab, if you set the [Batch processing](#) option to **No**, an ER format is run immediately in interactive mode. When this execution is successfully completed, a generated outbound document is made available for download.



If you set the **Batch processing** option to **Yes**, an ER format is run in **batch** mode. The appropriate batch job is created, based on the parameters that you specify on the **Run in the background** tab of the ER parameters dialog box.

**NOTE**

The job description informs you about the run of an ER format mapping. It also contains the name of the ER component that is run.



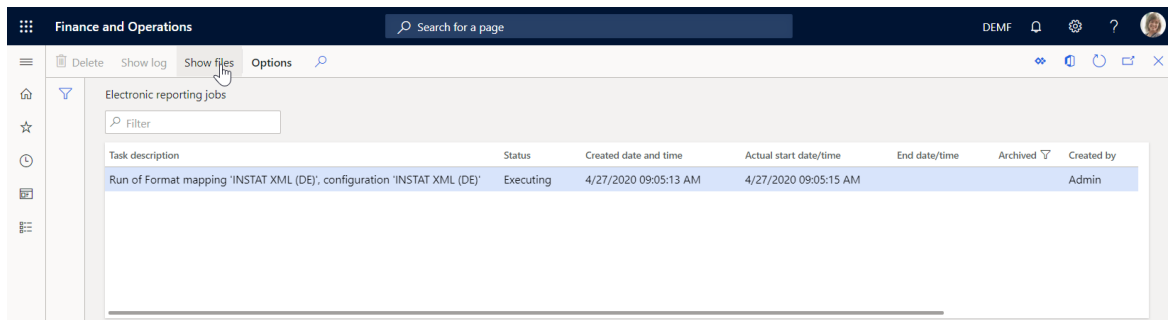
You can find information about this job in several places:

- Go to **Common > Inquiries > Batch jobs > My batch jobs** to check the status of the scheduled job.
- Go to **Organization administration > Electronic reporting > Electronic reporting jobs** to check the status of the scheduled job and the execution results of the completed job. When job execution is successfully completed, select **Show files** on the **Electronic reporting jobs** page to get a generated outbound document.

**NOTE**

This document is stored as an attachment of the current job record and is controlled by the **Document management** framework. The **document type** that is used to store ER artifacts of this type is configured in the **ER parameters**.

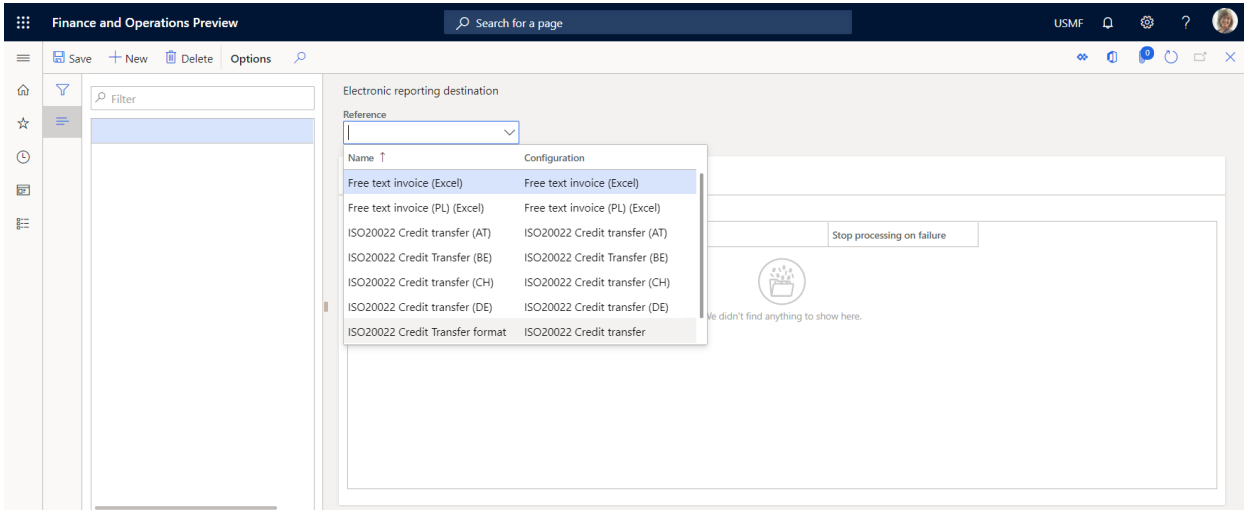
- On the **Electronic reporting jobs** page, select **Show files** to view the list of any errors and warnings that were generated during job execution.



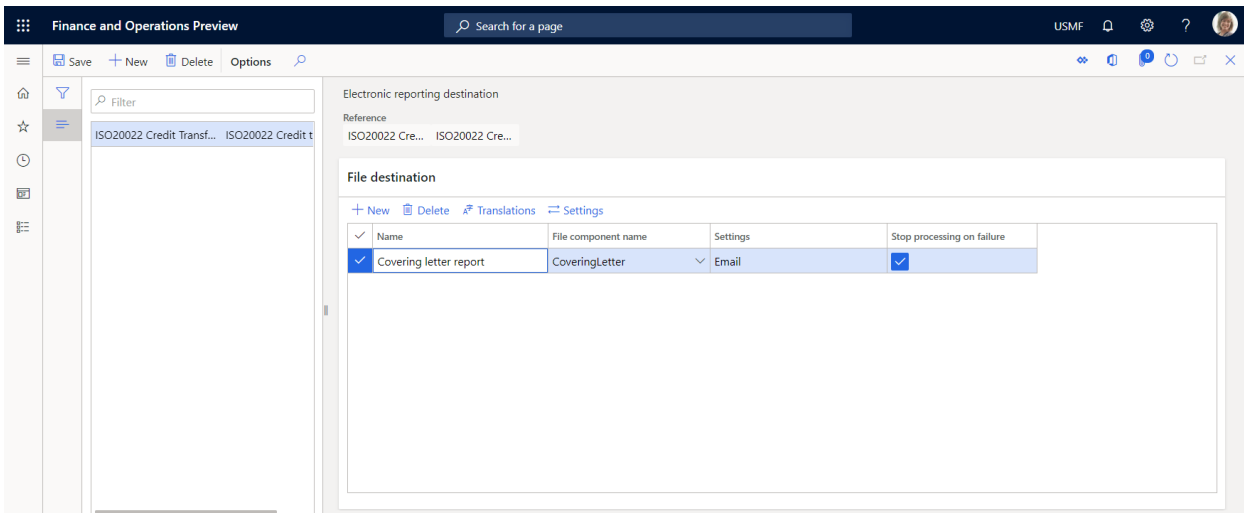
**User-configured behavior**

On the **Electronic reporting destination** page, you can override the default behavior for a configuration. Imported configurations aren't shown on this page until you select **New** and then, in the **Reference** field, select

a configuration to create destination settings for.



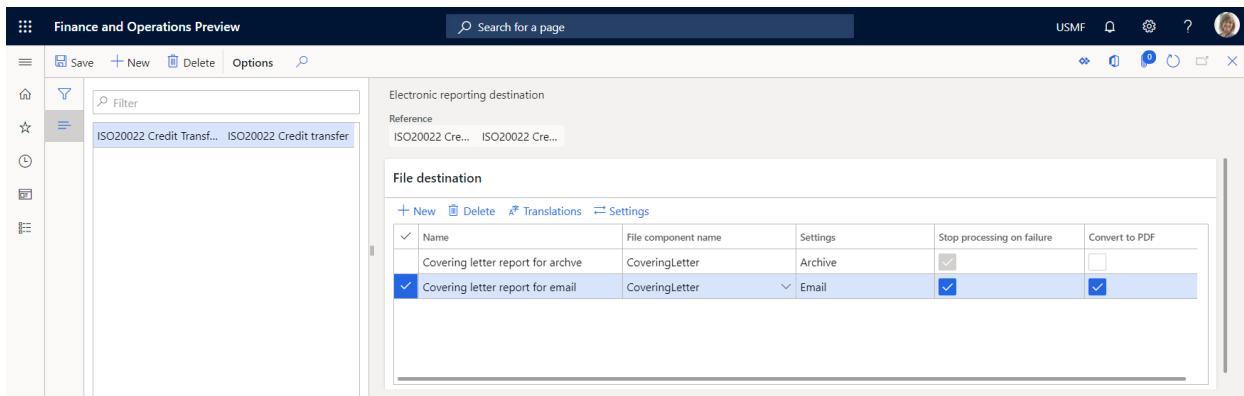
After you create a reference, you can create a file destination for each **Folder** or **File** output component of the referenced ER format.



Next, in the **Destination settings** dialog box, you can enable and disable individual destinations for the file destination. The **Settings** button is used to control all the destinations for a selected file destination. In the **Destination settings** dialog box, you can control each destination separately by setting the **Enabled** option for it.

In versions of Finance **before version 10.0.9**, you can create **one file destination** for each output component of the same format, such as a folder or a file that is selected in the **File Name** field. However, in **version 10.0.9 and later**, you can create **multiple file destinations** for each output component of the same format.

For example, you can use this capability to configure file destinations for a file component that is used to generate an outbound document in Excel format. One destination (**Archive**) can be configured to store the original Excel file in the ER jobs archive, and another destination (**Email**) can be configured to simultaneously **convert** the Excel file to PDF format and send the PDF file by email.



When you run an ER format, all destinations that were configured for components of the format are always run. In addition, in Finance **version 10.0.17 and later**, the ER destinations functionality has been improved and now lets you configure different sets of destinations for a single ER format. This configuration marks each set as configured for a particular user action. The ER API has been [extended](#) so that an action can be provided that the user performs by running an ER format. The action code that is provided is passed to ER destinations. You can run different destinations of an ER format, depending on the action code that is provided. For more information, see [Configure action-dependent ER destinations](#).

## Destination types

The following destinations are currently supported for ER formats. You can disable or enable all types at the same time. In this way, you can either do nothing or send the component to all configured destinations.

- [Email](#)
- [Archive](#)
- [File](#)
- [Screen](#)
- [Power BI](#)
- [Print](#)

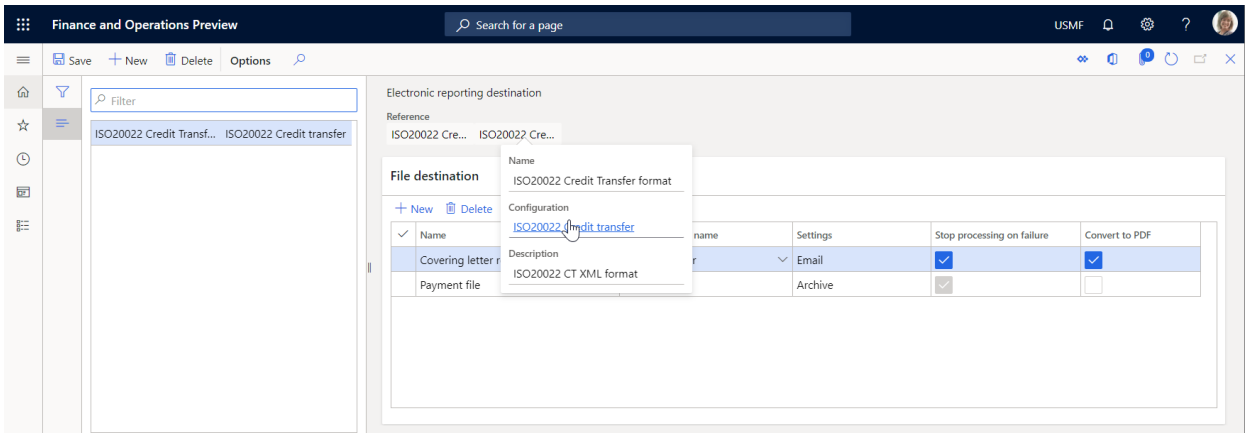
## Applicability

You can set up destinations only for ER configurations that have been imported, and for the formats that are available on the [Electronic reporting configurations](#) page.

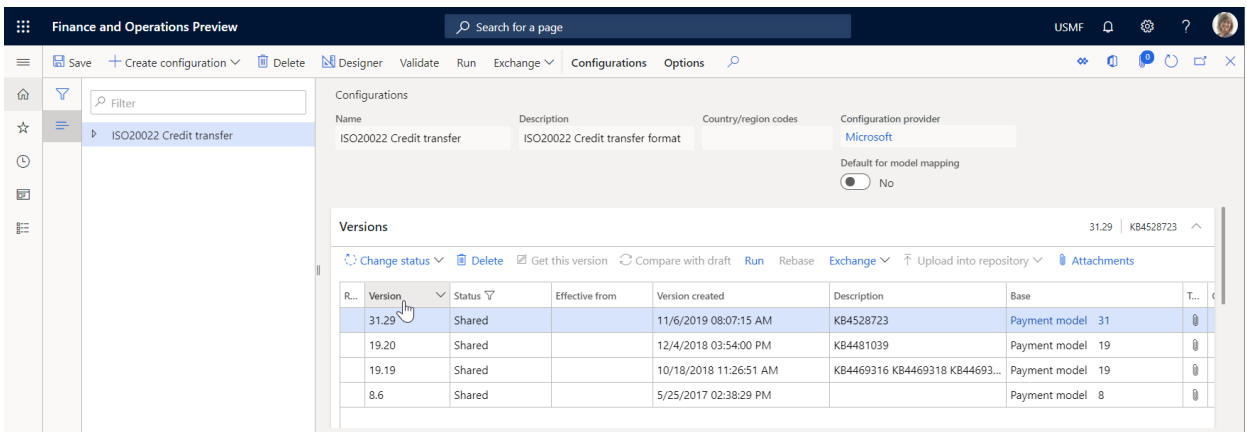
### NOTE

Configured destinations are company-specific. If you plan to use an ER format in different companies of the current Finance instance, you must configure destinations for that ER format for each of those companies.

When you configure file destinations for a selected format, you configure them for the whole format.

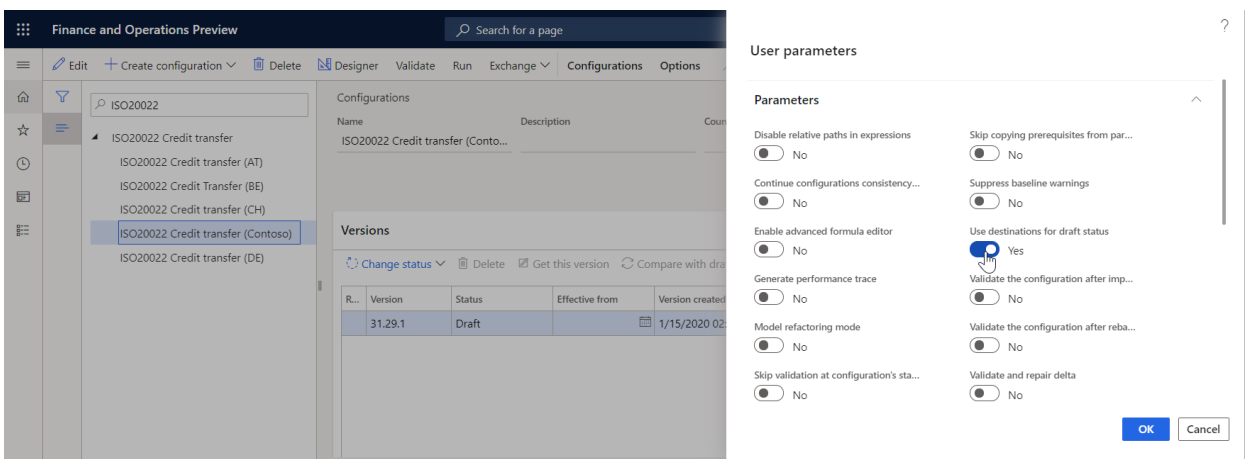


At the same time, you might have multiple [versions](#) of the format that have been imported into the current Finance instance. You can view them if you select the **Configuration** link that is offered when you select the **Reference** field.



By default, configured destinations are applied only when you run an ER format version that has a status of either **Completed** or **Shared**. However, you must sometimes use configured destinations when the draft version of an ER format is run. For example, you modify a draft version of your format, and you want to use configured destinations to test how generated output will be delivered. Follow these steps to apply destinations for an ER format when the draft version is run.

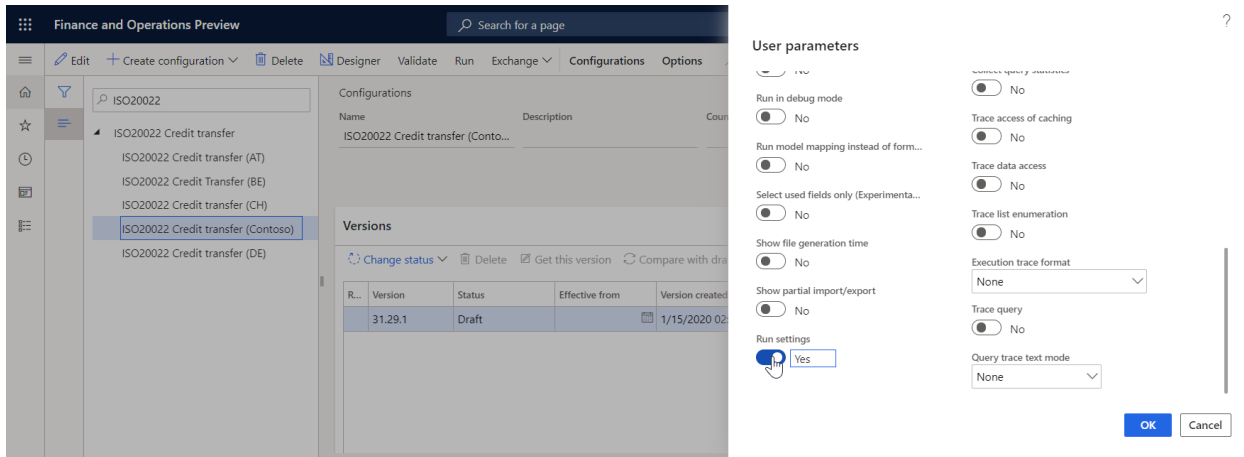
1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
3. Set the **Use destinations for draft status** option to **Yes**.



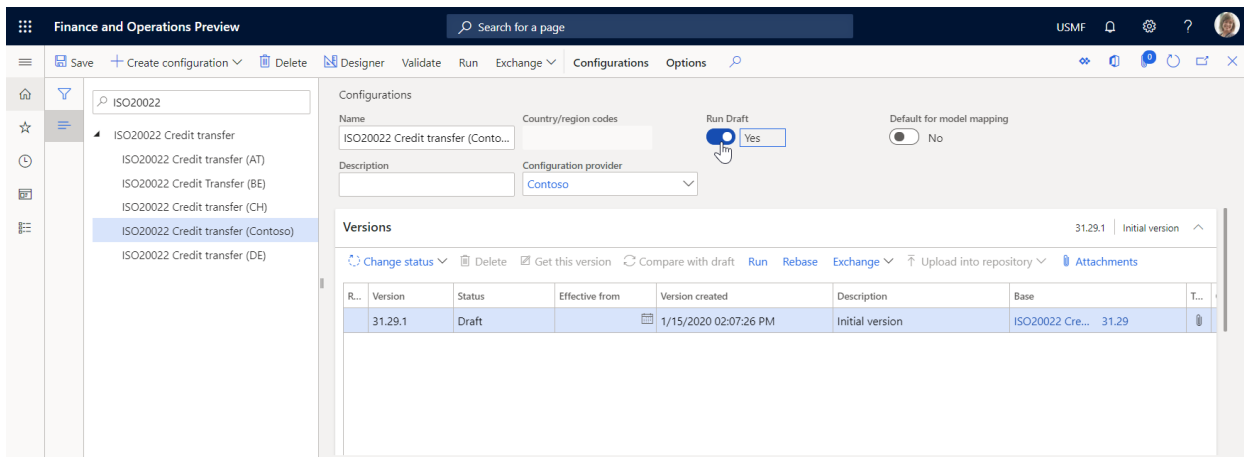
To use the draft version of an ER format, you must mark the ER format accordingly.

1. Go to **Organization administration > Electronic reporting > Configurations**.

- On the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
- Set the **Run setting** option to **Yes**.



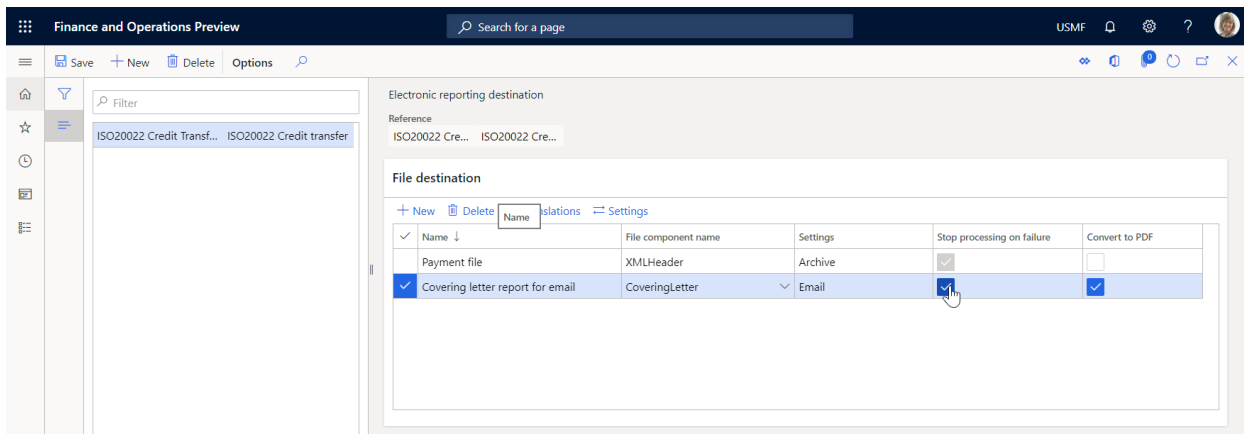
After you complete this setup, the **Run draft** option becomes available for ER formats that you modify. Set this option to **Yes** to start to use the draft version of the format when the format is run.



## Destination failure handling

Usually, an ER format is run within the scope of a specific business process. However, the delivery of an outbound document that is generated during execution of an ER format must sometimes be considered part of that business process. In this case, if delivery of a generated outbound document to a configured destination is unsuccessful, execution of the business process must be canceled. To configure the appropriate ER destination, select the **Stop processing on failure** option.

For example, you configure vendor payment processing so that the **ISO20022 Credit Transfer** ER format is run to generate the payment file and supplementary documents (for example, the covering letter and control report). If a payment should be considered successfully processed only if the covering letter is successfully delivered by email, you must select the **Stop processing on failure** check box for the **CoveringLetter** component in the appropriate file destination, as shown in the following illustration. In this case, the status of the payment that is selected for processing will be changed from **None** to **Sent** only when the covering letter that is generated is successfully accepted for delivery by an email provider that is configured in the Finance instance.



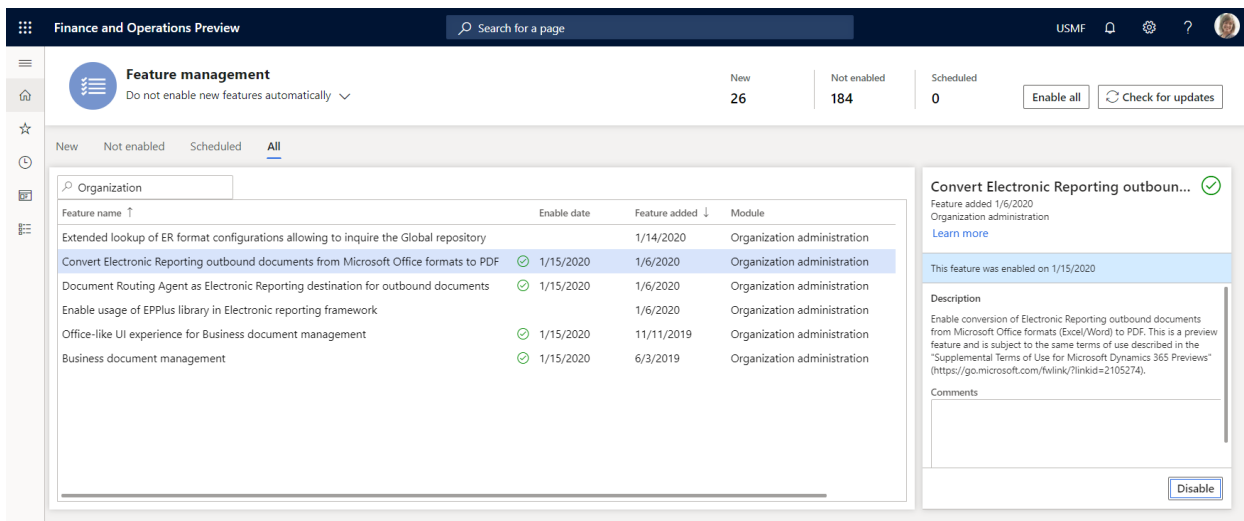
If you clear the **Stop processing on failure** check box for the **CoveringLetter** component in the destination, a payment will be considered successfully processed even if the covering letter isn't successfully delivered by email. The status of the payment will be changed from **None** to **Sent** even if the covering letter can't be sent because, for example, the email address of the recipient or sender is missing or incorrect.

## Output conversion to PDF

You can use the PDF conversion option to convert output in Microsoft Office (Excel or Word) format to PDF format.

### Make PDF conversion available

To make the PDF conversion option available in the current Finance instance, open the **Feature management** workspace, and turn on the **Convert Electronic Reporting outbound documents from Microsoft Office formats to PDF** feature.



### Applicability

The PDF conversion option can be turned on only for file components that are used to generate output in Office (Excel or Word) format (**Excel file**). When this option is turned on, output that is generated in Office format is automatically converted to PDF format.

### Limitations

The PDF conversion option is available only for cloud deployments.

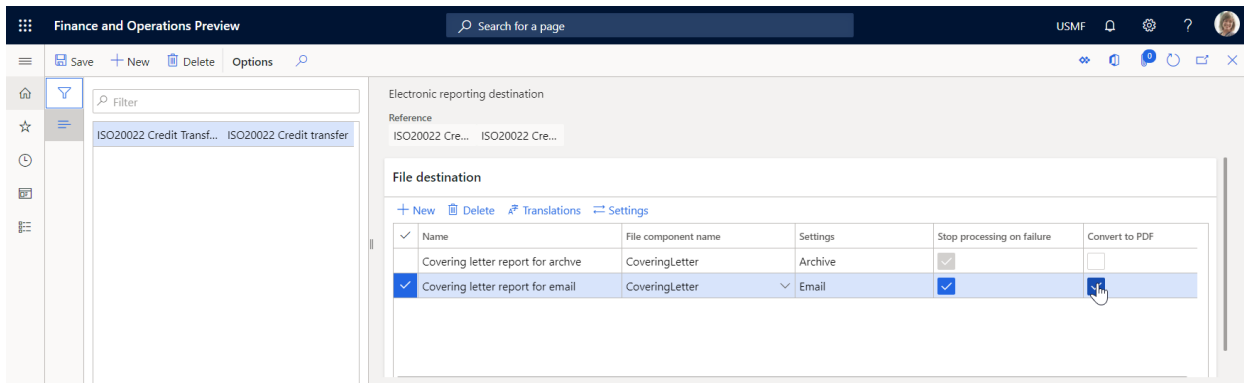
The PDF document that is produced is limited to a maximum length of 300 pages.

In Finance **version 10.0.9**, only landscape page orientation is supported in the PDF document that is produced from Excel output. In Finance **version 10.0.10 (May 2020) and later**, you can [specify the page orientation](#) of the PDF document that is produced from Excel output while you configure an ER destination.

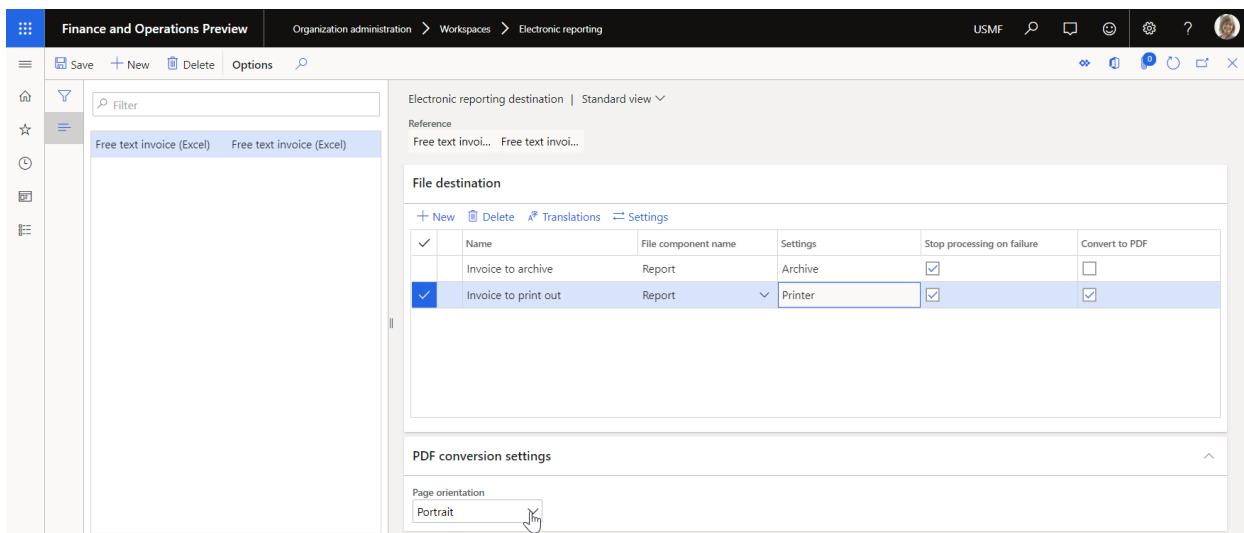
Only the common system fonts of the Windows operating system are used to convert output that contains no embedded fonts.

## Use the PDF conversion option

To turn on PDF conversion for a file destination, select the **Convert to PDF** check box.



If you generate an ER configuration in Excel format and want to convert it to PDF format, you can specify the page orientation of the PDF document. When you select the **Convert to PDF** check box to turn on PDF conversion for a file destination that produces an output file in Excel format, the **Page orientation** field becomes available on the **PDF conversion settings** FastTab. In the **Page orientation** field, select the preferred orientation.



### NOTE

To have the option to select the PDF page orientation, you must install Finance version 10.0.10 or later.

The selected page orientation is applied to all ER configurations that are generated in Excel format and then converted to PDF format.

If an ER configuration in Word format is converted to PDF format, the page orientation of the PDF document is taken from the Word document.

## Security considerations

Two types of privileges and duties are used for ER destinations. One type controls a user's overall ability to maintain the destinations that are configured for a legal entity (that is, it controls access to the **Electronic reporting destinations** page). The other type controls an application user's ability to override, at runtime, the destination settings that an ER developer or ER functional consultant has configured.

ROLE (AOT NAME)	ROLE NAME	DUTY (AOT NAME)	DUTY NAME
ERDeveloper	Electronic reporting developer	ERFormatDestinationConfigure	Configure electronic reporting format destination
ERFunctionalConsultant	Electronic reporting functional consultant	ERFormatDestinationConfigure	Configure electronic reporting format destination
PaymAccountsPayablePaymentsClerk	Accounts payable payments clerk	ERFormatDestinationRuntimeConfigure	Configure electronic reporting format destination during runtime
PaymAccountsReceivablePaymentsClerk	Accounts receivable payments clerk	ERFormatDestinationRuntimeConfigure	Configure electronic reporting format destination during runtime

#### NOTE

Two privileges are used in the preceding duties. These privileges have the same names as the corresponding duties: ERFormatDestinationConfigure and ERFormatDestinationRuntimeConfigure.

## Frequently asked questions

**I have imported electronic configurations, and I see them on the Electronic reporting configurations page. But why don't I see them on the Electronic reporting destinations page?**

Make sure that you select **New** and then select a configuration in the **Reference** field. The **Electronic reporting destinations** page shows only configurations that destinations have been configured for.

**Is there any way to define which Microsoft Azure Storage account and Azure Blob storage are used?**

No. The default Microsoft Azure Blob storage that is defined and used for the document management system is used.

**What is the purpose of the File destination in the destination settings? What does that setting do?**

The **File** destination is used to control a dialog box of your web browser when you run an ER format in interactive mode. If you enable this destination, or if no destination is defined for a configuration, an open or save dialog box appears in your web browser after an output file is created.

**Can you give an example of the formula that refers to a vendor account that I can send email to?**

The formula is specific to the ER configuration. For example, if you use the ISO 20022 Credit Transfer configuration, you can use '\$PaymentsForCoveringLetter'.Creditor.Identification.SourceID or model.Payments.Creditor.Identification.SourceID to get an associated vendor account.

**One of my format configurations contains multiple files that are grouped into one folder (for example, Folder1 contains File1, File2, and File3). How do I set up destinations so that Folder1.zip isn't created at all, File1 is sent by email, File2 is sent to SharePoint, and I can open File3 immediately after the configuration is run?**

Your format must first be available in the ER configurations. If this prerequisite is met, open the **Electronic reporting destination** page, and create a new reference to the configuration. You must then have four file destinations, one for each output component. Create the first file destination, give it a name such as **Folder**, and select a file name that represents a folder in your configuration. Then select **Settings**, and make sure that all the destinations are disabled. For this file destination, the folder won't be created. By default, because of hierarchical dependencies between files and parent folders, the files will behave in the same way. In other words, they won't



be sent anywhere. To override that default behavior, you must create three more file destinations, one for each file. In the destination settings for each, you must enable the destination that the file should be sent to.

## Additional resources

[Electronic reporting \(ER\) overview](#)

[Configure action-dependent ER destinations](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

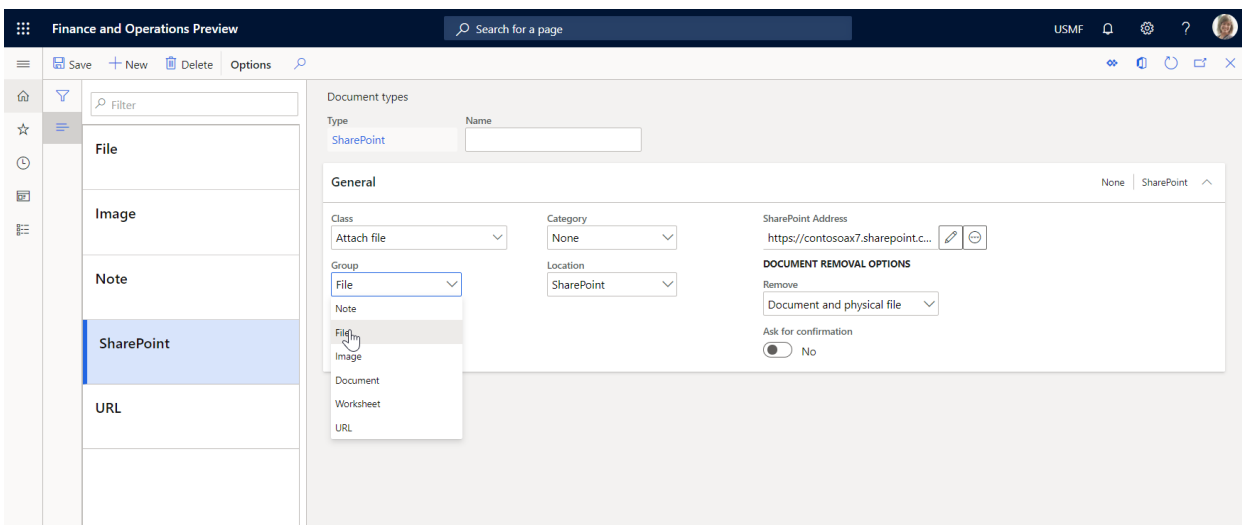
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Archive ER destination type

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can configure an archive destination for each **Folder** or **File** component of an Electronic reporting (ER) format that is configured to generate outbound documents. Based on the destination setting, a generated document is stored as an attachment of a record of the ER jobs list. To view the results, go to **Organization administration > Electronic reporting > Electronic reporting jobs**.

You can use this option to send the generated document to a Microsoft SharePoint folder or Microsoft Azure Storage. Set **Enabled** to **Yes** to send output to a destination that is defined by the selected document type. Only document types where the group is set to **File** are available for selection. You define document **types** at **Organization administration > Document management > Document types**. The configuration for ER destinations is the same as the configuration for the document management system.



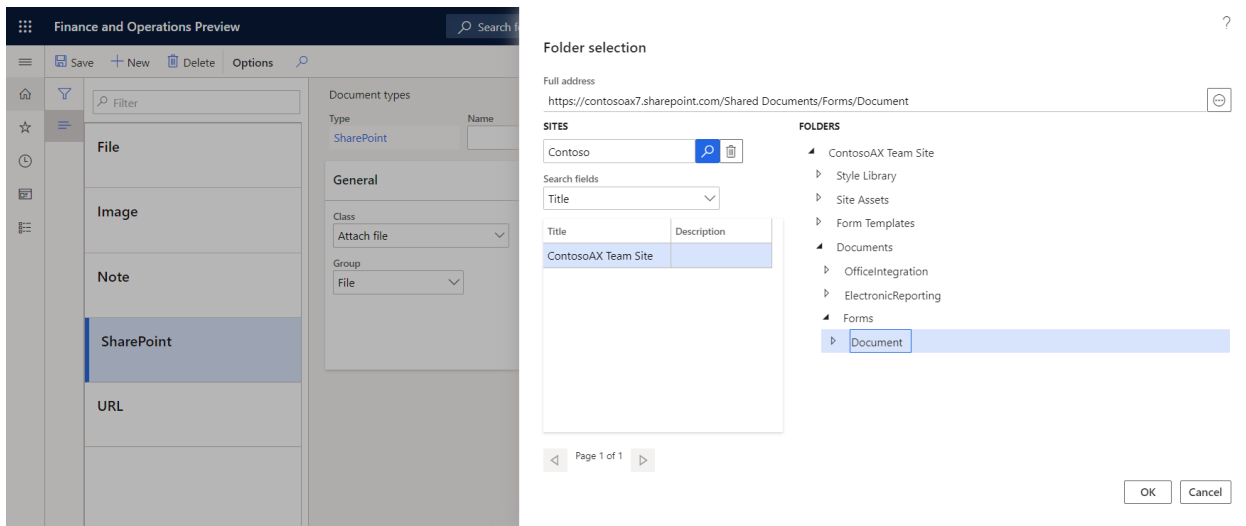
The location determines where the file is saved. After the **Archive** destination is enabled, the results can be saved in the Job archive. You can view the results at **Organization administration > Electronic reporting > Electronic reporting archived jobs**.

## NOTE

Select a document type for the Job archive by navigating to **Organization administration > Workspaces > Electronic reporting > Electronic reporting parameters**. For more information, see [Configure the Electronic reporting \(ER\) framework](#).

## SharePoint

You can save a file in a designated SharePoint folder. To define the default SharePoint server, go to **Organization administration > Document management > Document management parameters**. On the **SharePoint** tab, configure the SharePoint folder. Then, you can select it as the folder where the ER output will be saved. The **SharePoint** location must be selected in this document type.



## Azure Storage

When the document type location is set to **Azure storage**, you can save a file to Azure Storage.

### NOTE

The ER framework permanently stores files in Azure Blob storage unlike the Data management framework that applies the seven-day retention policy for documents that must be processed. For more information, see [API for getting message status](#) and [Status check API](#). The ER-related files will be stored in Azure Blob storage as attachments of application table records as long as necessary. A single file will be deleted from Azure Blob storage along with the application table record that this file was attached to.

## Additional resources

- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) destinations](#)
- [Configure document management](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Email ER destination type

2/18/2021 • 7 minutes to read • [Edit Online](#)

When an Electronic reporting (ER) format is run, one or more outbound documents can be generated. **Folder** or **File** format components are used in ER formats to specify the structure of outbound documents. You can configure an email destination for these types of components to send outbound documents as email attachments.

You can configure an email destination for each **Folder** or **File** component of an ER format. In this case, **each outbound document is emailed individually**. Based on this destination setting, a generated document is delivered as an attachment of an email.

## NOTE

If no document is generated, because the **Enabled** expression for the relevant **File** component has been configured to return a **False** Boolean value, no email is sent, even if an email destination is configured and enabled for the component.

You can also **group** several **Folder** or **File** components together, and then configure an email destination for all the components in the group. In this case, all outbound documents that are generated by components that belong to the group **are sent as multiple attachments of a single email**. Based on this destination setting, each generated document is delivered as an attachment of a single email.

## NOTE

If at least one document is generated by a **File** component in a group of components, an email is sent. If no document is generated by grouped components, because the **Enabled** expression for each **File** component has been configured to return a **False** Boolean value, no email is sent, even if an email destination is configured and enabled for that group of components.

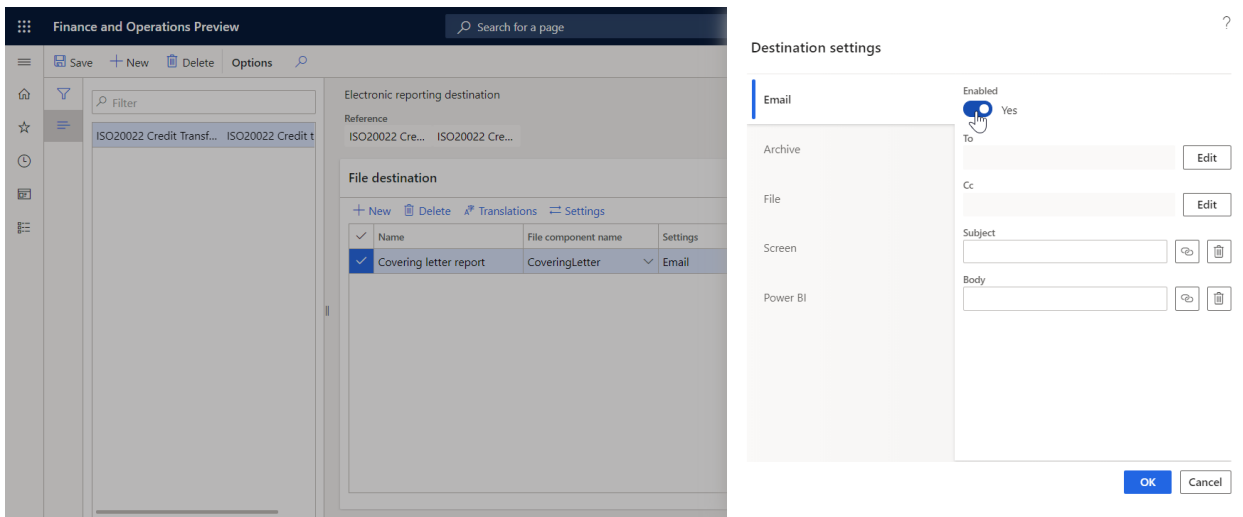
**Email** is the only destination that can be configured for a group of components. To deliver a document that is emailed based on the email destination setting for a group, add one more destination record, select the component that you want, and then configure another destination for this record.

Multiple groups of components can be configured for a single ER format configuration. In this way, you can configure an email destination for every group of components and an email destination for every component.

## Configure an email destination

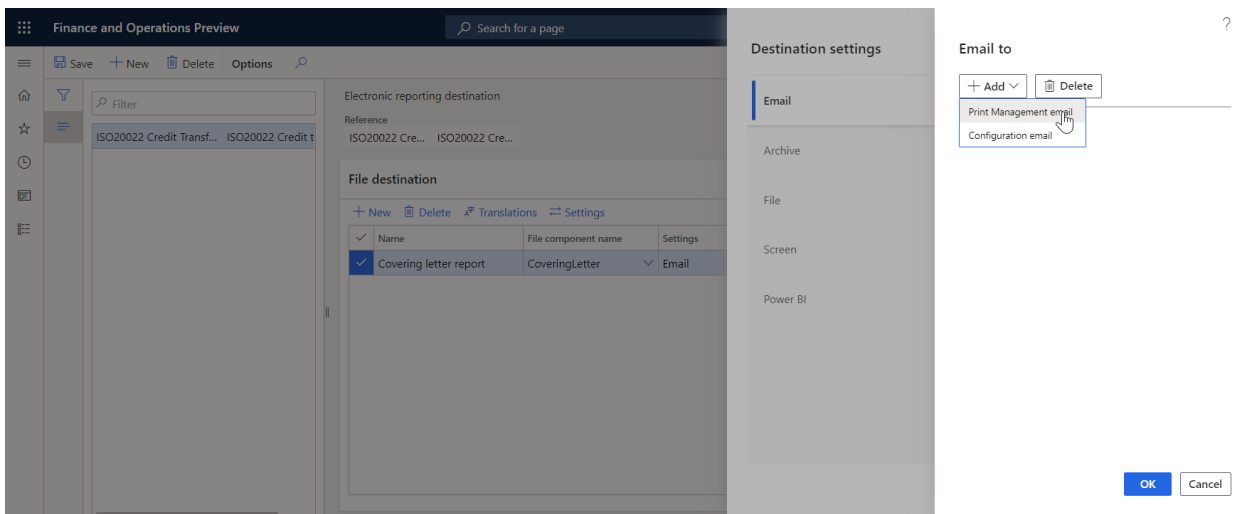
To send an output file or several output files by email, on the **Electronic reporting destination** page, on the **File destination** FastTab, select a component or group of components in the grid, and then select **Settings**. In the **Destination settings** dialog box that appears, on the **Email** tab, set the **Enabled** option to **Yes**. You can then specify email recipients, and edit the subject and body of the email message. You can either set up constant text for the email subject and body, or use ER [formulas](#) to dynamically create email texts.

You can configure email addresses for ER in two ways. The configuration can be completed in the same way that the Print Management feature completes it, or you can resolve an email address by using a direct reference to the ER configuration through a formula.



## Email address types

If you select **Edit** next to the **To** or **Cc** field in the **Destination settings** dialog box, the **Email to** dialog box appears. Select **Add**, and then select the type of email address to use. Two types are currently supported: **Print Management email** and **Configuration email**.



### Print Management email

If you select **Print Management email** as the email address type, you can enter fixed email addresses in the **Email to** dialog box by setting the following fields:

- In the **Email source** field, select **None**.
- In the **Additional email addresses, separated by ";"** field, enter the fixed email addresses.

Alternatively, you can obtain email addresses from the contact details of the party that you generate an outbound document for. To use email addresses that aren't fixed, in the **Email source** field, select the **role** of the party for a file destination. The following roles are supported:

- Customer
- Vendor
- Prospect
- Contact
- Competitor
- Worker
- Applicant
- Prospective vendor

- Disallowed vendor

For example, to configure an email destination for an ER format that is used to process vendor payments, select the **Vendor** role.

After you select the desired role, select the **Bind** button (chain symbol) next to the **Email source account** field to open the [Formula designer](#) page. You can then use this page to configure a formula that returns, at runtime, the account number of the party that is assigned to the configured role from the processed document to the email destination.

#### NOTE

Formulas are specific to the ER configuration.

On the **Formula designer** page, in the **Formula** field, enter a document-specific reference to a supported role. Instead of typing the reference, in the **Data source** pane, find and select the data source node that represents an account of the configured role, and then select **Add data source** to update the formula. For example, if you configure the email destination for the **ISO 20022 Credit Transfer** configuration that is used to process vendor payments, the node that represents a vendor account is

```
'$PaymentsForCoveringLetter'.Creditor.Identification.SourceID
```

The screenshot shows the 'Destination settings' dialog box in Microsoft Dynamics 365 Finance and Operations. The 'Email to' section is selected, and the 'Vendor' role is chosen. The 'Email source account' field is empty, and the 'Company of email source' field is also empty. The 'To' field is empty with an 'Edit' button next to it. The 'OK' and 'Cancel' buttons are at the bottom right.

If the account numbers of the configured role are unique for the whole instance of Microsoft Dynamics 365 Finance, the **Company of email source** field in the **Email to** dialog box can remain blank.

Alternatively, you might have a situation where different parties in the [Global address book](#) have been registered in different companies ([legal entities](#)) in such a way that they all use the same account number to fill the configured role. In this case, account numbers for the configured role aren't unique for the whole Finance instance. Therefore, to explicitly select a party, you can't specify only an account number. You must also specify the company that the party has been registered in the scope of to fill the configured role. Select the **Bind** button (chain symbol) next to the **Company of email source** field in the **Email to** dialog box to open the [Formula designer](#) page. You can then use this page to configure a formula that returns, at runtime, the code of the company that the desired source must be found in the scope of.

### TIP

If you must use the company code to run an ER format, but the ER format doesn't provide any data source that the company code can be obtained from, configure the `GetCurrentCompany()` formula by using the built-in `GETCURRENTCOMPANY` ER function.

### NOTE

Formulas are specific to the ER configuration.

To specify the type of email addresses that must be used at runtime, in the **Email to** dialog box, select **Edit** next to the **To** field to open the **Assign email address** drop-down dialog box. Then set the following fields:

- In the **Purpose** field, select the desired purposes. Only email addresses of the selected purposes from contacts of the discovered party will be used.
- Set the **Primary contact** option to **Yes** to use an email address that is configured for the discovered party as the primary email address.

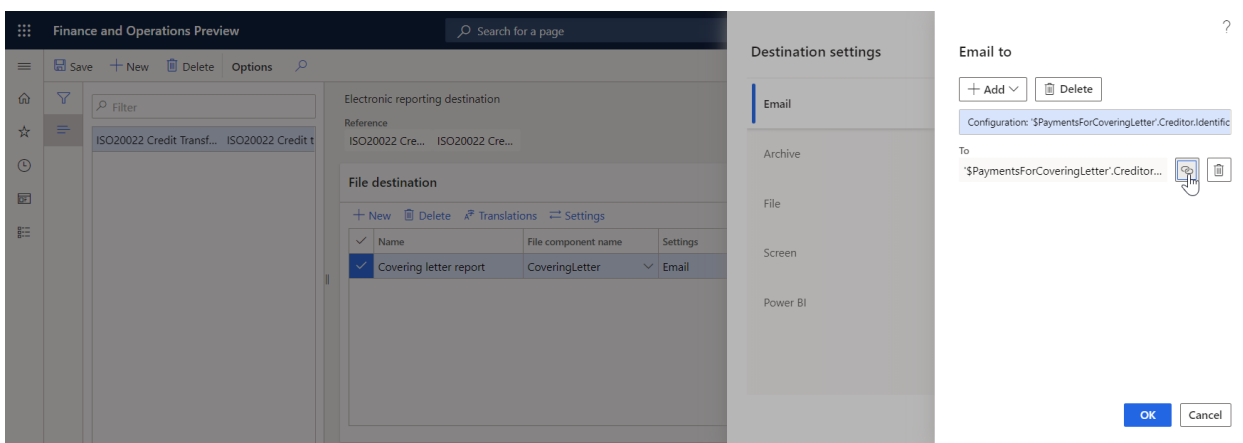
### NOTE

If purposes are selected in the **Purpose** field and the **Primary contact** option is set to **Yes** at the same time, every email that satisfies at least one configured criterion will be used at runtime.

## Configuration email

Select **Configuration email** as the email address type if the configuration that you use has a node in the data sources that returns either a single email address or multiple email addresses that are separated by semicolons (;). You can use [data sources](#) and [functions](#) in the formula designer to get a correctly formatted email address or correctly formatted email addresses that are separated by semicolons. For example, if you use the **ISO 20022 Credit Transfer** configuration, the node that represents the primary email address of a vendor from the vendor contact details that the covering letter should be sent to is

```
'$PaymentsForCoveringLetter'.Creditor.ContactDetails.Email
```



## Group format components

To group format components, on the **Electronic reporting destination** page, on the **File destination** FastTab, select the components in the grid, and then select **Group**.

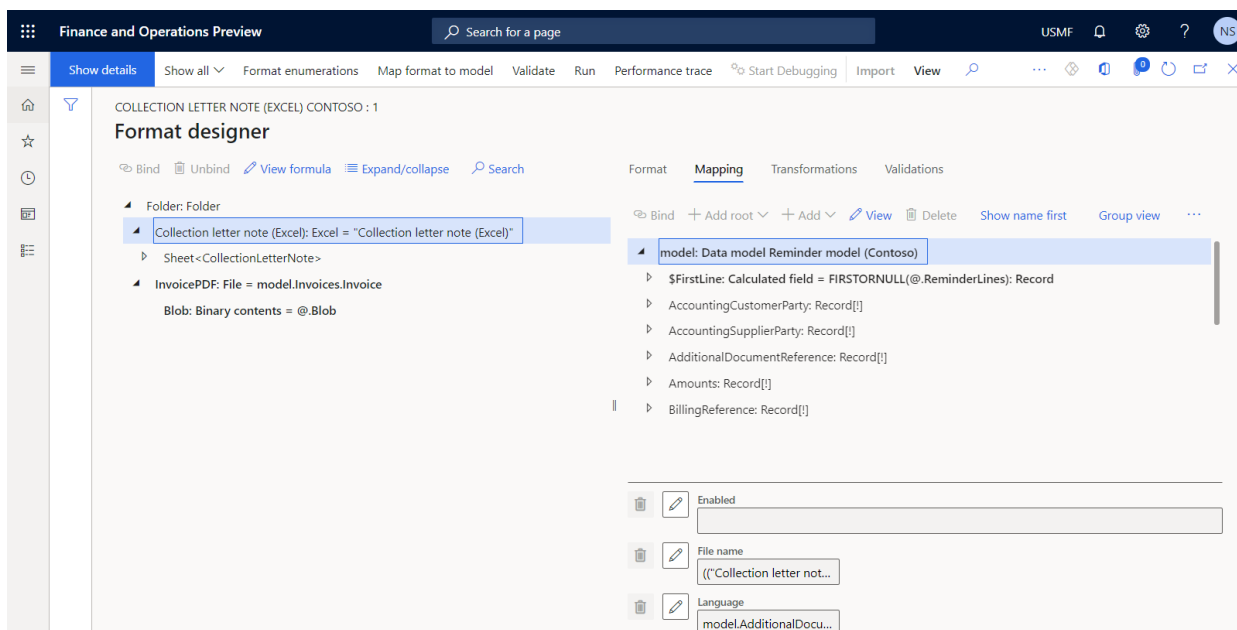
**Email** is the only previously configured destination that is still available for the selected components. No other previously configured destinations are available, because they are considered unsupported for a group of components. You will be notified about these changes as appropriate.

The record that you previously added is considered the header of the group that is created. This header record holds the email destination settings for the group. Other records are group members that will use the email destination settings of the group's header record.

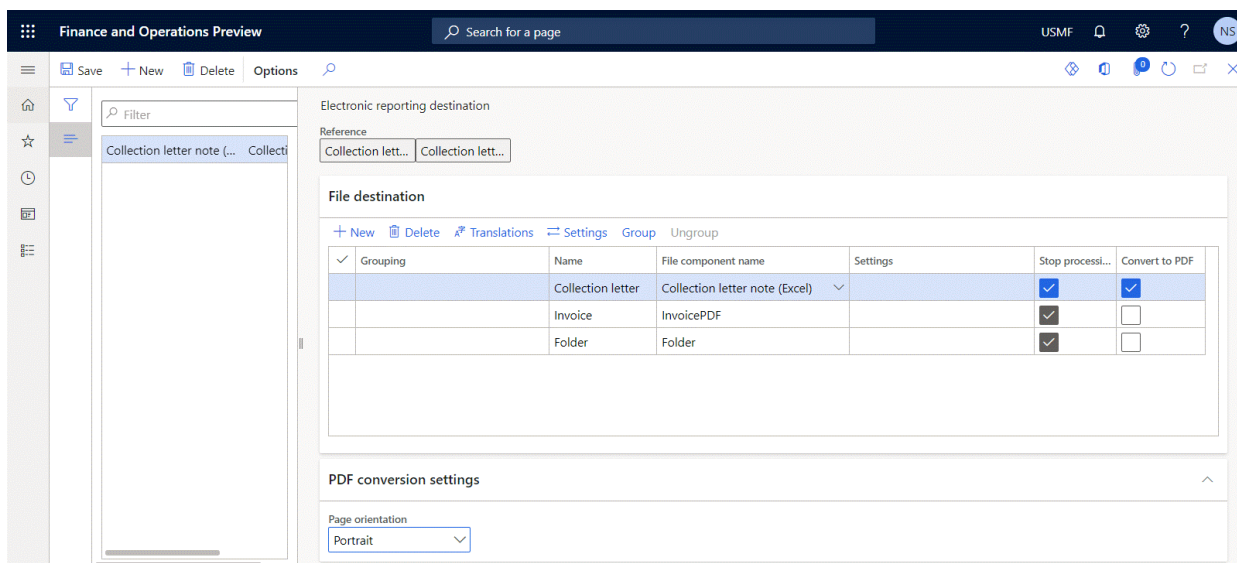
To ungroup format components, on the **File destination** FastTab, select a record that belongs to the group, and then select **Ungroup**.

- If you select a header record, the whole group will be ungrouped.
- If you select a member record, and it's the last member record in a group, the whole group will be ungrouped.
- If you select a member record that isn't the last member record in a group, that record will be excluded from the current group.

The following illustration shows the structure of an ER format that was configured to produce a zipped outbound file that contains a collection letter note and appropriate customer invoices in PDF format.



The following illustration shows the process, as described in this topic, of grouping individual components and enabling the **Email** destination for the new group, so that a collection letter note is sent together with appropriate customer invoices as email attachments.



## Additional resources



- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) destinations](#)
- [Formula designer in Electronic reporting \(ER\)](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

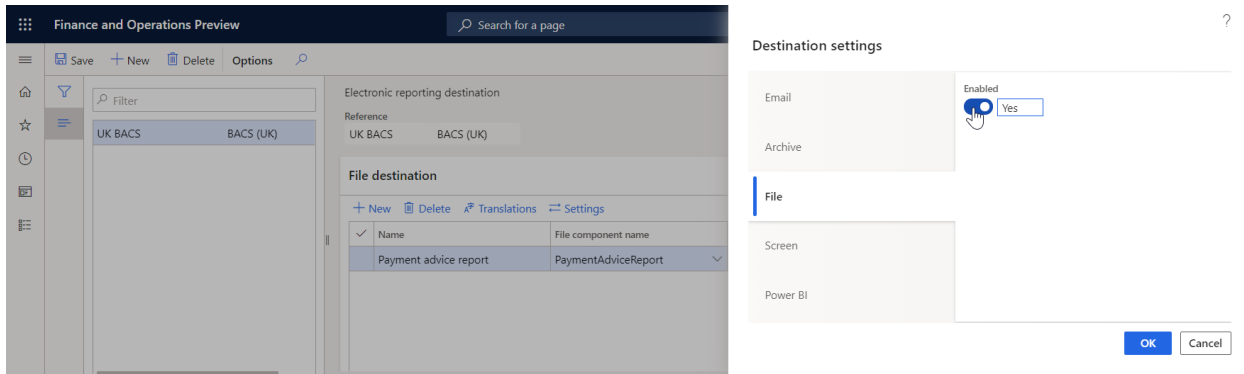
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# File destination

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can configure a file destination for each FOLDER or FILE component of an Electronic reporting (ER) format that is configured to generate outbound documents. Based on the setting of the destination, a generated document is available for download from the web browser.

On the **Destination settings** page, if you set **Enabled** to **Yes**, an open or save dialog box opens when the configuration has finished running.



## Additional resources

- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) destinations](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

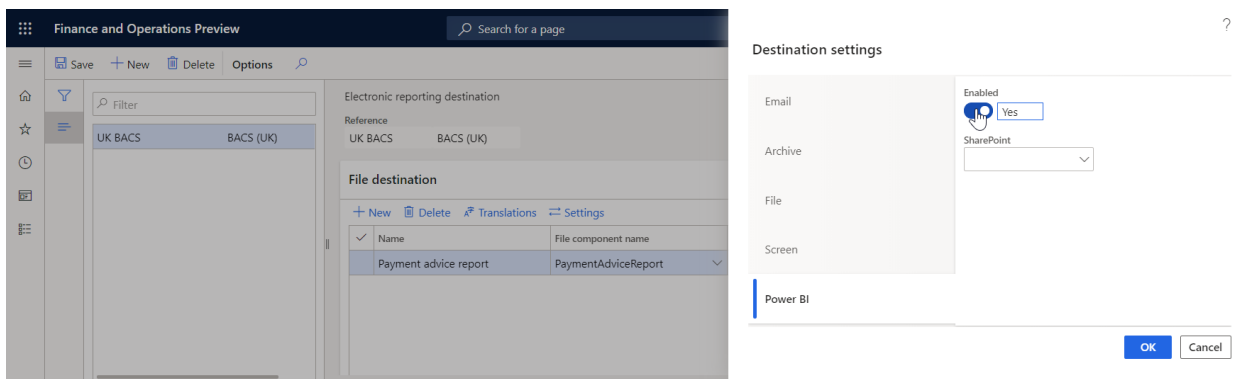
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Power BI destination

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can configure a Microsoft Power BI destination for each folder or file component of an Electronic reporting (ER) format that is configured to generate outbound documents. Based on the setting of the destination, a generated document is stored in a previously configured SharePoint folder.

Set **Enabled** to **Yes** to use your ER configuration to arrange the transfer of data from your Dynamics 365 Finance instance to Microsoft Power BI services. The transferred files are stored on a Microsoft SharePoint Server instance that must be configured for that purpose. For more information, see [Configure Electronic reporting \(ER\) to pull data into Power BI](#).



## Additional resources

- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) destinations](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Printer destination

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can send a generated document directly to a network printer for direct printing.

## Prerequisites

Before you begin, you must install and configure the Document Routing Agent, and then register the network printers. For more information, see [Install the Document Routing Agent to enable network printing](#).

## Make the Printer destination available

To make the **Printer** destination available in the current instance of Microsoft Dynamics 365 Finance, go to the **Feature management** workspace, and turn on the following features, in this order:

1. Convert Electronic Reporting outbound documents from Microsoft Office formats to PDF
2. Document Routing Agent as Electronic Reporting destination for outbound documents

Feature name ↑	Enable date	Feature added ↓	Module
Extended lookup of ER format configurations allowing to inquire the Global repository		1/14/2020	Organization administration
Convert Electronic Reporting outbound documents from Microsoft Office formats to PDF		1/6/2020	Organization administration
<b>Document Routing Agent as Electronic Reporting destination for outbound documents</b>		1/6/2020	Organization administration
Enable usage of EPPlus library in Electronic reporting framework		1/6/2020	Organization administration
Office-like UI experience for Business document management	1/15/2020	11/11/2019	Organization administration
Business document management	1/15/2020	6/3/2019	Organization administration

## Applicability

The **Printer** destination can be configured only for file components that are used to generate output in either printable PDF format (PDF Merger or PDF file format elements) or Microsoft Office Excel/Word format (Excel file). When output is generated in PDF format, it's sent to a printer. When output is generated in Microsoft Office format, it's automatically converted to PDF format and then sent to a printer.

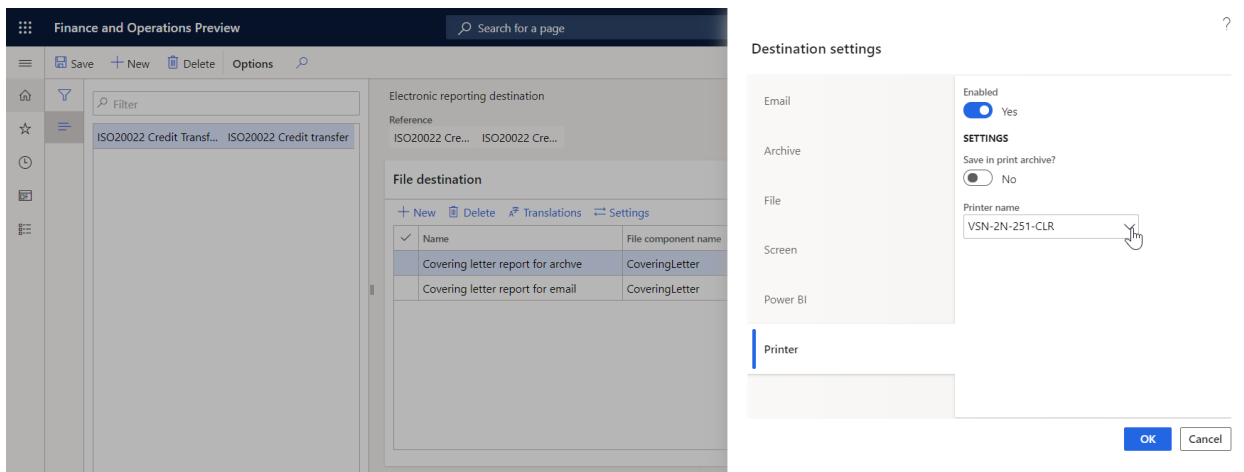
## Limitations

This feature is a preview feature and is subject to the terms of use that are described in [Supplemental Terms of Use for Microsoft Dynamics 365 Previews](#).

The **Printer** destination is implemented only for cloud deployments.

## Use the Printer destination

1. Set the **Enabled** option to **Yes** to send a generated document to a printer.
2. In the **Printer** name field, select the required network printer.
3. Set the **Save in print archive?** option to **Yes** to store the generated output in the print archive, so that it's available for further printing. To access archived output later, go to **Organization administration** > **Inquiries and reports** > **Report archive**.



#### NOTE

The **Convert to PDF** option doesn't have to be turned on when you configure the **Printer** destination. The PDF conversion for printing purposes will occur even if the option is turned off.

To use a specific [page orientation](#) when you print an outbound document in Excel format, you must turn on the **Convert to PDF** option. When you set the **Convert to PDF** option to **Yes**, the **Page orientation** field becomes available. In the **Page orientation** field, you can select a page orientation.

## Additional resources

- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) destinations](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

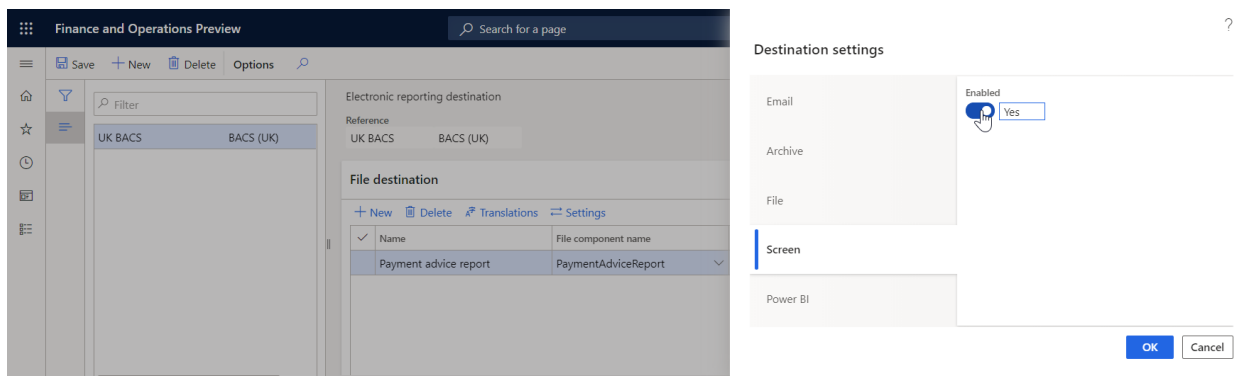
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Screen destination

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can configure a screen destination for each folder or file component of an Electronic reporting (ER) format that is configured to generate outbound documents. Based on the setting of the destination, a generated document is opened for preview in a separate browser tab.

If you set **Enabled** to **Yes**, a preview of the output is created. You can view some file types, such as XML, TXT, or PDF, directly in a browser window. For other file types, such as Microsoft Excel or Word, the Microsoft Office Online service is used.



## Additional resources

- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) destinations](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER Configure destinations

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure demonstrates how to set up and use different destinations for Electronic reporting (ER) output components, such as a folder or a file. The demo data company used to create this procedure is DEMF. Germany is the country\region of the legal entity's primary address, however you can use any legal entity for this procedure.

The format used in this example is ISO20022 Credit transfer, but you can use any format that you have already imported. Note, this procedure is an example of a single file and a single destination setup. More information about Electronic reporting destination management can be found in the Dynamics 365 Finance Help.

1. Go to Organization administration > Electronic reporting > Electronic reporting destination.
2. Click New to create a new set of destinations for a format.
3. In the Reference field, select a format for which you want to configure destinations.
  - If you don't have a value to select, it means that you have not imported any Electronic reporting format configurations. You must import a format configuration before setting up destinations.
4. Click New to create a new file destination.
  - Note, you can create one file destination for each output component of the same format, such as a folder or a file. You will be able to enable and disable destinations separately in the settings.
5. In the Name field, enter the user-friendly name of output component.
  - We recommend that you use meaningful names, such as "Payment file" or "Control report". These names will be presented to users at configuration runtime along with the destination settings.
6. In the File name, select a file or folder that is specific to the format.
7. Click Settings.
8. Select Yes in the Enabled field.
  - The Enabled check box on each tab enables and disables each destination separately. In this example, you'll enable sending an output file to a mail recipient when the file is generated.
9. Click Edit, to set up email recipients.
10. Click Add.
11. Click Print Management email.
12. In the Email source field, select an option.
  - You can select different email source types, such as a customer or a vendor type. This defines the type of argument that will be returned by the Email source account formula. The Email source account formula, described in a following step, is the place where you bind an email source. Select Vendor if your formula will return a vendor account. Use Vendor if you are using the ISO 20022 Credit Transfer configuration example.
13. Click Email source bind button.
14. In the Formula, enter a document-specific reference to a party type that you selected earlier.
  - Instead of typing, you can find a data source node that represents the party account, and click the Add data source button to update the formula. For example, if you use the ISO 20022 Credit Transfer configuration, the node representing a vendor account is '\$PaymentsForCoveringLetter'.Creditor.Identification.SourceID. Otherwise, enter any string value, such as "DE-001", to save a formula.
15. Click Save.
16. Close the page.
17. Click Edit to configure contact details for the party.

18. Select Yes in the Primary contact field.

- You may use different options to indicate what contact type of the party should be used as an email address for this destination. We use primary contact in this example.

19. Click OK.

20. Click OK.

21. In the Subject field, type a value.

22. Click OK.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Allow users to set up an ER format reference inquiring a format from the Global repository

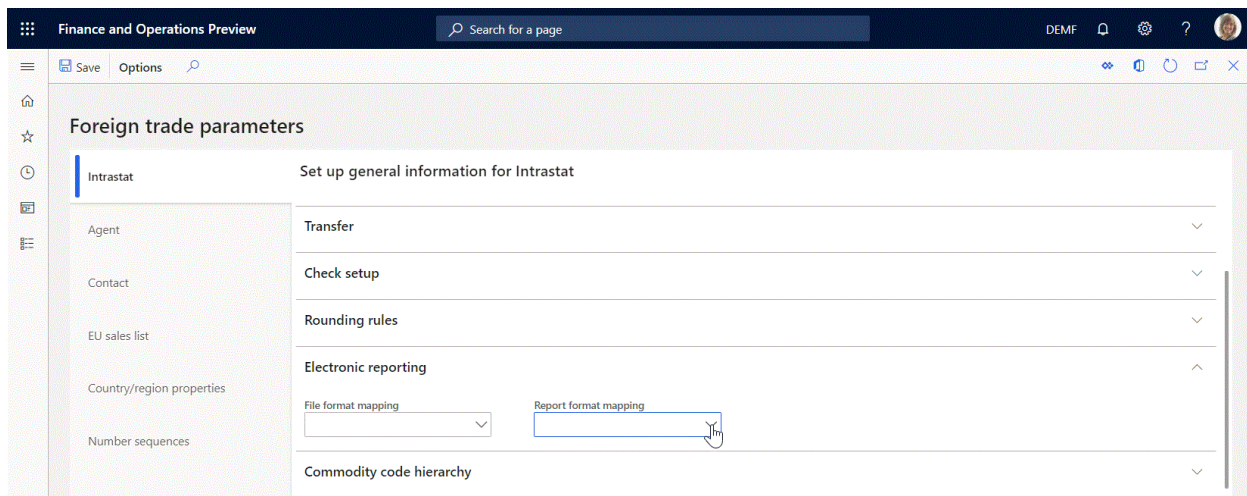
2/18/2021 • 4 minutes to read • [Edit Online](#)

You can use the [Electronic reporting](#) (ER) framework to configure [formats](#) for outbound documents in accordance to the legal requirements of various countries/regions. You can also use the ER framework to configure [formats](#) for parsing inbound documents and use the information from those documents to append or update application data. Each of these formats can be used in your Dynamics 365 Finance instance for handling inbound or outbound business documents as part of a certain business process.

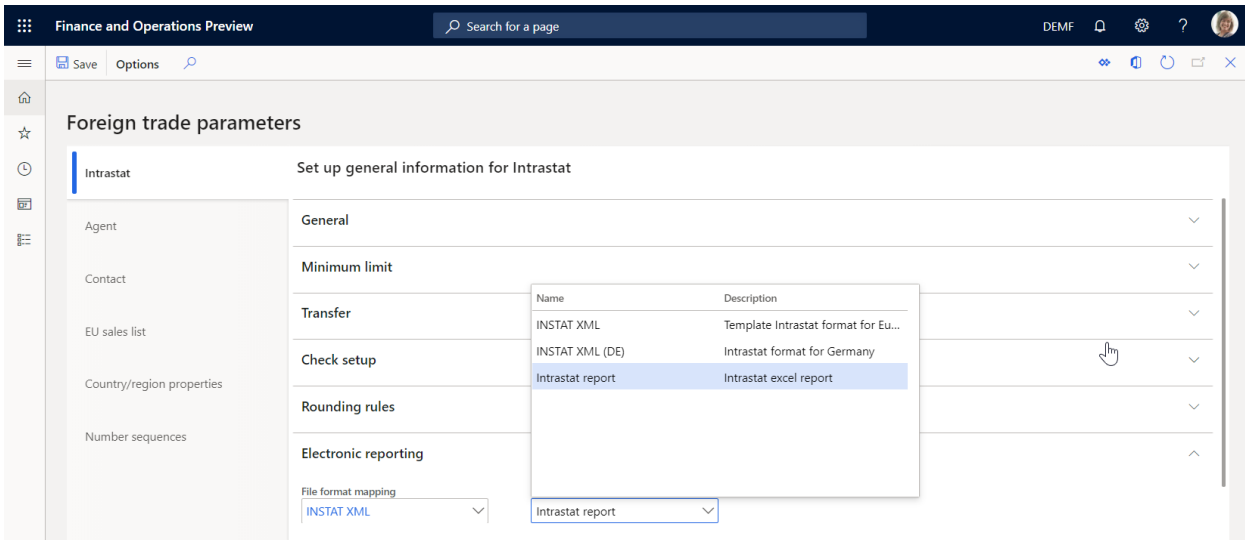
Usually, you must specify what ER format must be used in a certain business process. To do that, select a single ER format in a lookup field that is configured as part of business process-specific parameters. These lookup fields are usually implemented by using the appropriate API of the ER framework. For more information, see [ER framework API - code to display a format mapping lookup](#).

For example, when you configure [foreign trade parameters](#), you need to set up the references to individual ER formats that will be used to generate the Intrastat declaration and the Intrastat declaration control report. The screenshots below show how the ER formats lookup field looks like in the **Foreign trade parameters** page.

If the current Finance instance contains no Intrastat business process-related ER formats, this lookup field will be empty.

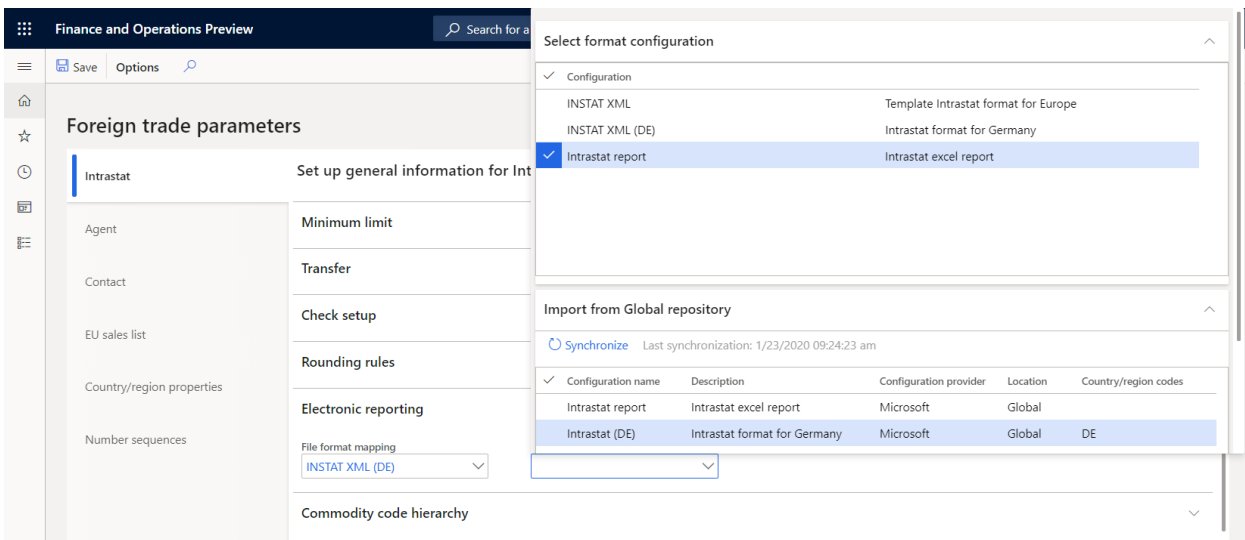


If the current Finance instance contains Intrastat business process related ER formats, this lookup field offers the ER formats.



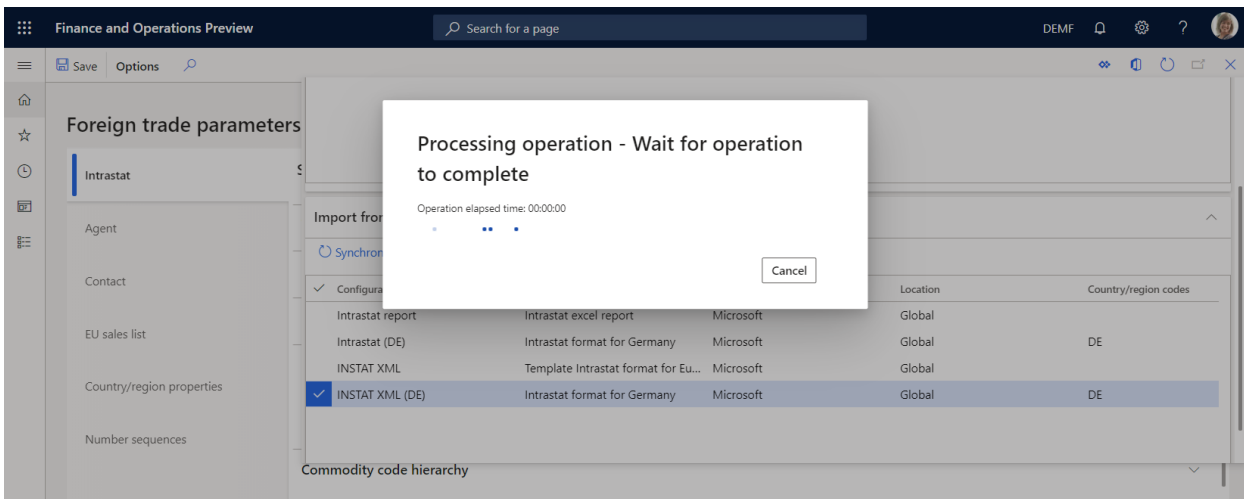
This lookup offers only the ER formats that have already been imported to the current Finance instance. To [import](#) ER solutions to the current Finance instance, you need to have permissions to run the appropriate function of the ER framework that supports the [lifecycle](#) of ER solutions that contain ER formats.

Starting in the Finance version 10.0.9 (April 2020 release), the user interface of the ER format lookup that is implemented by using the ER framework API, has been extended. You can still select the existing ER formats, which on the **Select format configuration** FastTab. In addition, the extended lookup offers the new option to search the Global repository (GR) to locate specific ER formats. All ER formats of the GR are offered on the **Import from Global repository** FastTab.

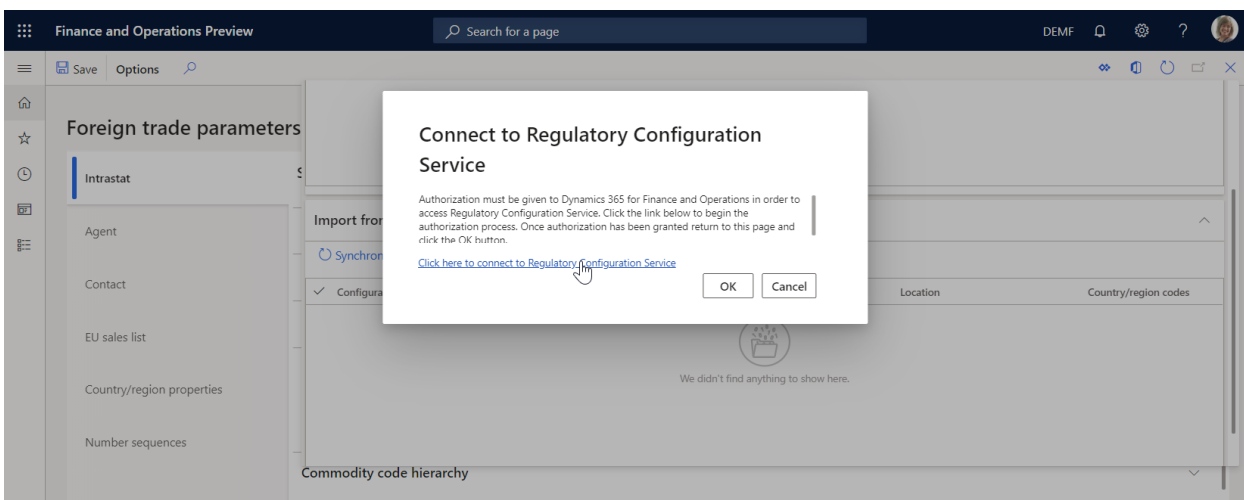


Similar to the **Select format configuration** FastTab, the **Import from Global repository** FastTab shows only the ER formats that are applicable to the business process for which an ER format is selected in this lookup field. In this example, the generation of Intrastat declaration. The ER format is applicable for the company to which the user is currently signed in, depending on the company country context.

When you select an ER format on the **Import from Global repository** FastTab, the selected ER format [configuration](#) is imported from the GR to the current Finance instance.



Then, if the import completes successfully, the reference to the imported ER format is stored in this lookup field. When you access the GR for the first time, you need to follow the link provided to sign up for the [Regulatory Configuration Service \(RCS\)](#) that is used to manage access to the GR storage.

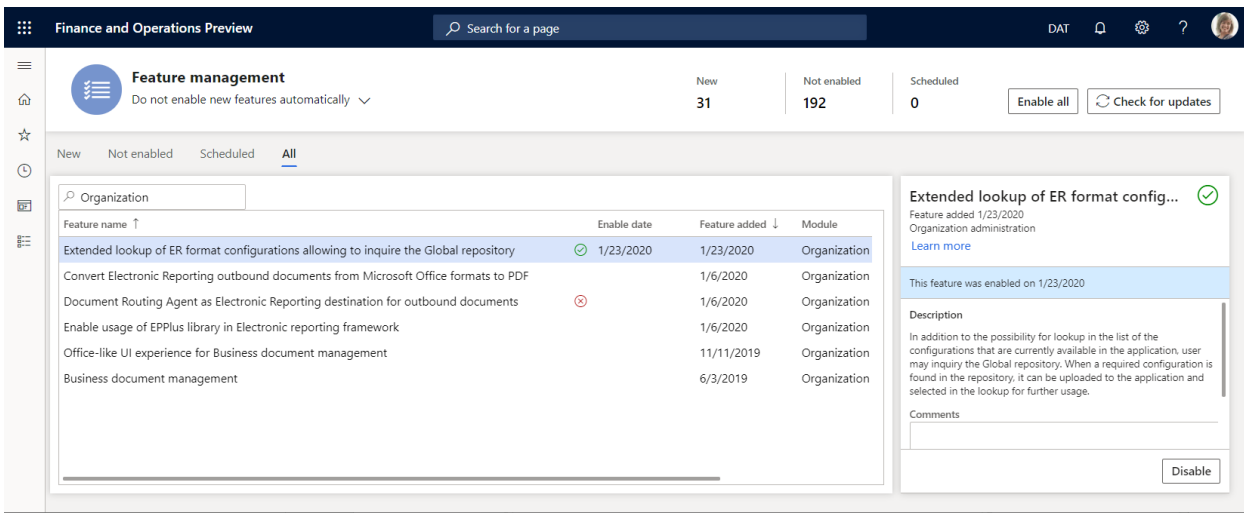


By default, the **Import from Global repository** FastTab presents the list of ER formats from the temporary storage that is automatically created based on the GR content for performance improvements. This happens when the **Import from Global repository** FastTab is opened the first time, which may take several seconds.

If you do not see the required ER format in the **Import from Global repository** FastTab, but you are sure that this ER format is stored in the GR, select the **Synchronize** option. This option will update the temporary storage and synchronize it with the current content of the GR.

## Feature activation

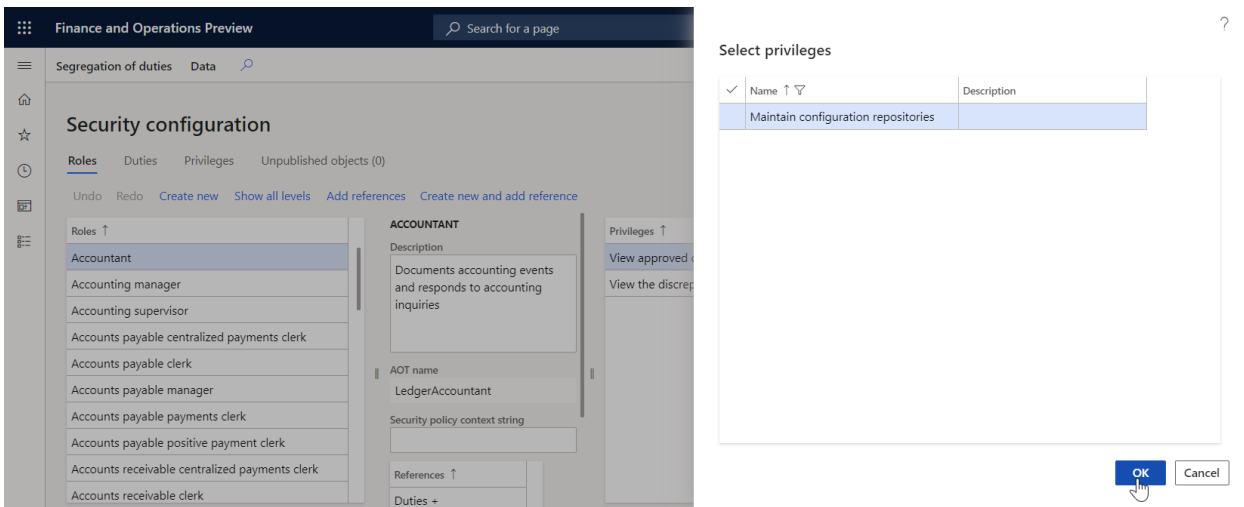
The availability of this functionality is controlled by the feature **Extended lookup of ER format configurations allowing to inquire the Global repository** in the **Feature management**. This feature is enabled by default.



## Security considerations

The **Maintain configuration repositories (ERMaintainSolutionRepositories)** privilege controls access to the GR for a user opening the ER format lookup with the enabled **Import from Global repository** FastTab. To allow users to access the GR content from the ER format lookups, you need to change the security settings by granting the **ERMaintainSolutionRepositories** privilege to users either directly or by using already assigned roles and duties.

The following screenshot shows how this privilege can be granted to users who are assigned to the **Accountant** role. This role allows users to configure foreign trade parameters and set up references to the ER formats in the **File format mapping** and **Report format mapping** fields on the **Foreign trade parameters** page.



## Limitations

Access to the GR in the ER format lookup is currently only supported for the selection of ER formats that are used to generate outbound documents.

## Frequently asked questions

### Why can't I access the Global repository from the ER format lookup?

If you have enabled the **Extended lookup of ER format configurations allowing to inquire the Global repository** feature on the **Feature management** page, but users can't see ER formats on the **Import from Global repository** FastTab and the **Synchronize** option is visible but disabled, make sure that the **Maintain configuration repositories (ERMaintainSolutionRepositories)** privilege has been granted to the user. Contact your system administrator to receive this privilege.

## Additional resources

- [Electronic reporting \(ER\) overview](#)
- [Electronic reporting \(ER\) framework API](#)
- [Manage Electronic reporting \(ER\) configurations lifecycle](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Trace generated report results and compare them with baseline values

2/18/2021 • 8 minutes to read • [Edit Online](#)

You can trace the results of Electronic reporting (ER) formats that generate outgoing electronic documents. When trace generation is turned on (by using the **Run in debug mode** ER user parameter), a new trace record is generated in the ER format execution log every time that an ER report is run. The following details are stored in each trace that is generated:

- All warnings that were generated by validation rules
- All errors that were generated by validation rules
- All generated files that are stored as attachments of the trace record

You can store individual baseline application files for any ER format. Files are considered baseline files when they describe the expected results of reports that are run. If a baseline file is available for an ER format that is run while trace generation is turned on, the trace stores, in addition to the details that were mentioned earlier, the result of the comparison of the generated electronic document with the baseline file. In one click, you can also get the generated electronic document and its baseline file in a single zip file. You can then do detailed comparison by using an external tool such as WinDiff.

You can evaluate the trace to analyze whether the electronic documents that are generated include the expected content. You can do this evaluation in a user acceptance testing (UAT) environment when the code base has been changed (for example, when you migrated to a new instance of the application, installed hotfix packages, or deployed code modifications). In this way, you can make sure that the evaluation doesn't affect the execution of ER reports that are used. For many ER reports, the evaluation can be done in unattended mode.

To learn more about this feature, play the **ER Generate reports and compare results (Part 1)** and **ER Generate reports and compare results (Part 2)** task guides, which are part of the **7.5.4.3 Test IT services/solutions (10679)** business process and can be downloaded from the [Microsoft Download Center](#). These task guides walk you through the process of configuring the ER framework to use baseline files to evaluate generated electronic documents.

## Example: Trace generated report results and compare them with baseline values

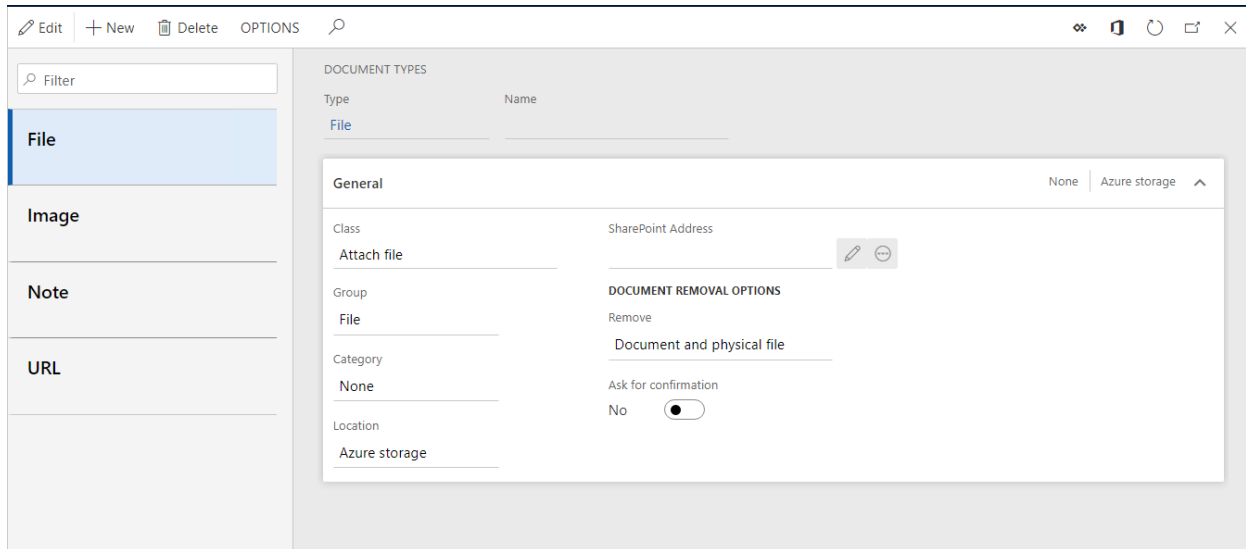
This procedure explains how to configure the ER framework to collect information about ER format executions and then evaluate the results of those executions. As part of that evaluation, generated documents are compared with their baseline files. In this example, you will create the required ER configurations for the Litware, Inc. sample company. This procedure is intended for users who have the System administrator or Electronic reporting developer role assigned to them. These steps can be completed by using any data set.

To complete the steps in this example, you must first complete the steps in [Create configuration providers and mark them as active](#).

1. Go to **Organization administration > Workspaces > Electronic reporting**.
2. On the **Localization configurations** page, in the **Configuration providers** section, verify that the configuration provider for the Litware, Inc. sample company is listed, and that it's marked as **Active**. If you don't see this configuration provider, follow the steps in [Create configuration providers and mark them as active](#).

## Configure document management parameters

1. Go to **Organization administration > Document management > Document types**, and create a new document type to store baseline files.
2. In the **Class** field, enter **Attach file**.
3. In the **Group** field, enter **File**.

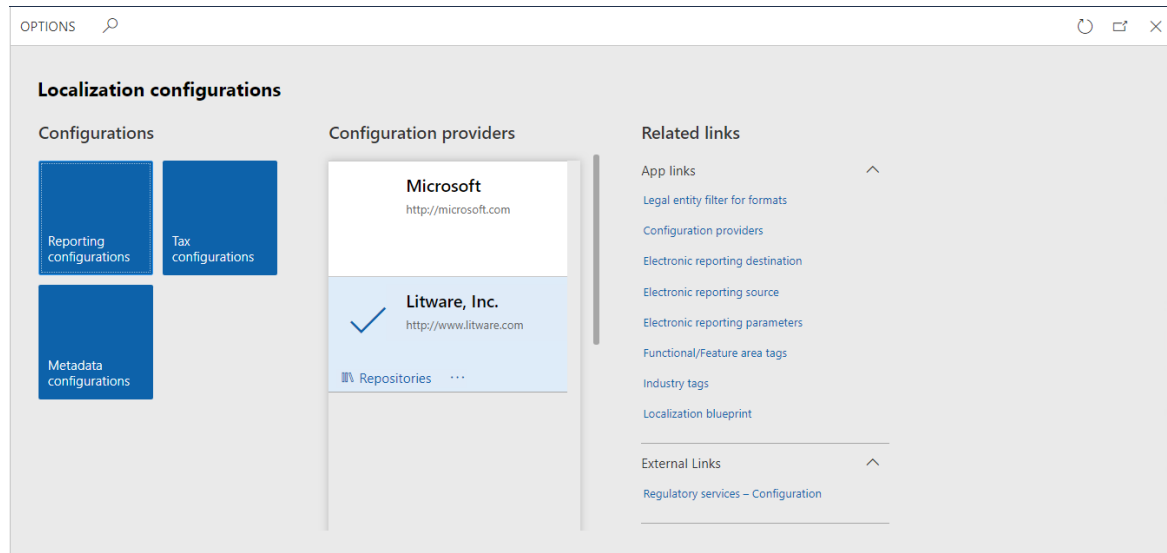


### NOTE

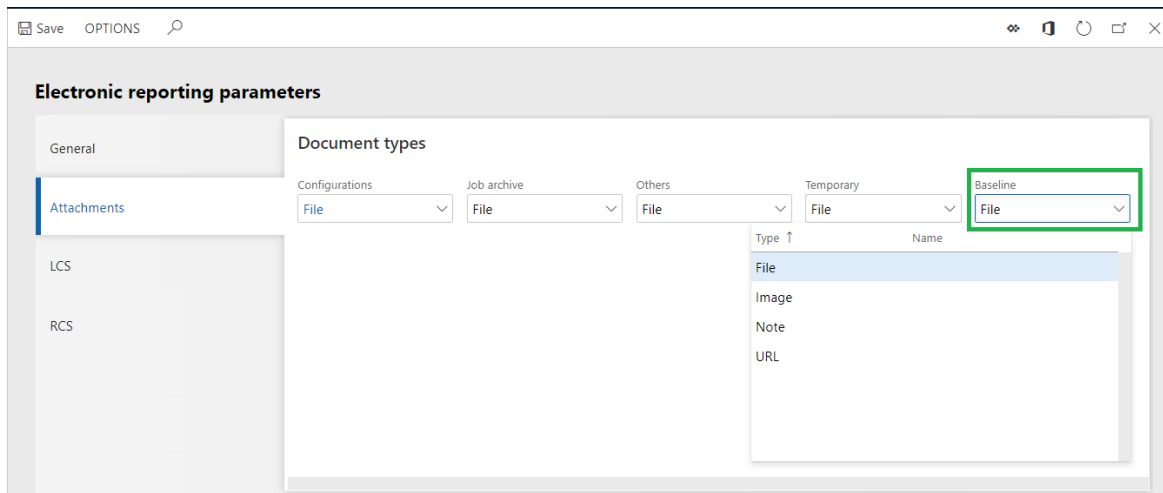
A new document type that has the same name must be configured for each data set where you plan to use the ER baseline feature.

## Configure ER parameters to start to use the baseline feature

1. In the **Electronic reporting workspace**, in the **Related links** section, select **Electronic reporting parameters**.



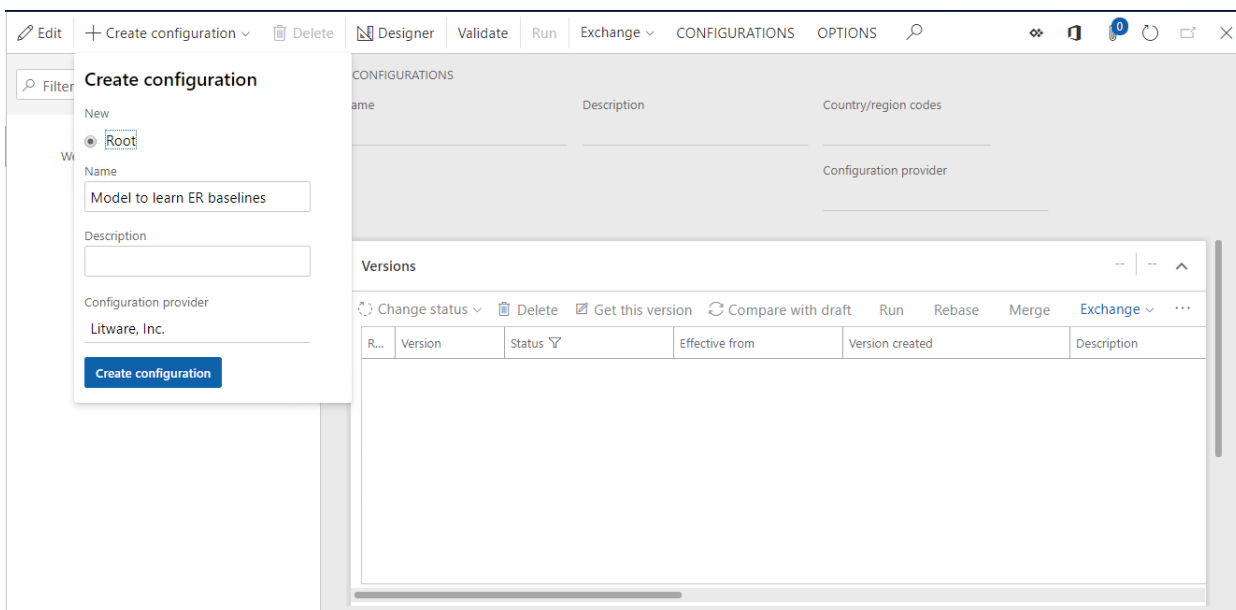
2. On the **Attachments** tab, in the **Baseline** field, enter or select the document type that you just created.



3. Select **Save**, and then close the **Electronic reporting parameters** page.

### Add a new ER model configuration

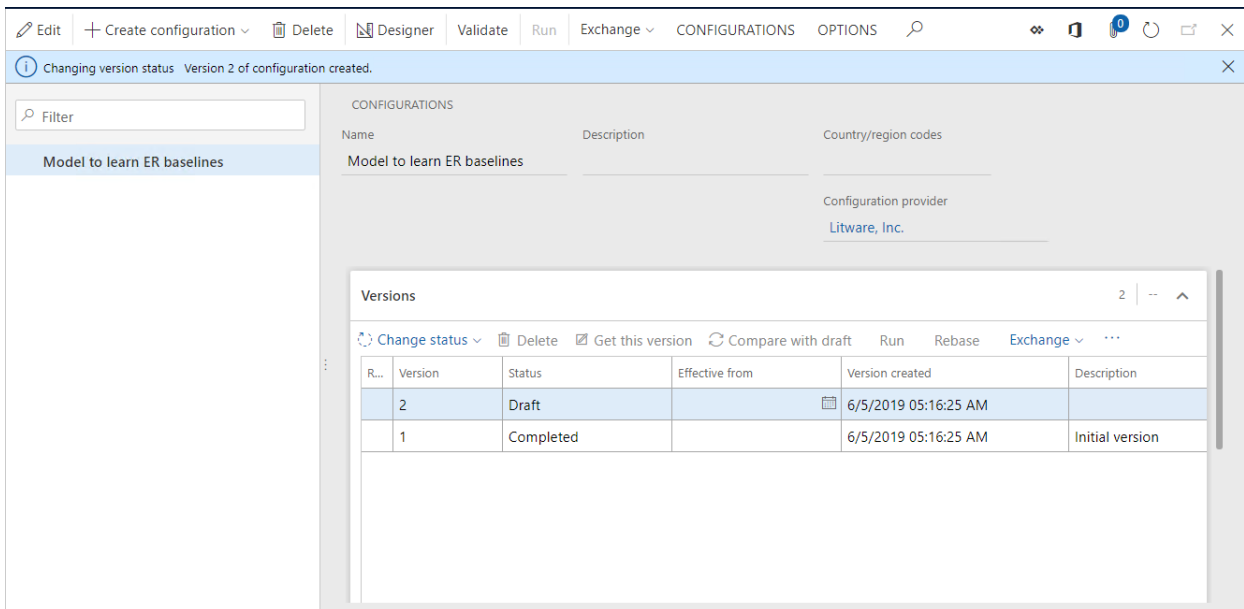
1. In the **Electronic reporting workspace**, in the **Configurations** section, select the **Reporting configurations** tile.
2. On the **Action Pane**, select **Create configuration**.
3. In the drop-down dialog box, in the **Name** field, enter **Model to learn ER baselines**.
4. Select **Create configuration** to confirm the creation of a new ER data model entry.



### Design a data model

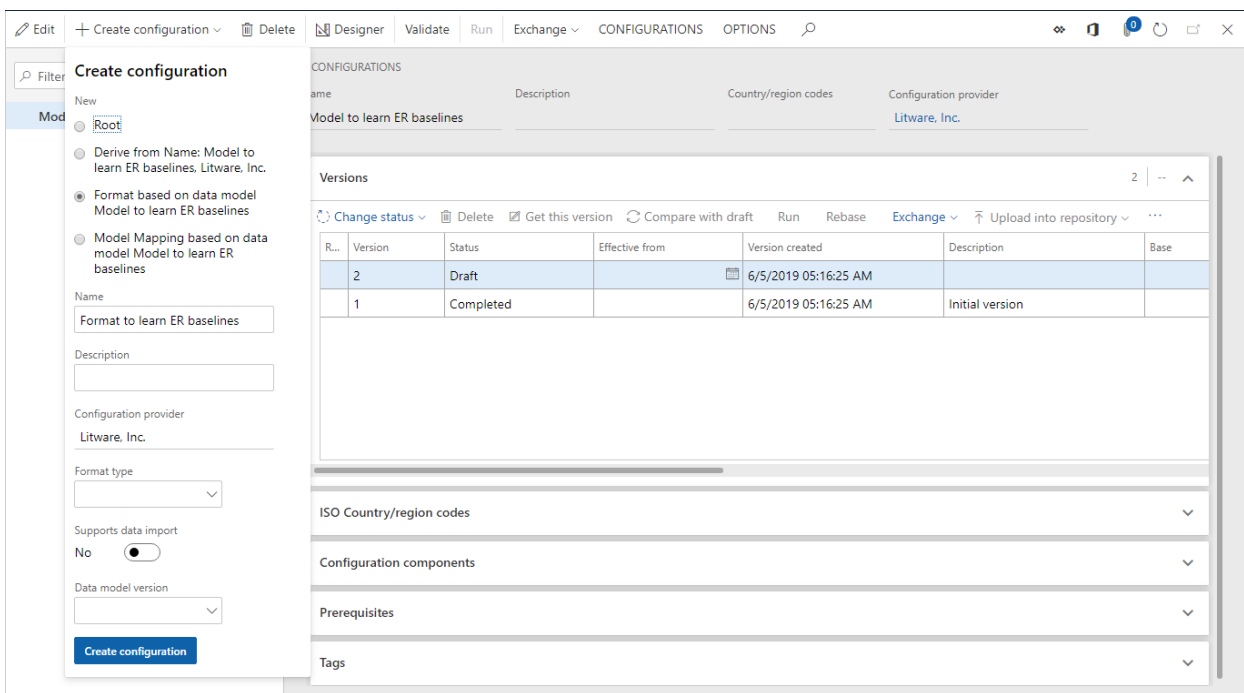
1. On the **Configurations** page, on the **Action Pane**, select **Designer**.
2. Select **New**.
3. In the drop-down dialog box, in the **Name** field, enter **Root**.
4. Select **Add**.
5. Select **Root reference**.
6. Select **OK**, and then select **Save**.
7. Close the **Model designer** page.
8. Select **Change status**.
9. Select **Complete**, and then select **OK**.





### Add a new ER format configuration

1. On the **Configurations** page, on the Action Pane, select **Create configuration**.
2. In the drop-down dialog box, in the **New** field group, select **Format based on data model Model to learn ER baselines**.
3. In the **Name** field, enter **Format to learn ER baselines**.
4. Select **Create configuration** to confirm the creation of a new ER format entry.

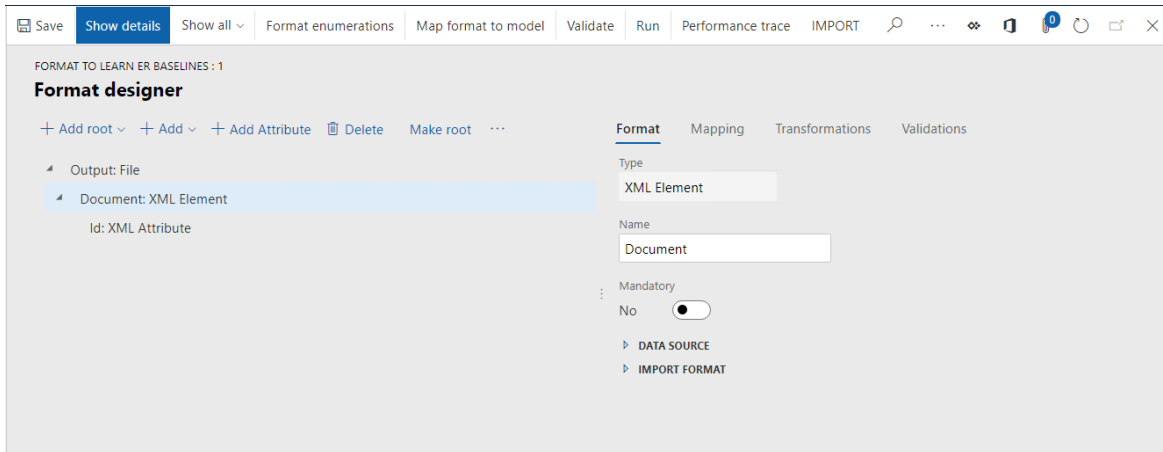


### Design a format

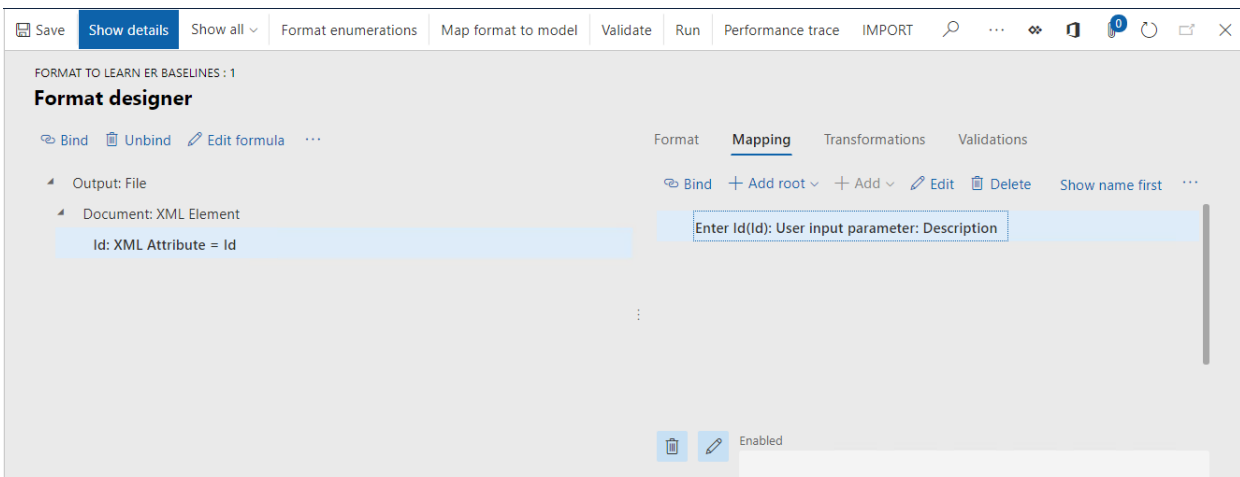
For this example, you will create a simple ER format to generate XML documents.

1. On the **Configurations** page, on the Action Pane, select **Designer**.
2. Select **Add root**.
3. In the drop-down dialog box, follow these steps:
  - a. In the tree, select **Common\File**.
  - b. In the **Name** field, enter **Output**.
  - c. Select **OK**.

4. Select **Add**.
5. In the drop-down dialog box, follow these steps:
  - a. In the tree, select **XML\Element**.
  - b. In the **Name** field, enter **Document**.
  - c. Select **OK**.
6. In the tree, select **Output\Document**.
7. Select **Add**.
8. In the drop-down dialog box, follow these steps:
  - a. In the tree, select **XML\Attribute**.
  - b. In the **Name** field, enter **ID**.
  - c. Select **OK**.



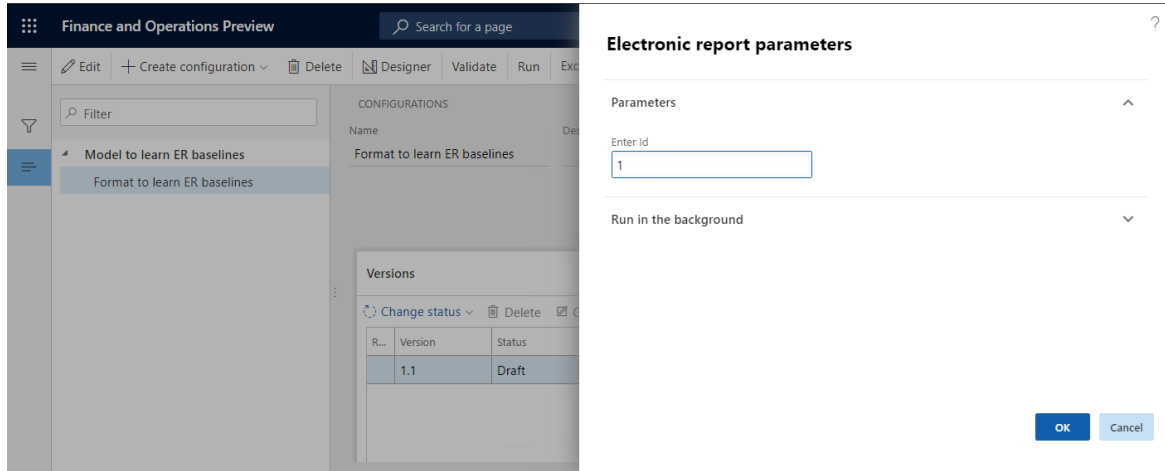
9. On the **Mapping** tab, select **Delete**.
10. Select **Add root**.
11. In the drop-down dialog box, in the tree, select **General\User input parameter**, and then follow these steps:
  - a. In the **Name** field, enter **ID**.
  - b. In the **Label** field, enter **Enter ID**.
  - c. Select **OK**.
12. In the tree, select **Output\Document\Id**.
13. Select **Bind**, and then select **Save**.



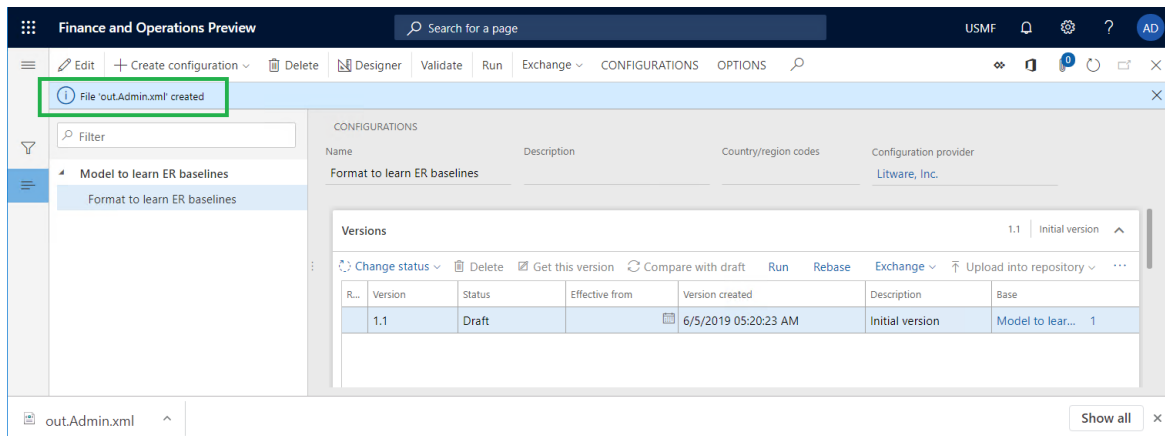
Based on the designed structure, the configured format will generate an XML file. This XML contains the Root element that has the ID attribute that is set to the value that the user enters in the ER runtime dialog box.

### Generate a new baseline file for a designed ER format

1. On the Configurations page, on the Versions FastTab, select Run.
2. In the Enter ID field, enter 1.
3. Select OK.

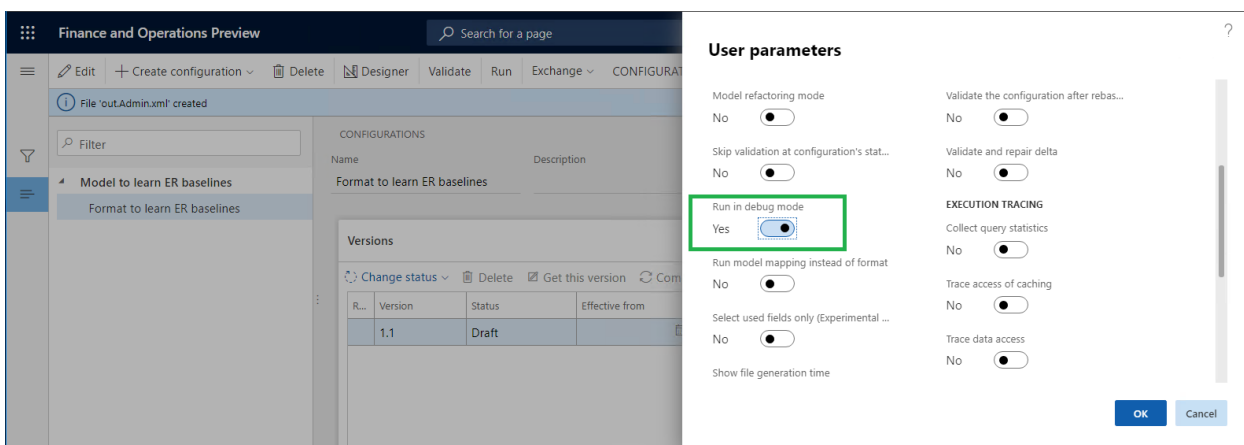


4. Save a local copy of the out.Admin.xml file that is generated, so that you can use it later as a baseline for this ER format.



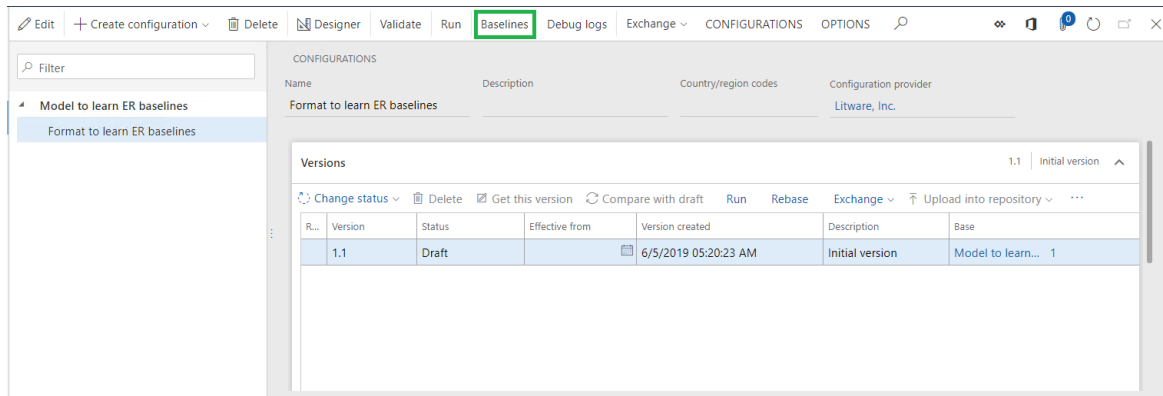
### Configure ER parameters to use the baseline feature

1. On the Configurations page, on the Action Pane, on the Configurations tab, select User parameters.
2. Set the Run in debug mode option to Yes.
3. Select OK.

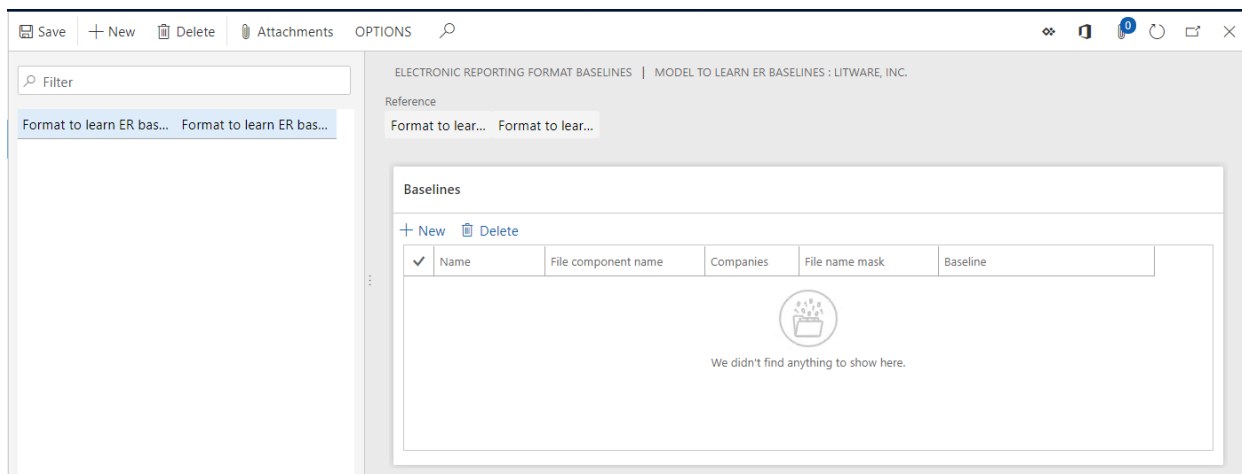


## Add a new baseline for designed ER format

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the Action Pane, select **Baselines**.



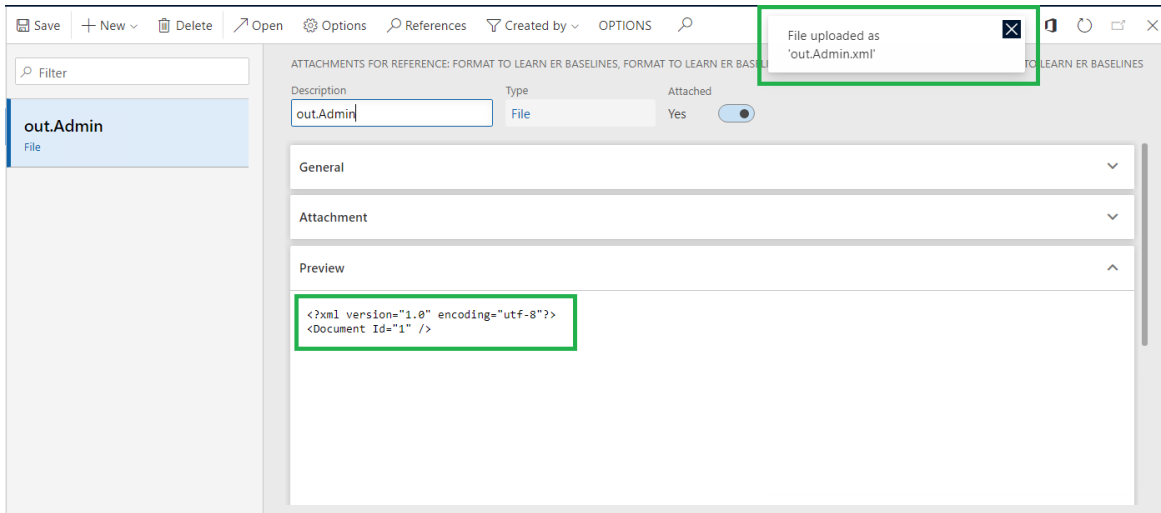
3. On the Action Pane, select **New**.
4. Select the **Format to learn ER baselines** ER format that you designed earlier.
5. Select **Save**.



The baseline is added for the **Format to learn ER baselines** format.

## Configure a baseline rule for the added baseline

1. On the **Electronic reporting format baselines** page, on the Action Pane, select the **Attachments** button (the paper clip symbol).
2. On the Action Pane, select **New > File**. In the ER parameters, the **File** document type should have been previously selected as the document type that is used to store baseline files.
3. Select **Browse**, and select the **out.Admin.xml** file that was generated when you ran the configured ER format earlier.

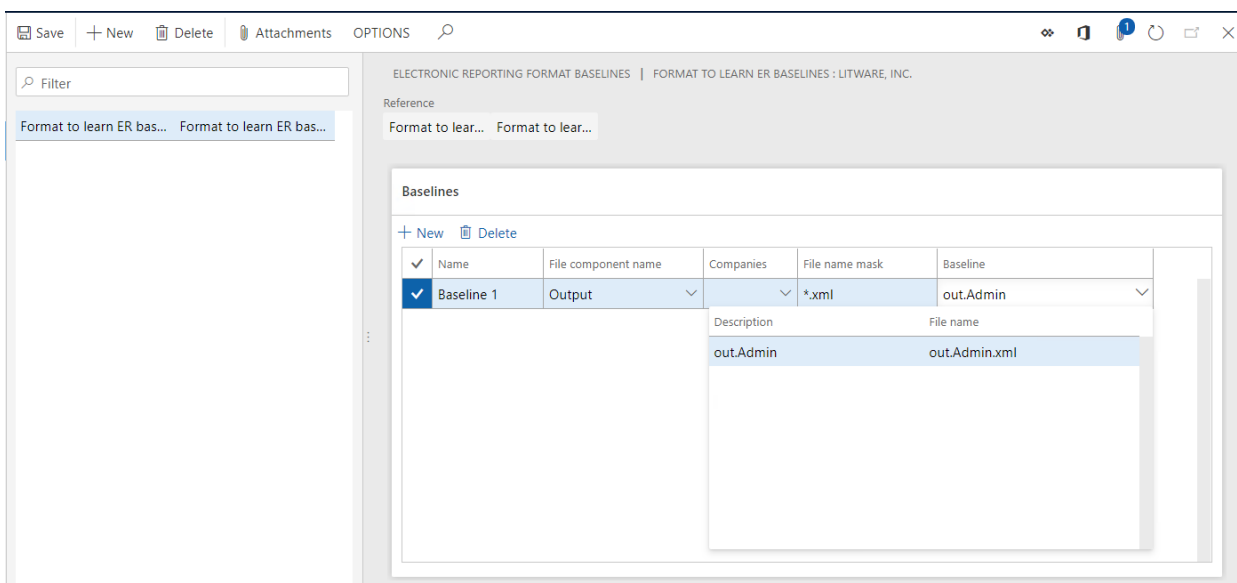


4. Close the **Attachments** page.
5. On the **Baselines** FastTab, select **New**.
6. In the **Name** field, enter **Baseline 1**.
7. In the **File component name** field, enter or select **Output**. This value indicates that the configured baseline will be compared with a file that is generated by using the **Output** format element.
8. In the **File name mask** field, enter **\*.xml**.

**NOTE**

You can define the file name mask. When the file name mask is defined, the baseline record will be used to evaluate the generated output only when the name of the output file that is generated satisfies that mask.

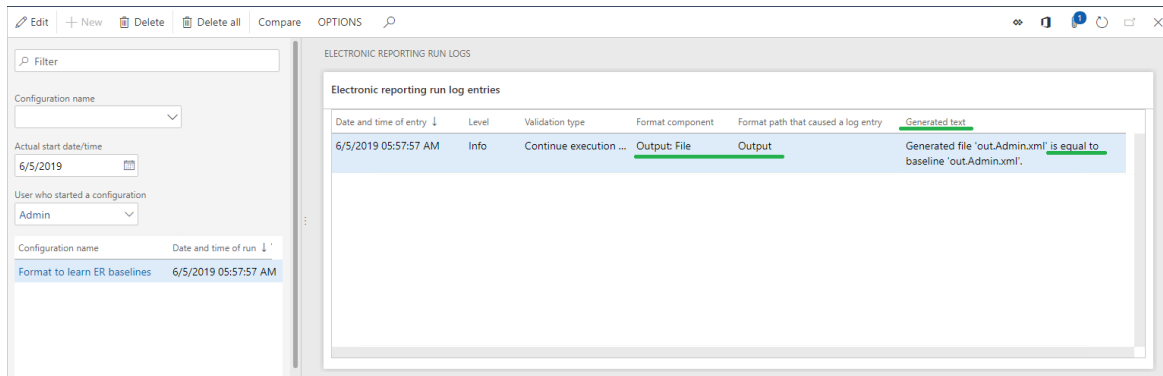
9. If the configured baseline should be used only when the **Format to learn ER baselines** ER format is run by users who are signed in to specific companies, select those companies in the **Companies** field.
10. In the **Baseline** field, enter or select the **out.Admin** attachment.
11. Select **Save**.



**Run the designed ER format and review the log to analyze the results**

1. Go to **Organization administration > Electronic reporting > Configurations**.

- In the tree, expand **Model to learn ER baselines**, and then select **Model to learn ER baselines\Format to learn ER baselines**.
- On the **Versions** FastTab, select **Run**.
- In the **Enter ID** field, enter **1**.
- Select **OK**.
- Go to **Organization administration > Electronic reporting > Configuration debug logs**.



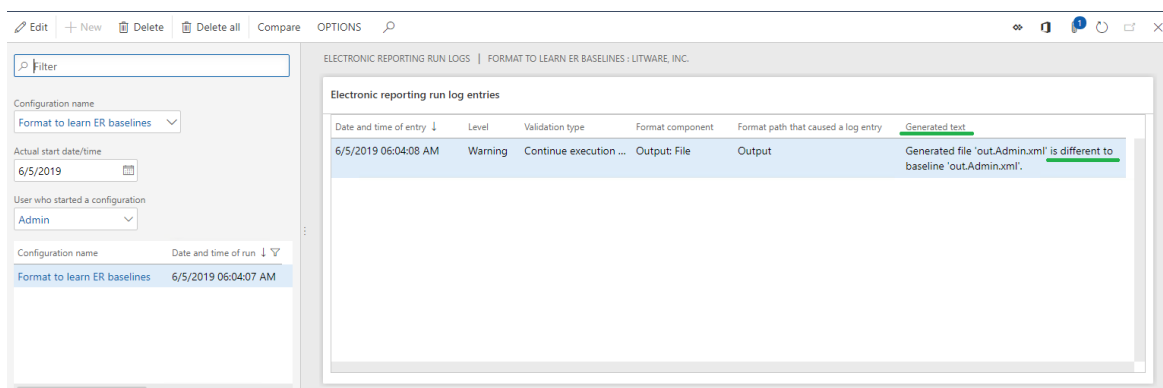
#### NOTE

The execution log contains information about the results of the comparison of the generated file with the configured baseline. In this example, the log indicates that the generated file and the baseline are equal.

- Select **Delete all**.

#### Run the designed ER format and review the log to analyze the results

- Go to **Organization administration > Electronic reporting > Configurations**.
- In the tree, expand **Model to learn ER baselines**, and then select **Model to learn ER baselines\Format to learn ER baselines**.
- On the **Versions** FastTab, select **Run**.
- In the **Enter ID** field, enter **2**.
- Select **OK**.
- Go to **Organization administration > Electronic reporting > Configuration debug logs**.



**NOTE**

The execution log contains information about the results of the comparison of the generated file with the configured baseline. In this example, the log indicates that the generated file and the baseline differ.

## 7. Select **Compare**.

**NOTE**

The generated file and the baseline file are offered as a zip file. You can use external comparison tools such as WinDiff to compare the files and review the differences.

## Additional resources

- [Configure the Electronic reporting \(ER\) framework](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Improvements in tracing the results of generated ER reports and comparing them with baseline values

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes the first set of improvements that have been made to the baseline feature of the Electronic reporting (ER) framework. These improvements are available in Microsoft Dynamics 365 for Finance and Operations version 10.0.3 (June 2019) and later.

## Automate the setting of baseline rules

The [Trace generated report results and compare them with baseline values](#) topic explains how to configure the ER framework to collect information about ER format executions and evaluate the results of those executions. The example in this topic shows the steps that must be completed.

Here are some of the steps:

- Run an ER format to generate an outbound file, and store the file locally.
- Add the locally stored file as an attachment of the baseline that was added for an ER format.
- Configure the baseline rule for the added baseline. This configuration includes the following steps:
  - Specify an ER format element that is used to generate an outbound file.
  - Select the attachment that refers to the generated outbound file.

### NOTE

These steps must be done manually, even though the new ER capabilities enable them to be automated. To learn more about this feature, complete the following example.

## Example: Automate the setting of baseline rules

To complete the steps in this example, you must first complete the steps in the example in the [Trace generated report results and compare them with baseline values](#) topic, up through the "Add a new baseline for a designed ER format" section.

### Review added baseline

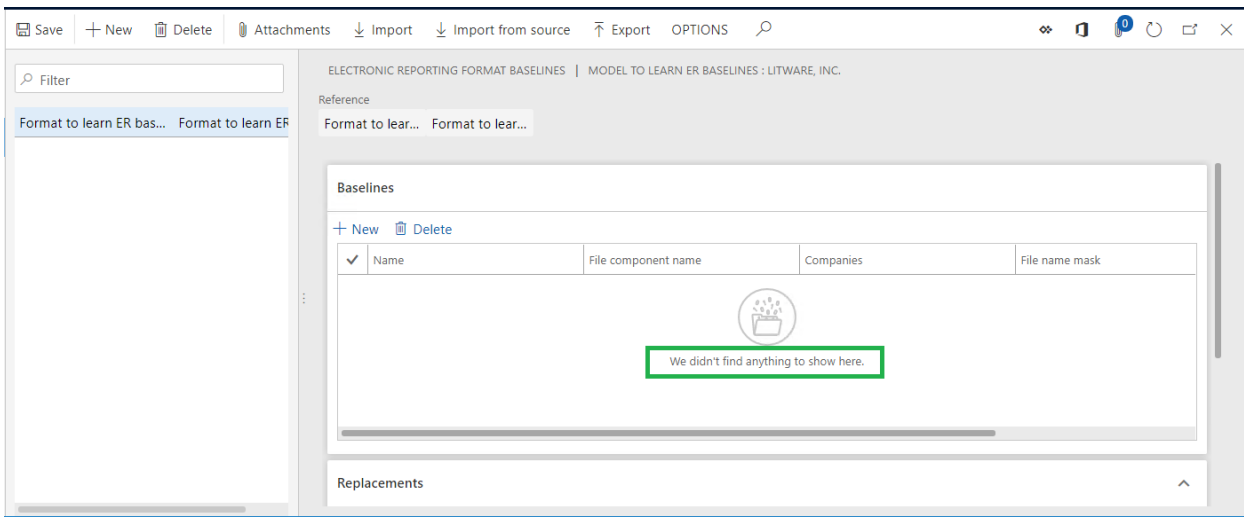
1. Go to **Organization administration > Electronic reporting > Configurations**.
2. Select **Baselines**.

### NOTE

The **Baselines** button on the Action Pane is available only when the **Run in debug mode** ER user parameter is turned on for the current company.

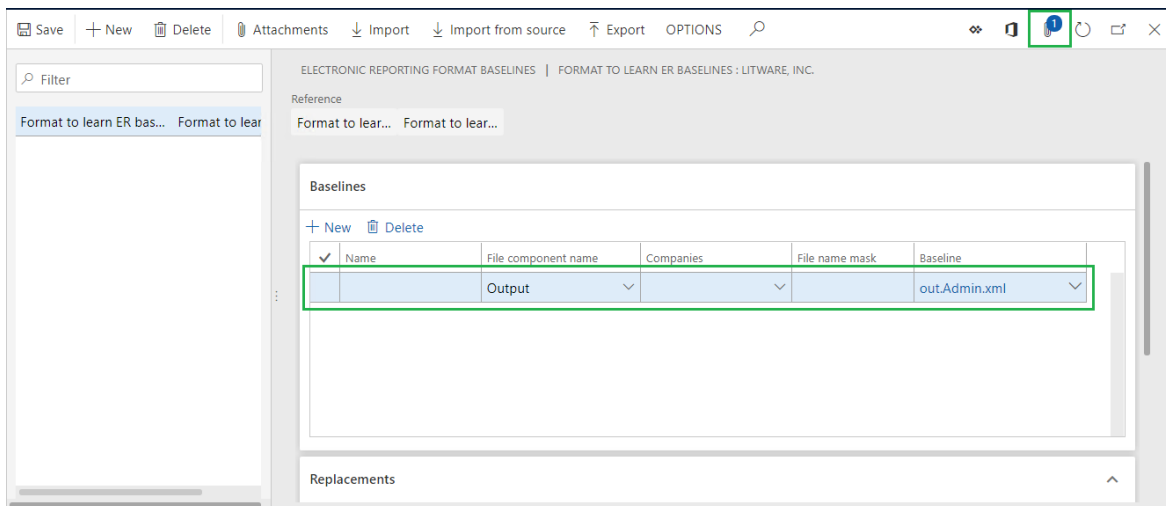
The baseline has been added for the selected **Format to learn ER baselines** format, but the baseline rules haven't yet been added for this baseline.





### Make a new baseline rule

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. In the tree, expand **Model to learn ER baselines**.
3. In the tree, select **Model to learn ER baselines\Format to learn ER baselines**.
4. On the **Versions** FastTab, select **Run**.
5. In the **Enter Id** field, enter **1**.
6. Set the **Make baseline files** option to **Yes**.
7. Select **OK**.
8. Select **Baselines**.



The generated outbound file has been automatically attached to the baseline of the executed ER format. The baseline rule has been automatically added to this baseline and also contains the reference to the attached file.

9. In the **Name** field, enter **Baseline 1**.
10. In the **File name mask** field, enter **.xml**.
11. Select **Save**.

### Run the format

You're now ready to complete the remaining steps in the example in the [Trace generated report results and compare them with baseline values](#) topic, starting from the "Run the designed ER format and review the log to

analyze the results" section.

#### NOTE

When you delete the automatically added baseline rule on the **Baselines** FastTab, the referenced attachment isn't automatically deleted.

## Configure the baseline so that it ignores constantly changing parts of the ER output

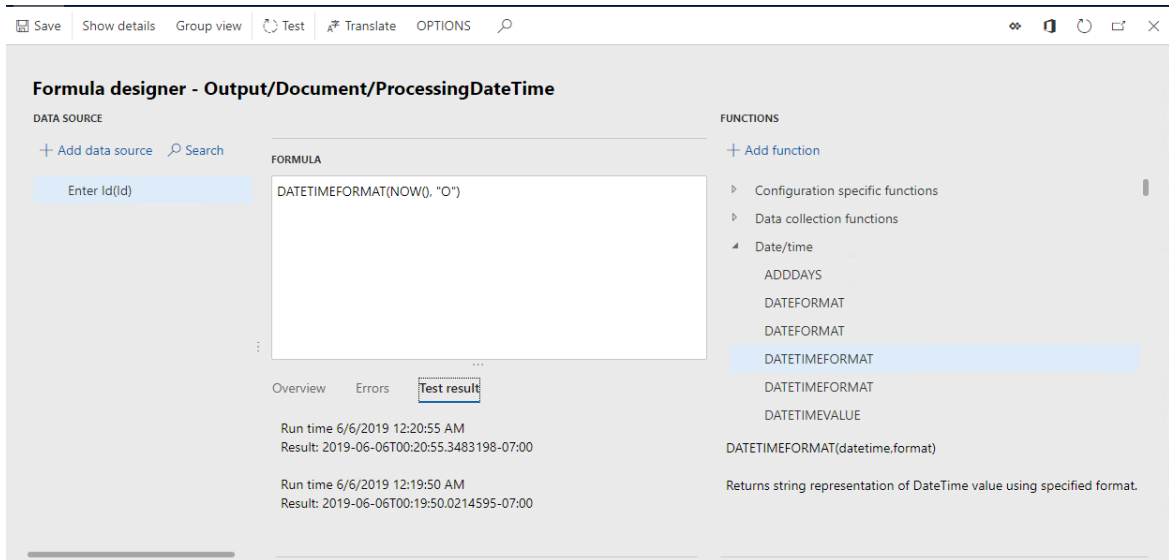
When an ER format has been designed to contain information that is changed when the format is run, the format must be required to use the ER baseline feature to compare the generated results with baseline values. For example, the information might be the processing date and time or the unique identifier of a generated document in different formats (globally unique identifier [GUID], and so on). The new ER capabilities let you configure the baseline rule so that it ignores changeable elements of an ER format when the format is run with the purpose of comparing baseline values with the results of the format's execution. To learn more about this feature, complete the following example.

### Example: Configure the baseline so that it ignores constantly changing parts of the ER output

To complete the steps in this example, you must first complete the steps in the example in the [Trace generated report results and compare them with baseline values](#) topic.

#### Modify a configured ER format

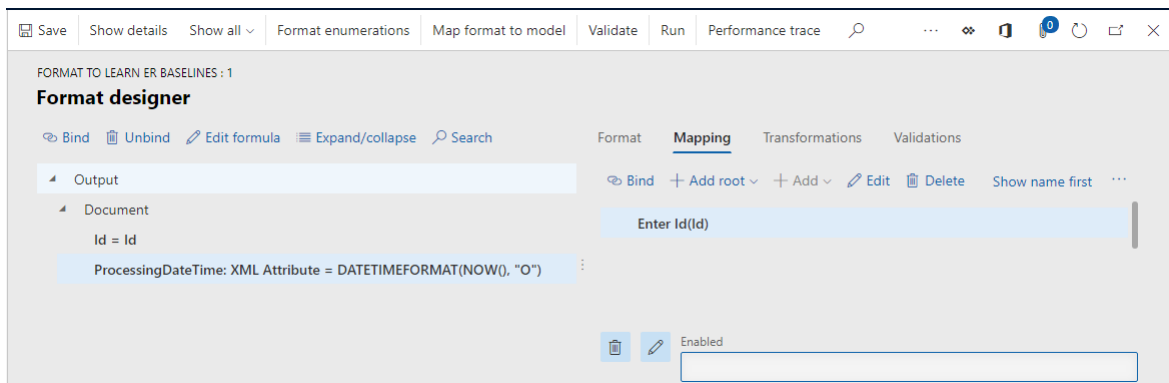
1. Go to **Organization administration > Electronic reporting > Configurations**.
2. In the tree, expand **Model to learn ER baselines**.
3. In the tree, select **Model to learn ER baselines\Format to learn ER baselines**.
4. Select **Designer**.
5. In the tree, select **Output\Document**.
6. Select **Add**.
7. In the drop-down dialog box, in the tree, select **XML\Attribute**.
8. In the **Name** field, enter **ProcessingDateTime**.
9. Select **OK**.
10. On the **Mapping** tab, in the tree, select **Output\Document\ProcessingDateTime**.
11. Select **Edit formula**.
12. In the **Formula** field, enter the following expression: `DATETIMEFORMAT(NOW(), "O")`
13. Select **Save**, and then select **Test**.
14. Select **Test** again to retest the configured expression.



**NOTE**

The **Test result** tab shows that the configured expression returns a different date and time value whenever it's called.

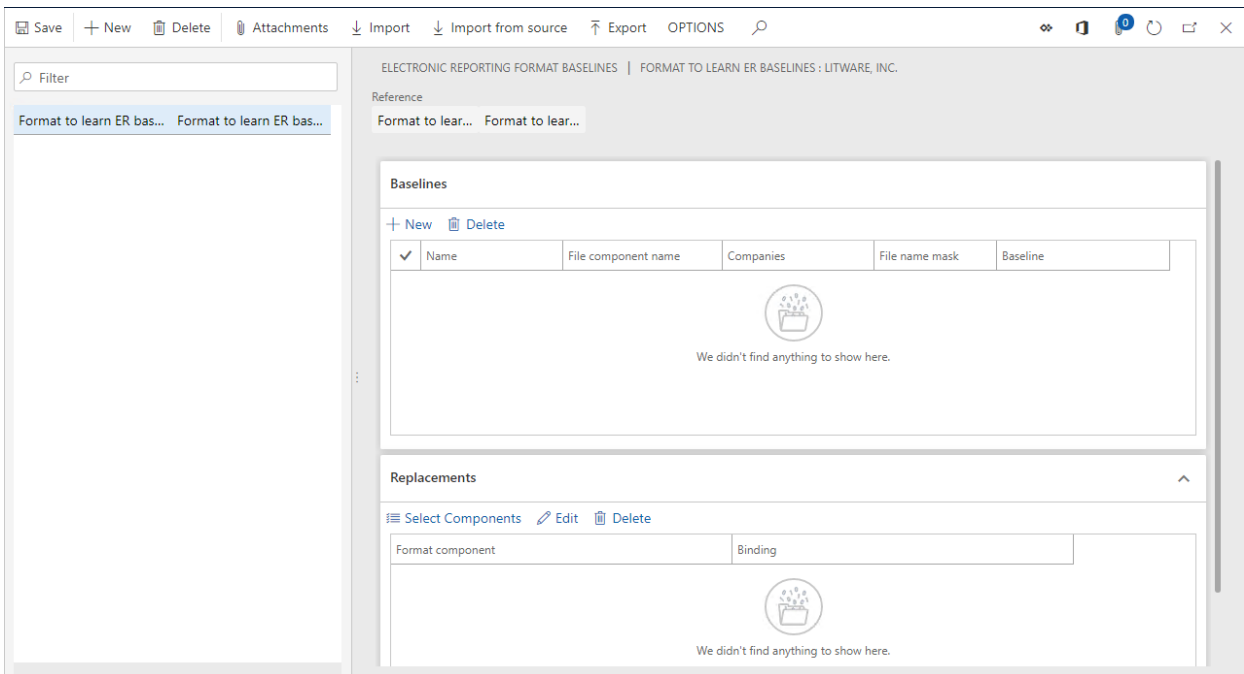
15. Close the Formula designer page, and then select **Save**.



16. Close the Format designer page.

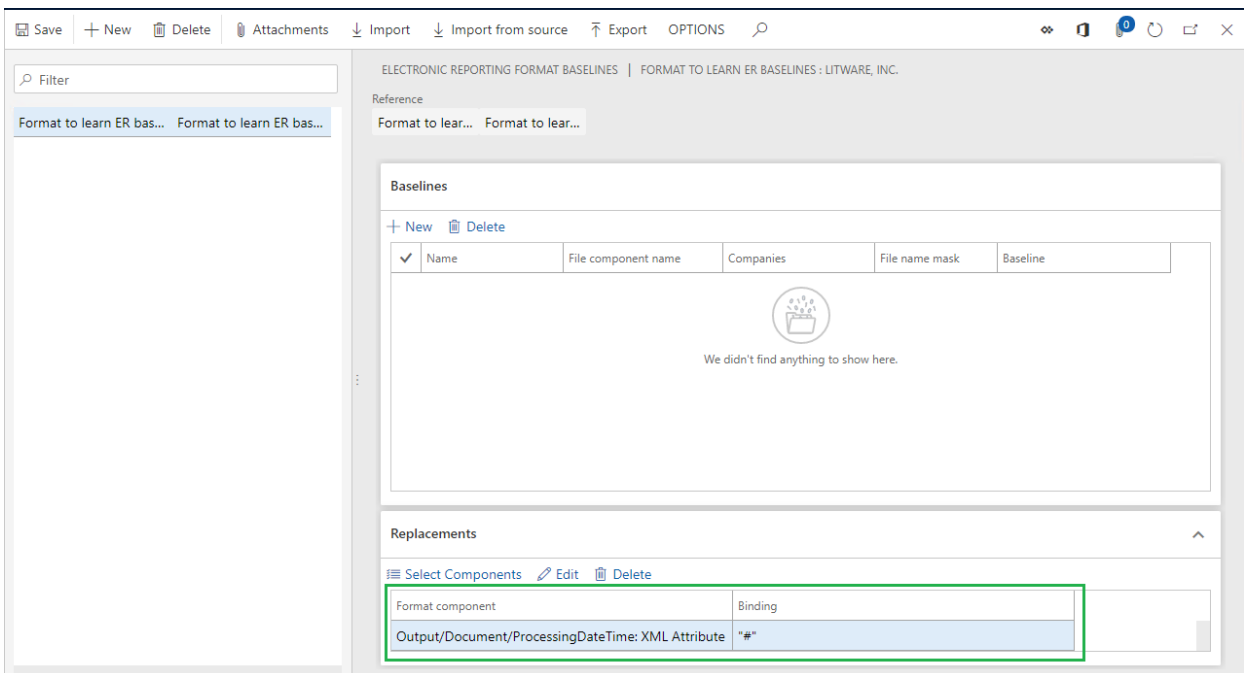
**Remove an existing baseline rule**

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. Select **Baselines**.
3. In the list of baselines, select the baseline that is configured for the **Format to learn ER baselines** format.
4. On the **Baselines FastTab**, select **Delete** to remove the baseline rule that you configured earlier.



### Define replacements for bindings of designed ER format

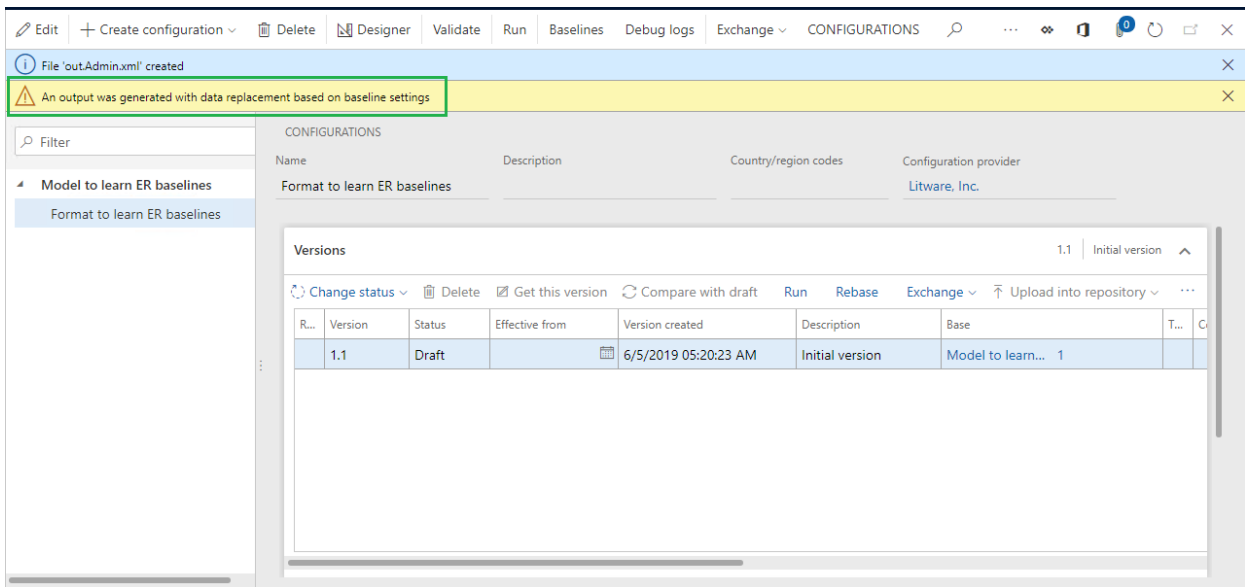
1. On the **Configurations** page, on the **Replacements** FastTab, select **Select components**.
2. In the format components tree, expand **Output**, expand **Output\Document**, and then select the check box for **Output\Document\ProcessingDateTime**.
3. Select **OK**.



The selected ER format component has been added to the list of components on the **Replacements** FastTab. When the base ER format is run in debug mode, the format's binding for each component will be replaced by the binding that is shown in the **Binding** column. To change the default binding for a component that is listed on the **Replacements** FastTab, select **Edit**.

### Make a new baseline rule

Follow the steps in the "Example: Automate the setting of baseline rules" section earlier in this topic. A notification warns you that the outbound file has been generated by using baseline settings, and that a forced replacement of the format bindings has occurred.



### Suppress warnings about the replacement of format bindings

By setting specific ER parameters, you can suppress notifications that warn about the replacement of format bindings. This suppression can be useful when format bindings are replaced in an unattended mode by using the Regression Suite Automation Tool. In this case, the warning can be considered a failure of the test case that is running.

1. On the **Configurations** page, on the Action Pane, on the **Configurations** tab, select **User parameters**.
2. Set the **Suppress baseline warnings** option to **Yes**, and then select **OK**.

### Review the generated baseline file

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. Select **Baselines**.
3. Select **Attachments**.

#### NOTE

The generated file contains the processing date and time text ("#") from the binding that was configured in the added baseline rule, not from the format's binding.

4. Close the **Attachments** page.

### Run the designed ER format and review the log to analyze the results

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. In the tree, expand **Model to learn ER baselines**.
3. In the tree, select **Model to learn ER baselines\Format to learn ER baselines**.
4. On the **Versions** FastTab select **Run**.
5. In the **Enter Id** field, type **1**.
6. Select **OK**.
7. Go to **Organization administration > Electronic reporting > Configuration debug logs**.

The execution log contains information about the results of the comparison of the generated file with the configured baseline. The log indicates that the generated file and the baseline are equal, even though the executed format contains the binding to enter a constantly changing date and time value in the outbound file.

## NOTE

Although the outbound file has been generated by using baseline settings that force the replacement of the format's bindings, you don't receive any warnings about the replacement.

# Exchange baseline settings between environments

## Export baseline settings

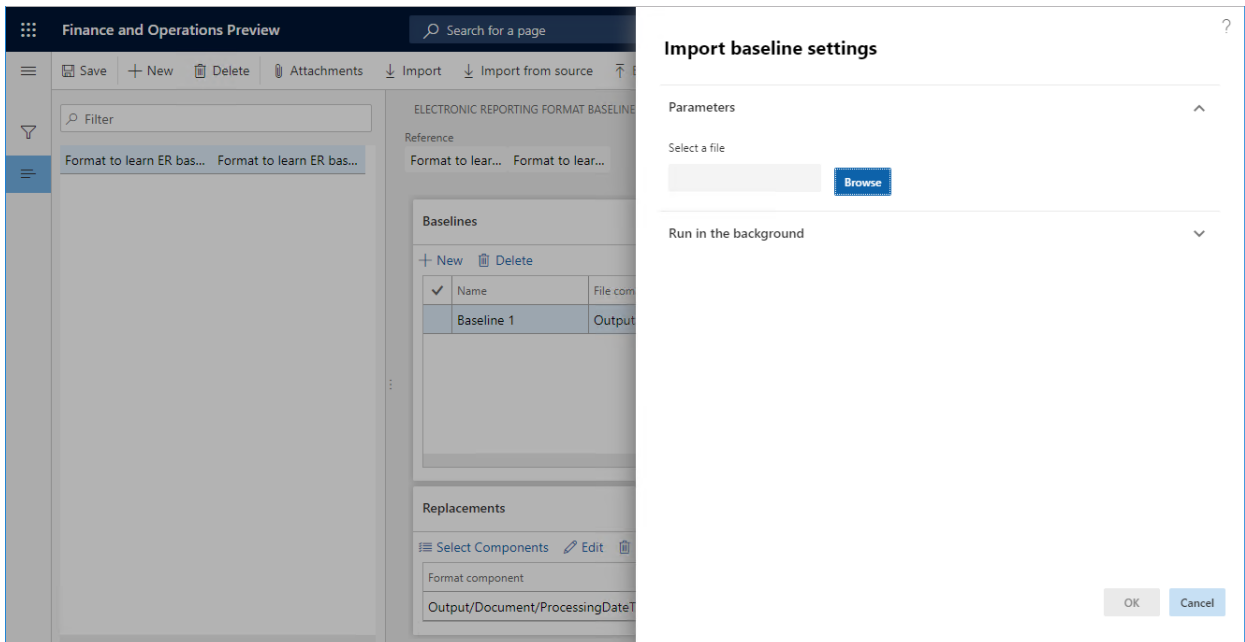
The new ER capabilities let you export baseline settings for the selected ER format from the current environment and store them as XML files.

To export baseline settings, on the **Electronic reporting format baselines** page, select **Export**.

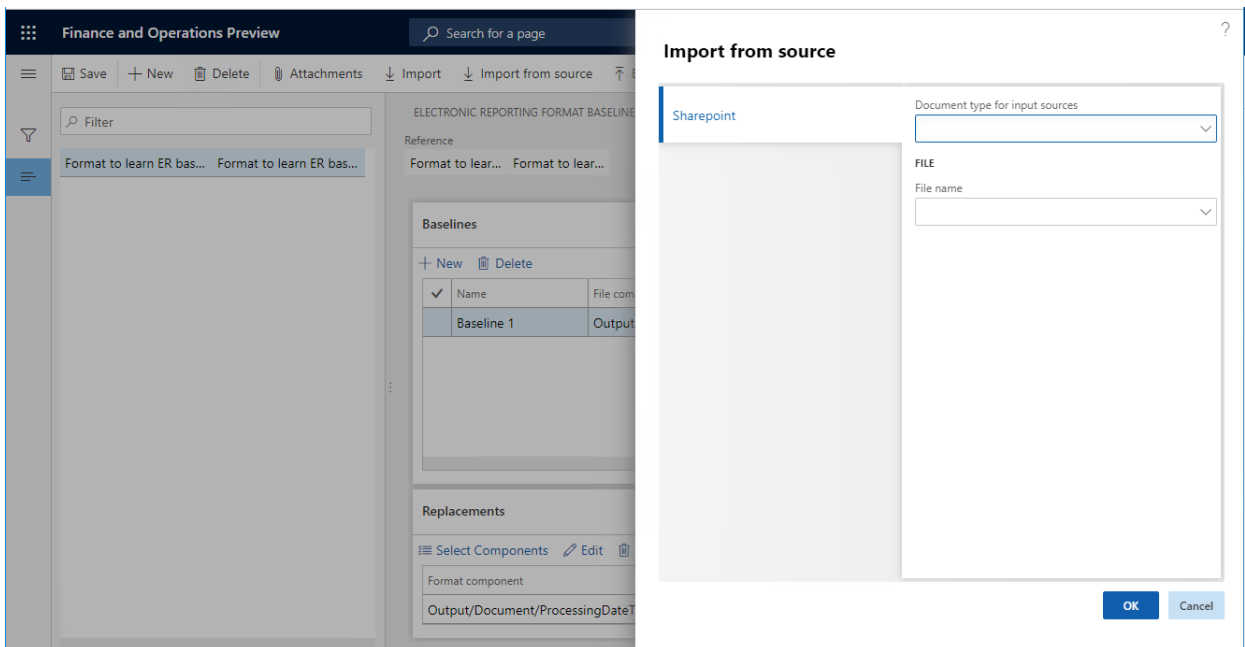
## Import baseline settings

Exported baseline settings can be imported into a different environment. The environment must first be imported as an ER format. You can then import the baseline settings.

To import baseline settings from a locally stored XML file, on the **Electronic reporting format baselines** page, select **Import**, and then select **Browse** to select the XML file.



To import baseline settings from an XML file that is stored on the Microsoft SharePoint Server, based on the current Document management settings and the selected document type, on the **Electronic reporting format baselines** page, select **Import from source**. Then select the document type and the XML file. The required document type to access the SharePoint folder must be configured in advance.



#### NOTE

You can use Task recorder to record the steps for selecting the required document type and the file name in the **Import from source** dialog box. In this way, you can keep required baseline settings on SharePoint Server and then automatically import them by playing a task recording when you run automated tests by using the Regression Suite Automation Tool.

## Additional resources

- [Trace generated report results and compare them with baseline values](#)
- [Task recorder resources](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Automate testing with Electronic reporting

2/18/2021 • 13 minutes to read • [Edit Online](#)

This topic explains how you can use the Electronic reporting (ER) framework to automate testing of some functionality. The example in this topic shows how to automate the testing of vendor payment processing.

The application uses the ER framework to generate payment files and corresponding documents during vendor payment processing. The ER framework consists of a data model, model mappings, and format components that support payment processing for different payment types and the generation of documents in different formats. These components can be downloaded from Microsoft Dynamics Lifecycle Services (LCS) and imported into the instance.

You also can customize each Microsoft component and use it as the basis of your own custom component. By creating a custom version, you can make changes that support specific requirements. For example, you can adjust the ER data model and ER model mapping to access customer-specific application data, or you can change an ER format to modify the layout of a generated document.

You can use customized ER formats to process payment files that generate vendor payments and also to process control reports. Versioning is supported in ER components. Therefore, Microsoft can provide updated versions of ER solutions for vendor payment processing, and you can automatically merge the updated version with your customized component by rebasing it. However, you must test the rebased version to make sure that it works as you expect.

ER data models and ER model mappings are common for many ER formats that are used to process payments of different types and to generate country/region-specific payment documents. Therefore, it's highly desirable to automate user acceptance and integration testing so that it's automatically done in multiple companies but considers the country/region context of each target company, uses different datasets, and so on.

For more information about how to create a custom version of a format that is based on the format that you received from a configuration provider, see [ER Upgrade your format by adopting a new, base version of that format](#).

## Key concepts

Functional power users can author user acceptance and integration testing without having to write source code.

- Use the ER baseline feature to compare generated documents to master copies. For more information, see [Trace generated report results and compare them with baseline values](#).
- Use Task recorder to record test cases, and include baseline assessment. For more information, see [Task recorder resources](#).
- Group test cases for required test scenarios. For more information, see [Create and automate user acceptance tests](#).
  - Use Business process modeler (BPM) in LCS to make libraries for user acceptance tests.
  - Use BPM test libraries to create a test plan and test suites in Microsoft Azure DevOps Services (Azure DevOps).

Functional power users can run user acceptance and integration tests.

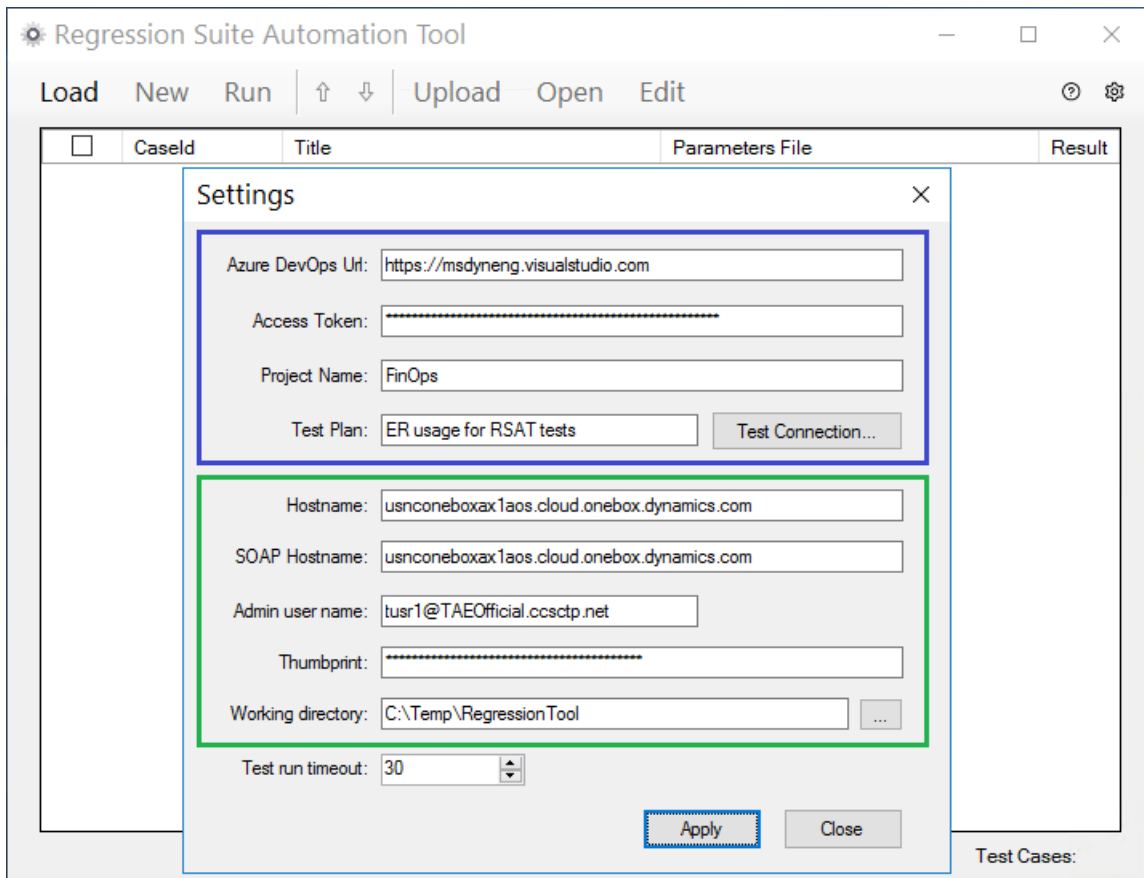
- Use Regression suite automation tool (RSAT) to run test cases of the desired test suite.
- Report the results of the testing to Azure DevOps, and use this service to investigate those results.



# Prerequisites

Before you can complete the tasks in this topic, you must complete the following prerequisites:

- Deploy a topology that supports test automation. You must have access to the instance of this topology for the **System administrator** role. This topology must contain the demo data that will be used in this example. For more information, see [Deploy and use an environment that supports continuous build and test automation](#).
- To run user acceptance and integration tests automatically, you must install RSAT in the topology that you're using and configure it in the appropriate manner. For information about how to install and configure RSAT and configure it to work with Finance and Operations apps and Azure DevOps, see [Regression Suite Automation Tool](#). Pay attention to the prerequisites for using the tool. The following illustration shows an example of the RSAT settings. The blue rectangle encloses the parameters that specify access to Azure DevOps. The green rectangle encloses the parameters that specify access to the instance.

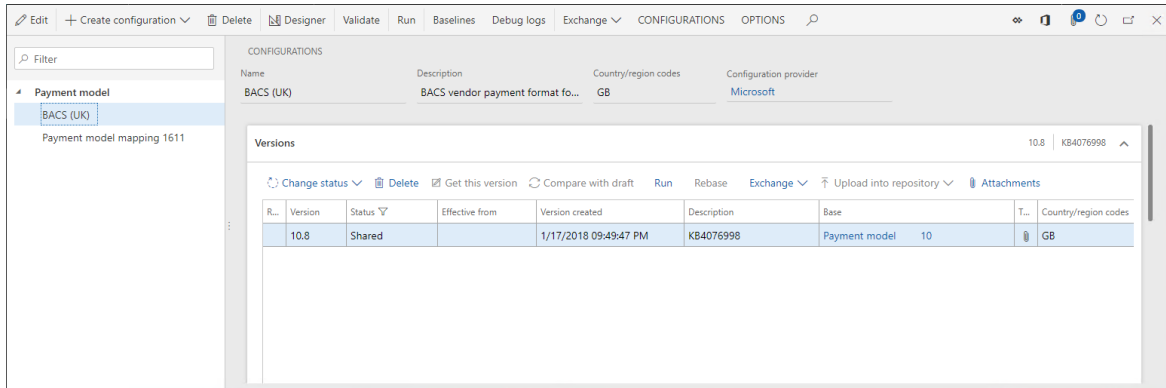


- To organize test cases in suites to help guarantee the correct execution sequence, so that you can collect logs of test executions for further reporting and investigation, you must have access to Azure DevOps from the deployed topology.
- To complete the example in this topic, we recommend that you download [ER usage for RSAT tests](#). This zip file contains the following task guides:

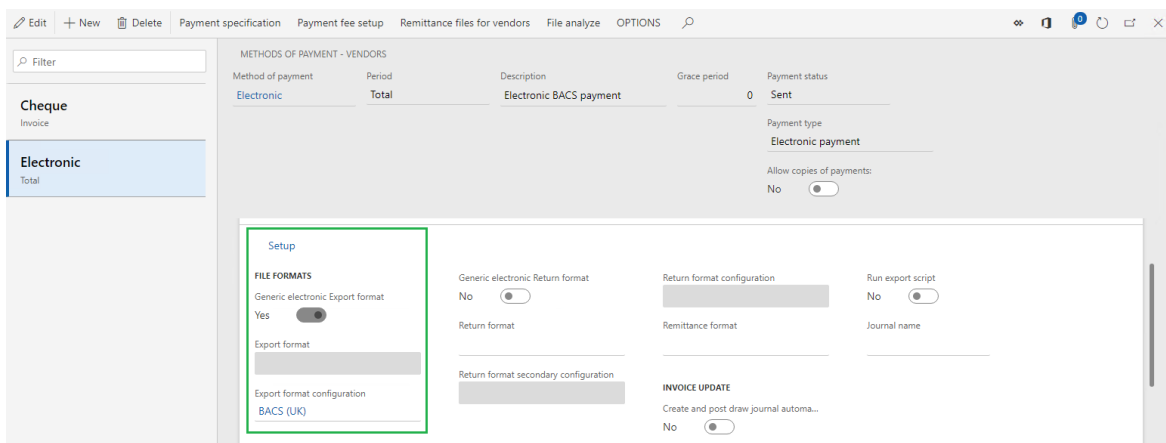
CONTENT	FILE NAME AND LOCATION
Sample task recording to prepare data for testing	Prepare\Recording.xml
Sample task recording to process vendor payment	Process\Recording.xml

# Prepare the Accounts payable module to process vendor payments

1. Sign in to your instance.
2. Download the following ER configurations from LCS. For instructions, see [ER Import a configuration from Lifecycle Services](#).
  - **Payment model** ER model configuration
  - **Payment model mapping 1611** ER model mapping configuration
  - **BACS (UK)** ER format configuration



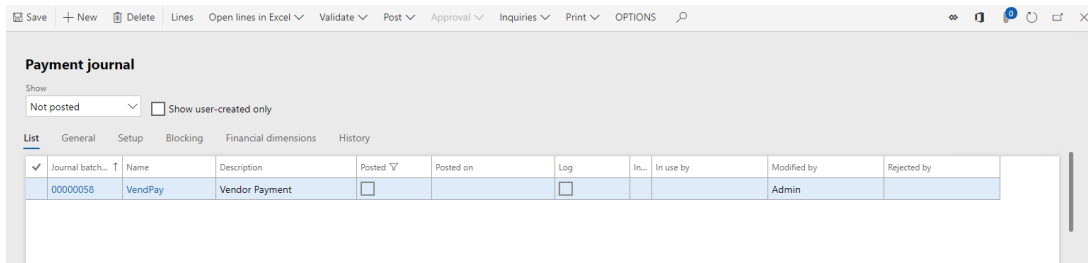
3. Select the GBSI demo data company, which has a country/region context in Great Britain.
4. Configure Accounts payable parameters:
  - a. Go to **Accounts payable > Payment setup > Methods of payment**.
  - b. Select the **Electronic** method of payment.
  - c. Configure the selected method of payment so that it uses the **BACS (UK)** ER format that you downloaded earlier for vendor payment processing:
    - a. On the **File formats** FastTab, set the **Generic electronic Export format** option to **Yes**.
    - b. In the **Export format configuration** field, select **BACS (UK)**.



## NOTE

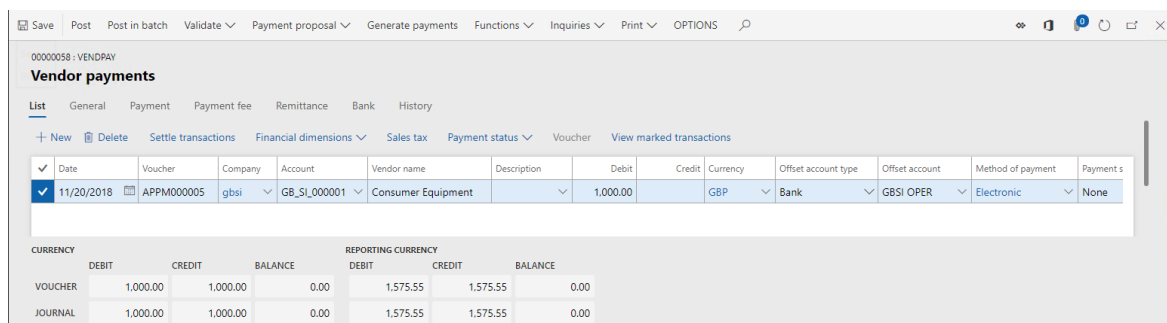
If you have the derived version of this ER format that was created to support customizations, you can select this configuration in the **Electronic** method of payment.

5. Create an example vendor payment:
  - a. Go to **Accounts payable > Payments > Payment journal**.
  - b. Make sure that you haven't posted the payment journal.



c. Select Lines, and enter a line that has the following information.

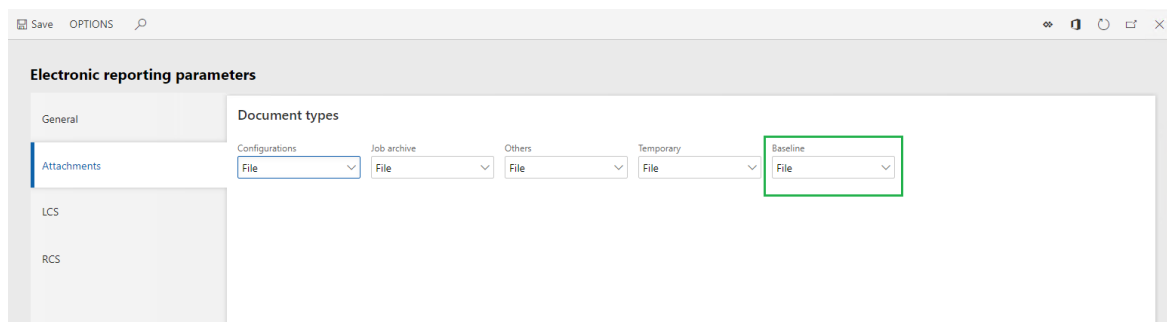
FIELD	EXAMPLE VALUE
Vendor name	GB_SI_000001
Debit	1,000.00
Currency	GBP
Offset account type	Bank
Offset account	GBSI OPER
Method of payment	Electronic



## Prepare the ER framework to test vendor payment processing

### Configure ER parameters

1. Go to **Organization administration > Electronic reporting > Electronic reporting parameters**.
2. On the **Attachments** tab, in the **Baseline** field, select **File** as the document type that the Document management (DM) framework uses to keep documents that are related to the baseline feature as DM attachments.

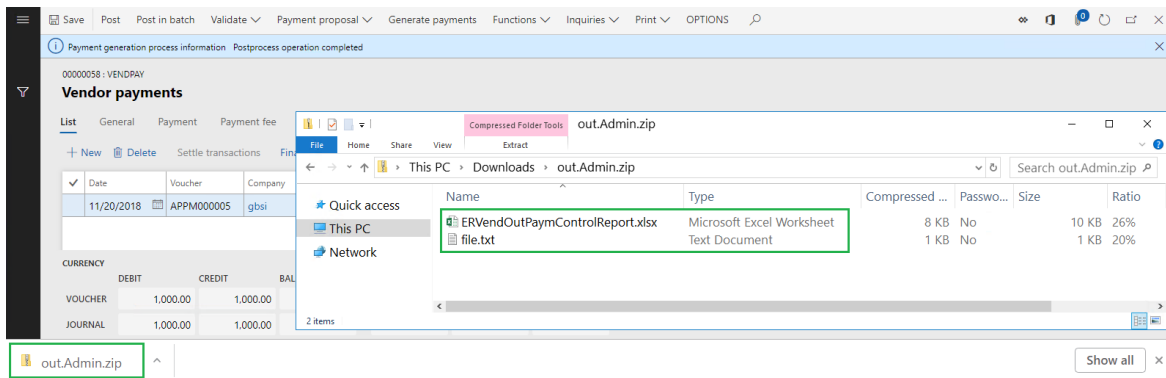


### Generate baseline copies of vendor payment-related documents

1. Go to **Accounts payable > Payments > Payment journal**.

2. Select **Lines**.
3. Select **Generate payments**.
4. Select the **Electronic** method of payment.
5. Select the **GBSI OPER** bank account.
6. Set the **Print control report** option to **Yes**.
7. Download the generated output as a zip file.
8. Open the downloaded file.
9. Extract following files from the downloaded file:

- **File** payment file in text format
- **ERVendOutPaymControlReport** control report file in XLSX format



### Turn on the ER baseline feature

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the Action Pane, on the **Configurations** tab, select **User parameters**.
3. Set the **Run in debug mode** option to **Yes**.

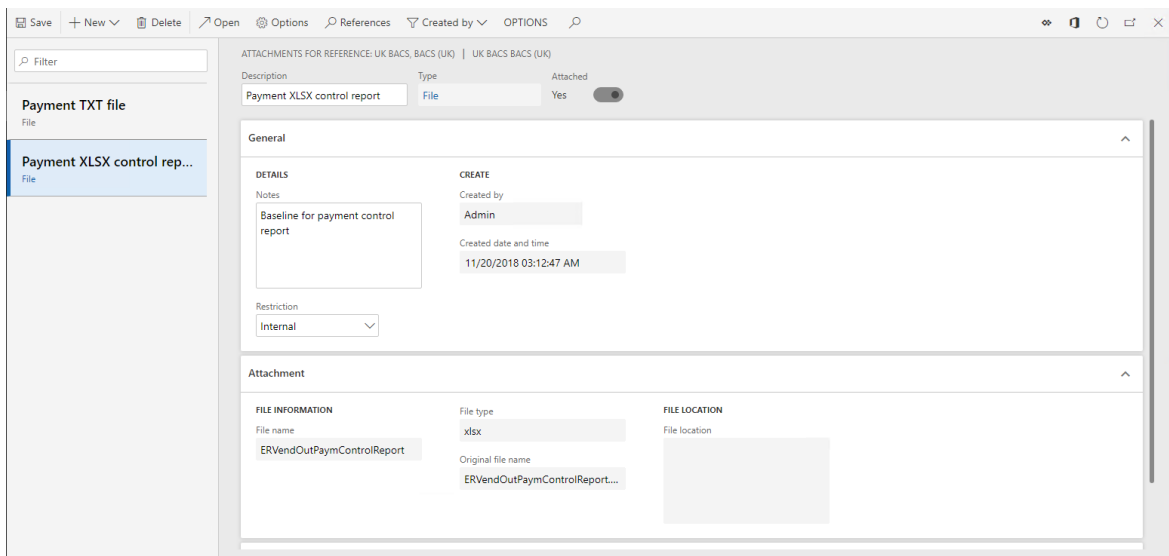
By turning on the **Run in debug mode** parameter, you force the ER framework to perform the following actions after the execution of any ER format that generates outgoing documents:

1. Determine whether a baseline was configured for any of components of the executed ER format.
2. Determine whether each configured baseline is applicable in the current conditions (company code of the signed-in company, file name and file name extension of the generated output, and so on).
3. For each applicable baseline, perform the following actions:
  - a. Compare the output that is generated during execution of the ER format with the corresponding baseline.
  - b. Store the results of the comparison in the ER configurations debug log.

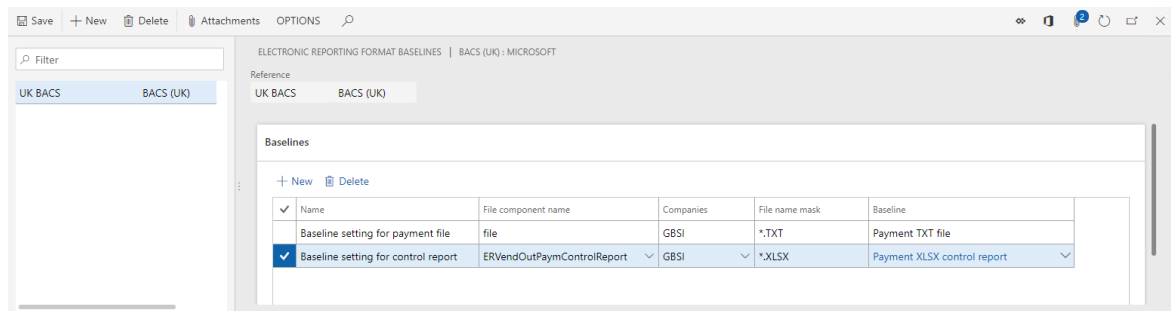
### Configure ER baselines for vendor payment processing

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. Select **Baselines**.
3. Select **New**.
4. In the **Reference** field, select the **BACS (UK)** format.
5. Select **Attachments**.
6. Add a new baseline for the vendor payment file:

- a. Select **New**.
  - b. In the **Type** field, select the **File DM** document type that you configured in the ER parameters to store baseline artifacts.
  - c. Browse to select the locally saved **File** payment file in text format.
  - d. In the **Description** field, enter **Payment TXT file**.
7. Add a new baseline for the control report for the vendor payment:
- a. Select **New**.
  - b. In the **Type** field, select the **File DM** document type that you configured in the ER parameters to store baseline artifacts.
  - c. Browse to select the locally saved **ERVendOutPaymControlReport** control report file in XLSX format.
  - d. In the **Description** field, enter **Payment XLSX control report**.



8. Close the page.
9. On the **Baselines** FastTab, select **New** to configure a baseline for the payment file:
  - a. Name the line **Baseline setting for payment file**.
  - b. In the **File component name** field, select **file** to apply this baseline to the ER format output that generates the payment file in BACS (UK) text format.
  - c. In the **Companies** field, select **GBSI** to apply this baseline when the **BACS (UK)** ER format is run in the GBSI company.
  - d. In **File name mask** field, enter **\*.TXT** to apply this baseline only to outputs of the file format component that have the **.txt** file name extension.
  - e. In the **Baseline** field, select **Payment TXT file** so that this baseline is used for comparison with the generated output.
10. Select **New** to configure a baseline for the control report:
  - a. Name the line **Baseline setting for control report**.
  - b. In the **File component name** field, select **ERVendOutPaymControlReport** to apply this baseline to the ER format output that generates the control report.
  - c. In the **Companies** field, select **GBSI** to apply this baseline when the **BACS (UK)** ER format is run in the GBSI company.
  - d. In **File name mask** field, enter **\*.XLSX** to apply this baseline only to outputs of the **ERVendOutPaymControlReport** format component that have the **.xlsx** file name extension.
  - e. In the **Baseline** field, select **Payment XLSX control report** so that this baseline is used for comparison with the generated output.



## Record tests to validate vendor payment processing

As a functional power user, you can record your own steps to test vendor payment processing. We recommend that you play (and edit, as required) the **Prepare\Recording.xml** task recording that you downloaded earlier. This recording is used to set all testing data to the correct state. That step is required because the testing can be done many times, and every test must use data that is in the same state.

### Reset user settings

1. Open the default dashboard.
2. Select the **Settings** button (the gear symbol).
3. Select **User options**.
4. Select **Usage data**.
5. Select **Reset**.
6. Select **Yes** to confirm that you want to reset usage data.
7. Close the page.

### Record the steps to prepare data for testing

1. Select the **Settings** button (the gear symbol).
2. Select **Task recorder**.
3. Select **Playback recording**.
4. Select **Open from this PC**.
5. Select **Browse**, and select the locally save **Prepare\Recording.xml** file.
6. Select **Start**.
7. Keep selecting **Play next pending step** until all the steps in the recording have been played.

This task recording performs the following actions:

1. Set the status of the processed payment line to **None**.

Prepare data to test processing of vendor's electronic payment

Dynamics 365 Finance and Operations

Contoso Consulting GB

November 2018

Bank management, My business processes, Benefits, Optimization advisor, Budget planning, Outbound work monitoring, Business processes for human resources, Outbound work planning

Work items assigned to me

Cash advance request: Record returned  
Expense reports: Record returned  
Catalog: catalog approval  
Expense reports: Expense report approval

Task recorder

PLAYBACK CONTROLS

- Play next pending step
- ▷ Play to selected step
- ▷ Play all pending steps

STEPS

0 steps recorded / 15 steps pending

1	Go to Accounts payable > Payments > Payment journal.
2	Click Lines.
3	Click Payment status.
4	Click None.
5	Close the page.
6	Close the page.
7	Go to Organization administration > Electronic

2. Turn on the Run in debug mode ER user parameter.

Prepare data to test processing of vendor's electronic payment

Dynamics 365 Finance and Operations

Contoso Consulting GB

November 2018

Bank management, My business processes, Benefits, Optimization advisor, Budget planning, Outbound work monitoring, Business processes for human resources, Outbound work planning

Work items assigned to me

Cash advance request: Record returned  
Expense reports: Record returned  
Catalog: catalog approval  
Expense reports: Expense report approval

Task recorder

PLAYBACK CONTROLS

- Play next pending step
- ▷ Play to selected step
- ▷ Play all pending steps

STEPS

0 steps recorded / 15 steps pending

7	Go to Organization administration > Electronic
8	On the Action Pane, click Configurations.
9	Click User parameters.
10	Select Yes in the Run in debug mode field.
11	Click OK.
12	Close the page.
13	Go to Organization administration > Electronic

3. Clean up the ER debug log that contains the results of the comparison of generated files to baselines.

Prepare data to test processing of vendor's electronic payment

Dynamics 365 Finance and Operations

Contoso Consulting GB

November 2018

Bank management, My business processes, Benefits, Optimization advisor, Budget planning, Outbound work monitoring, Business processes for human resources, Outbound work planning

Work items assigned to me

Cash advance request: Record returned  
Expense reports: Record returned  
Catalog: catalog approval  
Expense reports: Expense report approval

Task recorder

PLAYBACK CONTROLS

- Play next pending step
- ▷ Play to selected step
- ▷ Play all pending steps

STEPS

0 steps recorded / 15 steps pending

9	Click User parameters.
10	Select Yes in the Run in debug mode field.
11	Click OK.
12	Close the page.
13	Go to Organization administration > Electronic
14	Click Delete all.
15	Click Yes.

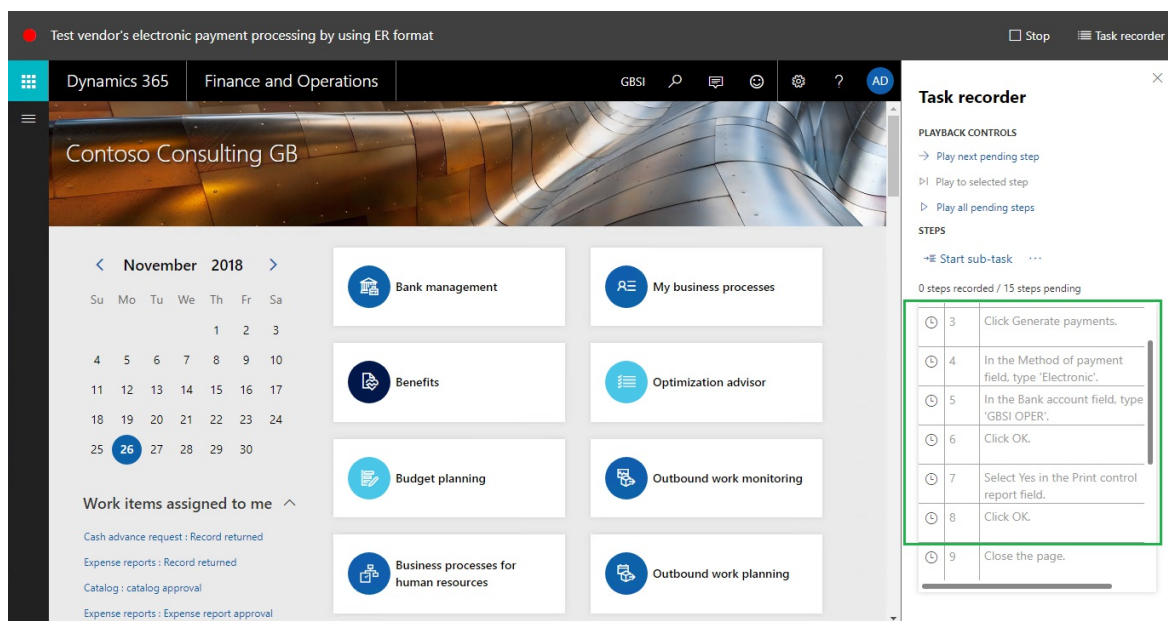
## Record the steps to test vendor payment processing

We recommend that you play (and edit, as required) the **Process\Recording.xml** task recording that you downloaded earlier. This recording is used to process vendor payments and validate the results of the comparison of generated documents to corresponding baselines.

1. Select the **Settings** button (the gear symbol).
2. Select **Task recorder**.
3. Select **Playback recording**.
4. Select **Open from this PC**.
5. Select **Browse**, and select the locally saved **Process\Recording.xml** file.
6. Select **Start**.
7. Keep selecting **Play next pending step** until all the steps in the recording have been played.

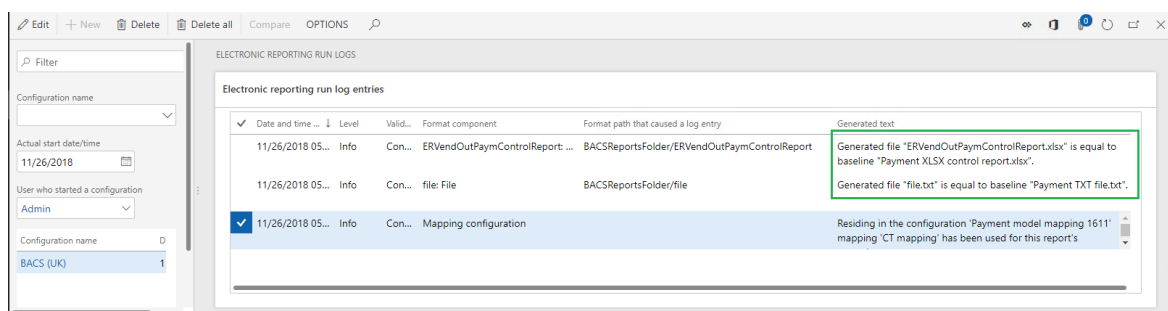
This task recording performs the following actions:

1. Start vendor payment processing.
2. Select the correct runtime parameters, and turn on generation of a control report.



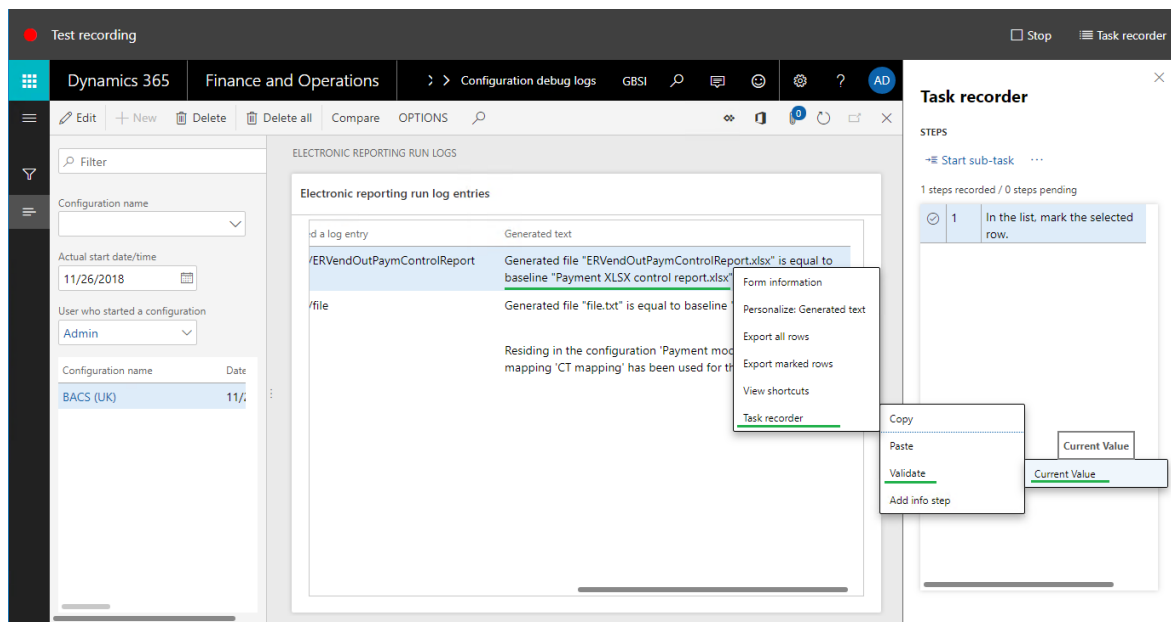
3. Access the ER debug log to record the results of the comparison of generated outputs to corresponding baselines.

In the ER debug log, the results of the comparison appear in the **Generated text** field. The **Format component** and **Format path that caused a log entry** fields refer to the file component for which the generated output has been compared to the baseline.

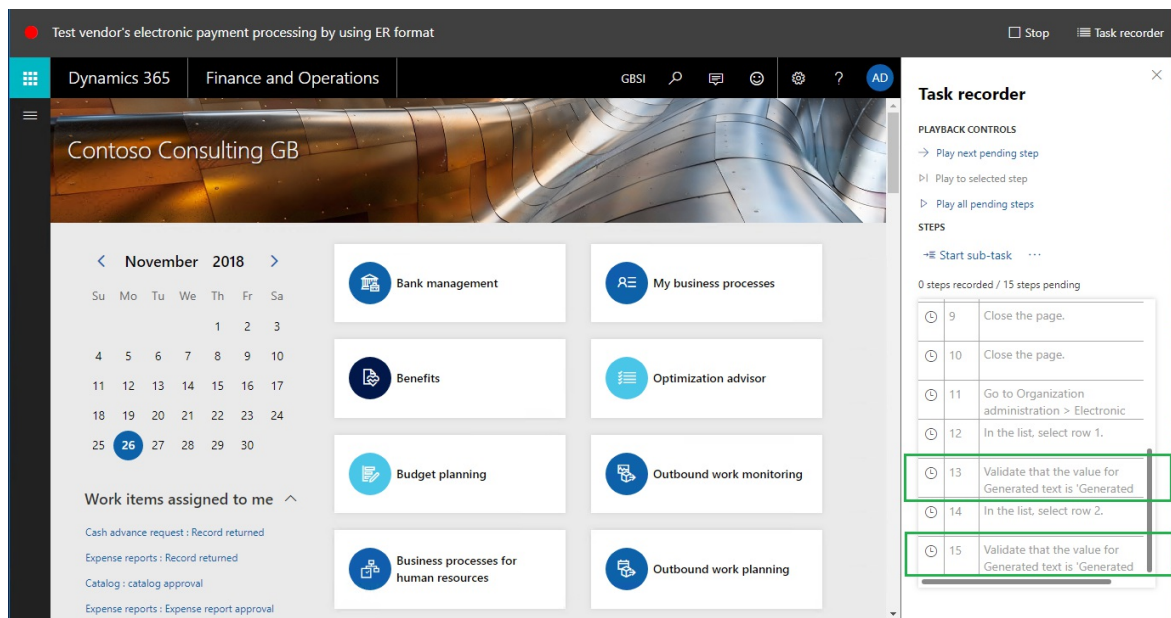


4. The comparison of the current output to the baseline is recorded by using the **Validate** Task recorder option and selecting **Current Value**.



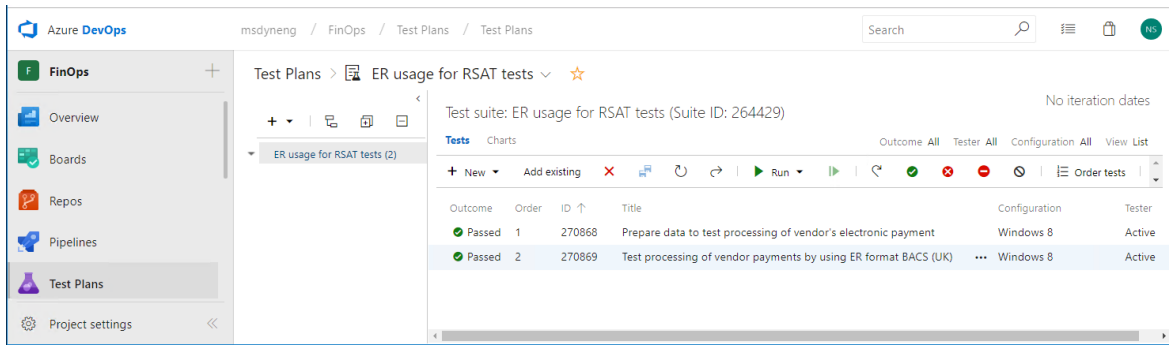


The following illustration shows what the recorded validation steps look like in the task recording.



## Add the recorded tests to Azure DevOps

1. Open the Azure DevOps environment.
2. Select the project that you defined in the RSAT parameters when you [configured the tool](#).
3. Select the test plan that you defined in the RSAT parameters when you [configured the tool](#).
4. Create a new test case for the selected test plan:
  - a. Name the test case **Prepare data to test processing of vendor's electronic payment**.
  - b. Attach the **Recording.xml** file from the **Prepare** folder that you downloaded earlier.
5. Create a new test case for the selected test plan:
  - a. Name the test case **Test processing of vendor payments by using ER format BACS (UK)**.
  - b. Attach the **Recording.xml** file from the **Process** folder that you downloaded earlier.



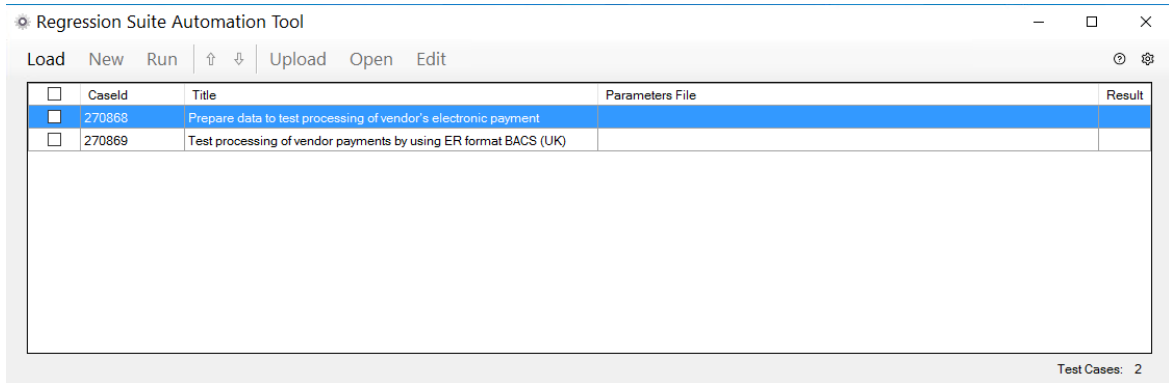
**NOTE**

Pay attention to the correct execution order of the tests that are added.

## Prepare RSAT to run the recorded tests

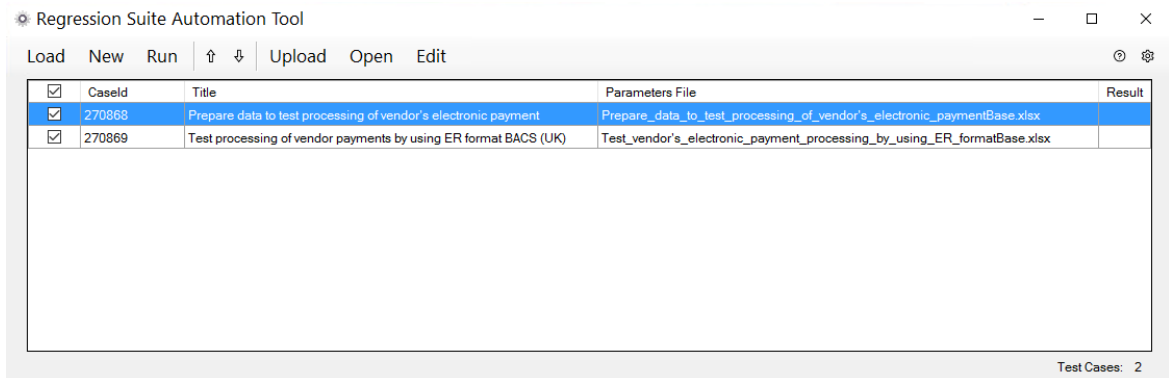
### Load the tests from Azure DevOps to RSAT

1. Open the local RSAT application in the current topology.
2. Select **Load** to load the tests that currently reside in Azure DevOps into RSAT.



### Create automation and parameters files

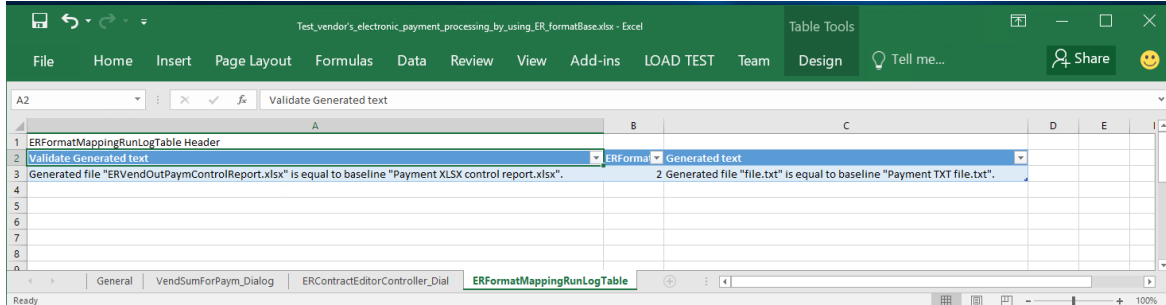
1. In RSAT, select the tests that you loaded from Azure DevOps.
2. Select **New** to create RSAT automation and parameters files.



### Modify the parameters files

1. In RSAT, select the **Prepare data to test processing of vendor's electronic payment** test case.
2. Select **Edit**.
3. In the Microsoft Excel workbook that is opened, on the **General** worksheet, change the company code to **GBSI**, because this company will be used for test execution.

- In RSAT, select the **Test processing of vendor payments by using ER format BACS (UK)** test case.
- Select **Edit**.
- In the Excel workbook that is opened, on the **General** worksheet, change the company code to **GBSI**.
- On the **ERFormatMappingRunLogTable** worksheet, notice that cells A:3 and C:3 contain the text of the fields in the ER debug log table that are used to validate the results of the comparison of the output to the baseline. These texts will be used to evaluate ER debug log records that are created during test execution.



## Run the tests and analyze the results

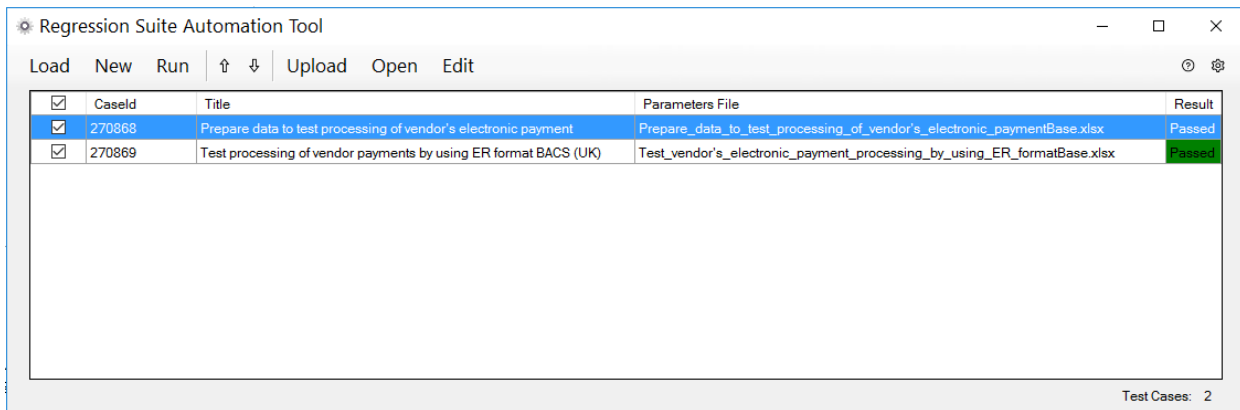
### Run the tests in RSAT

- In RSAT, select the loaded tests.
- Select **Run**.

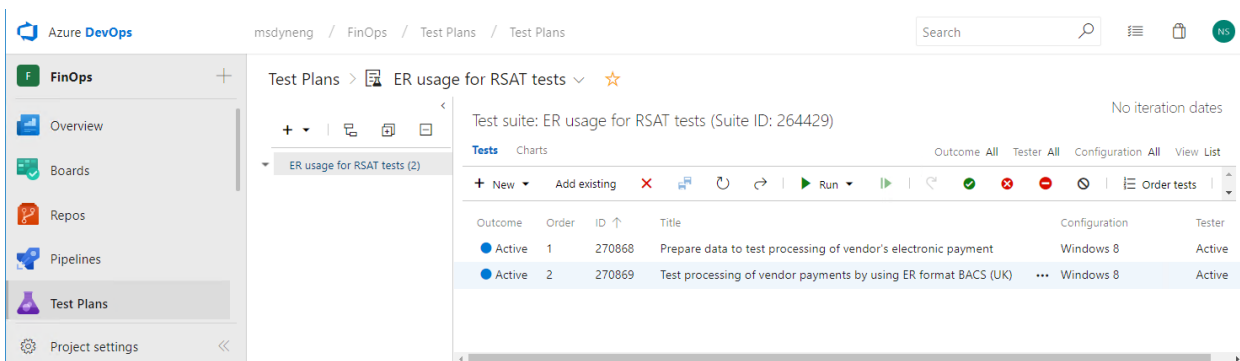
Notice that test cases are automatically run in the application by using a web browser.

### Analyze the results of test execution

The results of the test execution are stored in RSAT. Notice that both tests were passed.



Notice that the results of the test execution are also sent to Azure DevOps so that you can do further analysis.



### Simulate a situation where tests fail

This test suite must fail when at least one of the generated outputs doesn't match the corresponding baseline. To

achieve this situation, you can use your derived version of the **BACS (UK)** format that will generate a payment file that has different content than the corresponding baseline. To simulate this situation, you can use the same **BACS (UK)** format but change the payment amount on the processed payment line.

1. Open the application and go to **Accounts payable > Payments > Payment journal**.
2. Select **Lines**.
3. Select the payment line, and then select **Payment status > None**.
4. In the **Debit** field, change the value from **1,000.00** to **2,000.00**.
5. Select **Save** to save your changes.

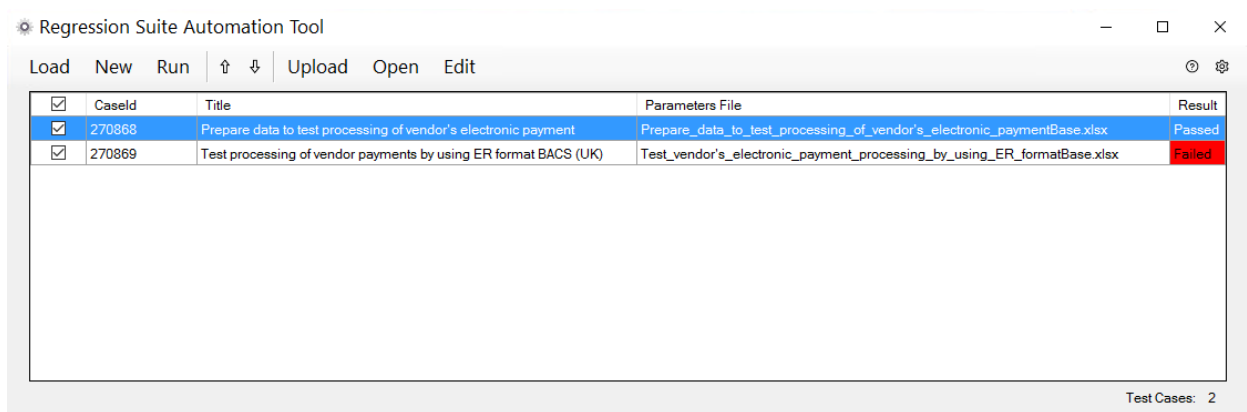
### Run the tests in RSAT

1. In RSAT, select the loaded tests.
2. Select **Run**.

Notice that test cases are automatically run in the application by using a web browser.

### Analyze the results of test execution

The results of the test execution are stored in RSAT. Notice that the second test failed during the second execution.

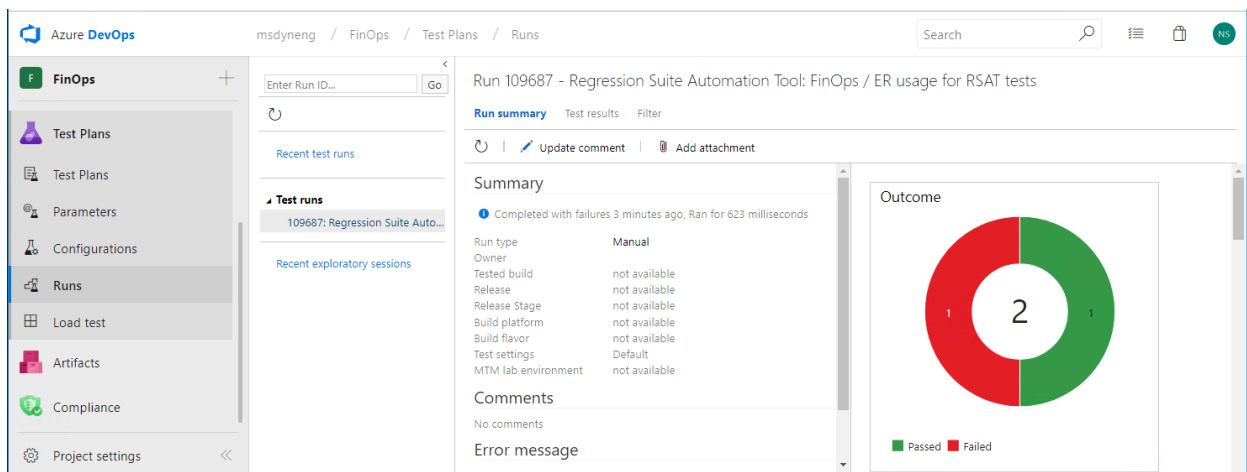


The screenshot shows the Regression Suite Automation Tool (RSAT) interface. At the top, there are menu options: Load, New, Run, Upload, Open, and Edit. Below the menu is a table with the following columns: Caselid, Title, Parameters File, and Result. The table contains two rows of test results.

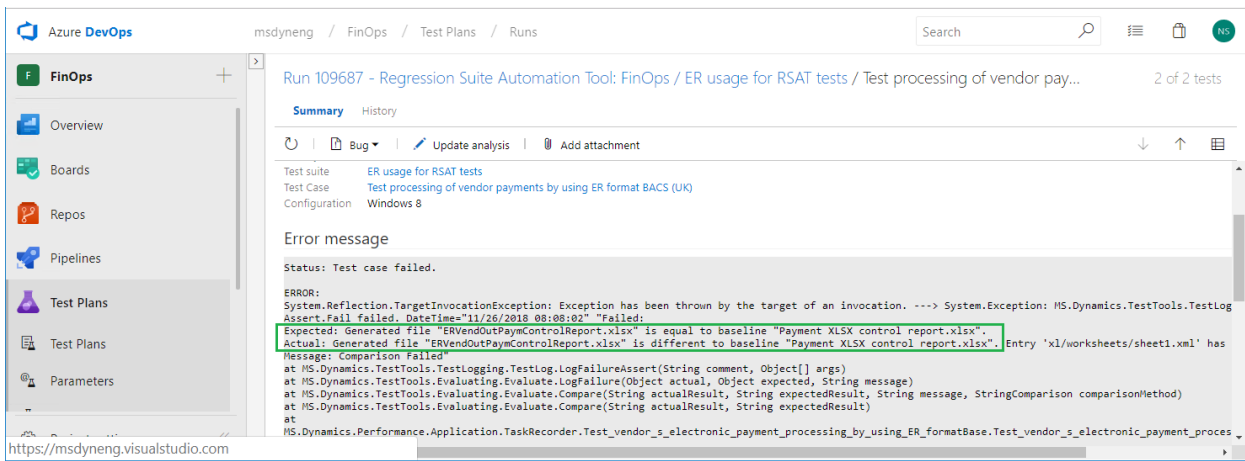
✓	Caselid	Title	Parameters File	Result
✓	270868	Prepare data to test processing of vendor's electronic payment	Prepare_data_to_test_processing_of_vendor's_electronic_paymentBase.xlsx	Passed
✓	270869	Test processing of vendor payments by using ER format BACS (UK)	Test_vendor's_electronic_payment_processing_by_using_ER_formatBase.xlsx	Failed

At the bottom right of the window, it says "Test Cases: 2".

Notice that the results of the test execution are also sent to Azure DevOps so that you can do further analysis.



You can access the status of each test. You can also access the execution log so that you analyze the reasons for any failure. In the following illustration, the execution log shows that the failure occurred because of the difference in content between the generated payment file and its baseline.



Therefore, as you've seen, the functioning of any ER format can be evaluated automatically by using RSAT as the testing platform and by using Task recorder-based test cases that use the ER baseline feature.

## Additional resources

- [Task recorder resources](#)
- [Regression suite automation tool](#)
- [Create and automate user acceptance tests](#)
- [Deploy and use an environment that supports continuous build and test automation](#)
- [Trace generated report results and compare them with baseline values](#)
- [ER Upgrade your format by adopting a new, base version of that format](#)
- [ER Import a configuration from Lifecycle Services](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure Electronic reporting (ER) to pull data into Power BI

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains how you can use your Electronic reporting (ER) configuration to arrange the transfer of data from your instance to Power BI services. As an example, this topic uses Intrastat transactions as business data that must be transferred. The Power BI map visualization uses this Intrastat transaction data to present a view for analysis of company import/export activities on the Power BI report.

## Overview

Microsoft Power BI is a collection of software services, apps, and connectors that work together to turn external sources of data into coherent, visually immersive, and interactive insights. Electronic reporting (ER) lets users easily configure data sources and arrange the transfer of data from the application to Power BI. The data is transferred as files in the OpenXML worksheet (Microsoft Excel workbook file) format. The transferred files are stored on a Microsoft SharePoint Server that has been configured for that purpose. The stored files are used in Power BI to make reports that include visualizations (tables, charts, maps, and so on). Power BI reports are shared with Power BI users, and they are accessed in Power BI dashboards and on the application pages. This topic explains the following tasks:

- Configure Microsoft Dynamics 365 Finance.
- Prepare your ER format configuration to get data from the Finance application.
- Configure the ER environment to transfer data to Power BI.
- Use transferred data to create a Power BI report.
- Make the Power BI report accessible in Finance.

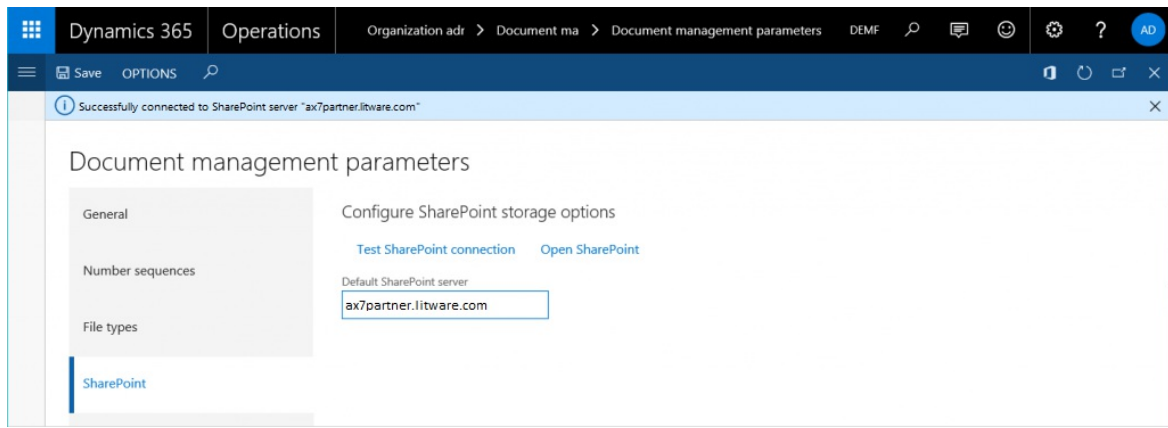
## Prerequisites

To complete the example in this topic, you must have the following access:

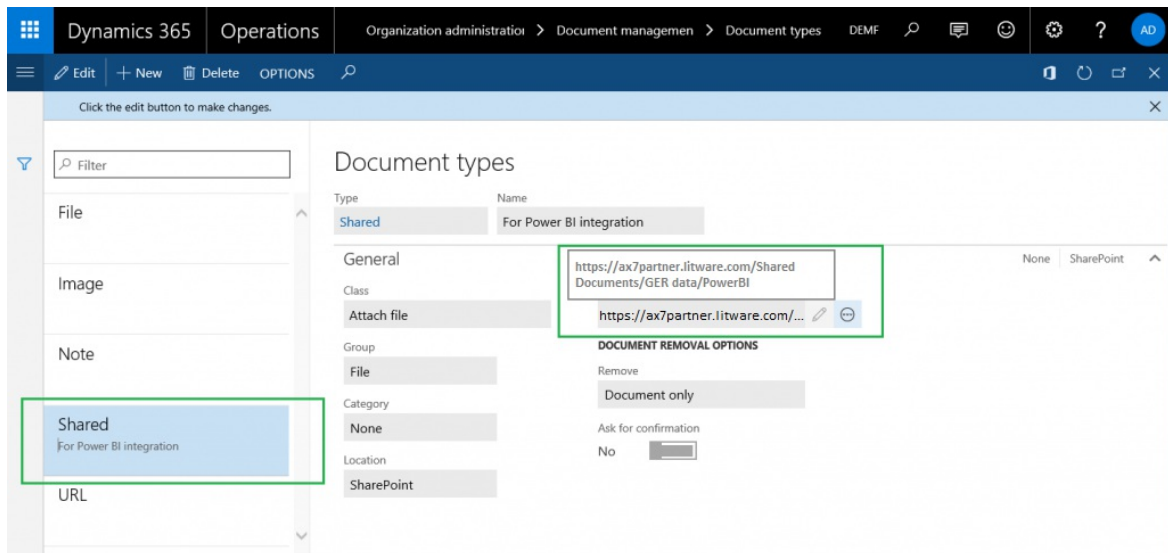
- Access for one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
- Access to the SharePoint Server that is configured for use with the application
- Access to the Power BI framework

## Configure document management parameters

1. On the **Document management parameters** page, configure access to the SharePoint Server that will be used in the company that you're signed in to (the DEMF company in this example).
2. Test the connection to the SharePoint Server to make sure that you've been granted access.



3. Open the configured SharePoint site. Create a new folder where ER will store Excel files that have the business data that the Power BI reports require as a source of Power BI datasets.
4. On the **Document types** page, create a new document type that will be used to access the SharePoint folder that you just created. Enter **File** in the **Group** field and **SharePoint** in the **Location** field, and then enter the address of the SharePoint folder.

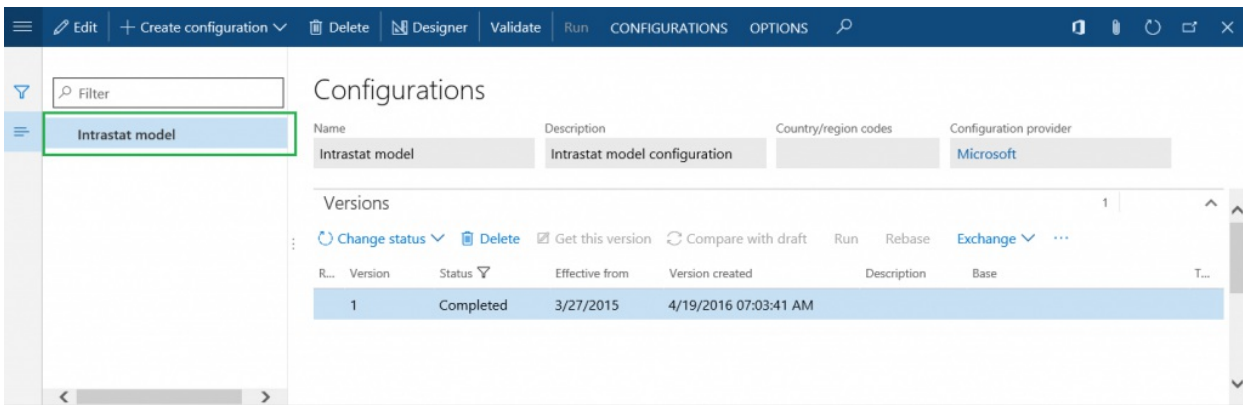


## Configure ER parameters

1. In the **Electronic reporting** workspace, click the **Electronic reporting parameters** link.
2. On the **Attachments** tab, select the **File** document type for all the fields.
3. In the **Electronic reporting** workspace, make the required provider active by clicking **Set active**. For more information, play the **ER Select service provider** task guide.

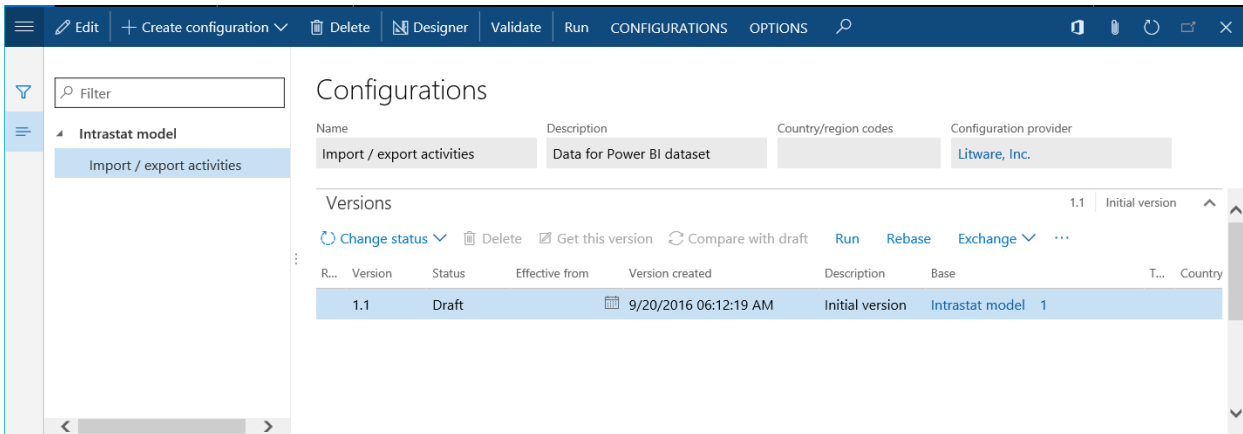
## Use an ER data model as the source of data

You must have an ER data model as the source of business data that will be used on Power BI reports. This data model is uploaded from the ER configurations repository. For more information, see [Download Electronic reporting configurations from Lifecycle Services](#), or play the **ER Import a configuration from Lifecycle Services** task guide. Select **Intrastat** as the data model that will be uploaded from the selected ER configurations repository. (In this example, version 1 of the model is used.) You can then access the **Intrastat ER** model configuration on the **Configurations** page.



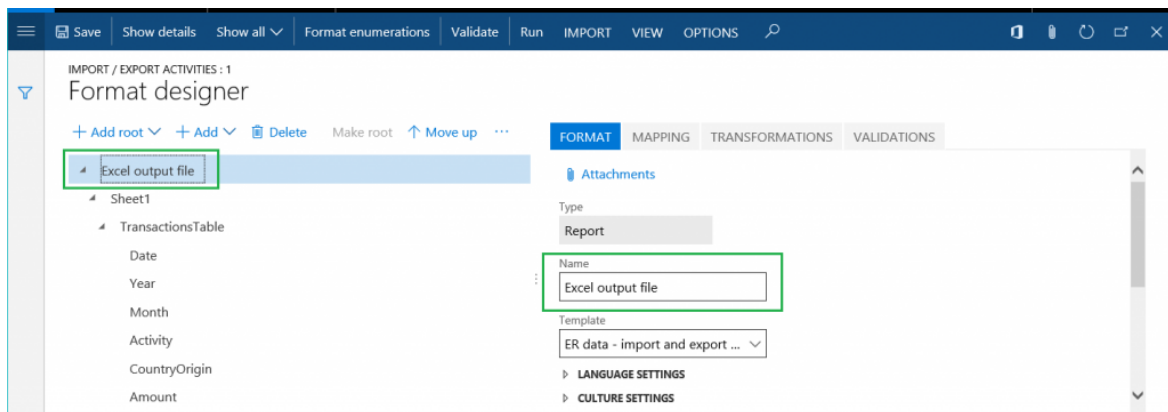
## Design an ER format configuration

You must create a new ER format configuration that uses the **Intrastat** data model as the source of business data. This format configuration must generate output results as electronic documents in OpenXML (Excel file) format. For more information, play the **ER Create a configuration for reports in OPENXML format** task guide. Name the new configuration **Import / export activities**, as shown in the following illustration. Use the [ER data - import and export details](#) Excel file as a template when you design the ER format. (For information about how to import a format template, play the task guide.)



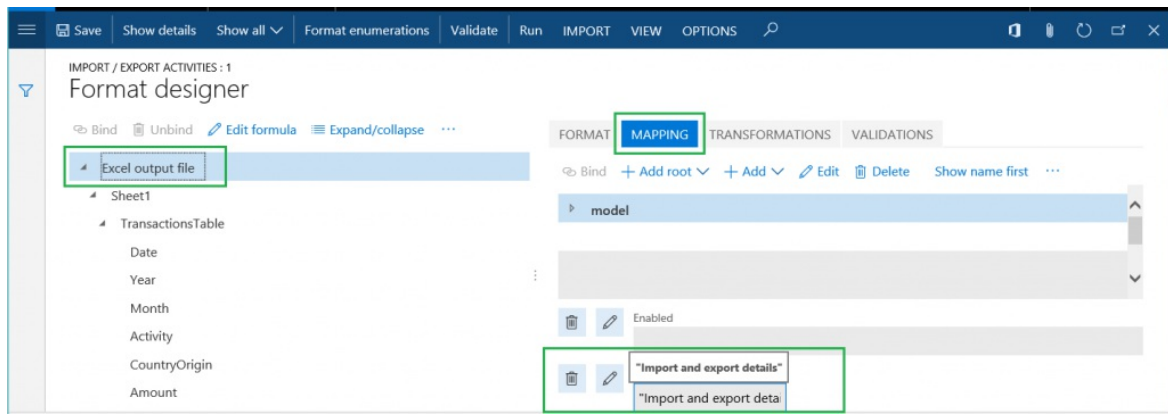
To modify the **Import / export activities** format configuration, follow these steps.

1. Click **Designer**.
2. On the **Format** tab, name the file element for this format **Excel output file**.

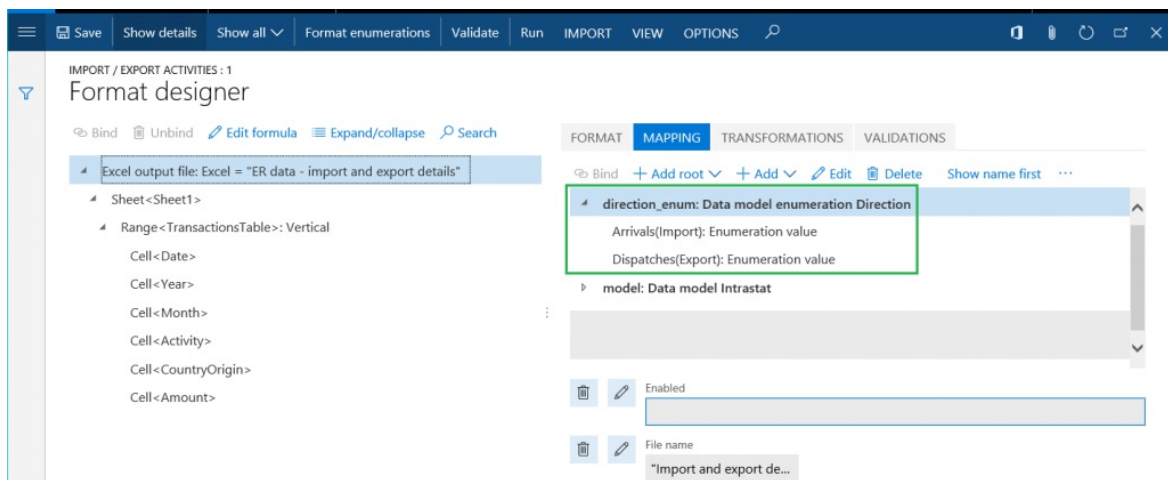


3. On the **Mapping** tab, specify the name of the Excel file that will be generated whenever this format is run. Configure the related expression to return the value **Import and export details** (the .xlsx file name extension will be added automatically).

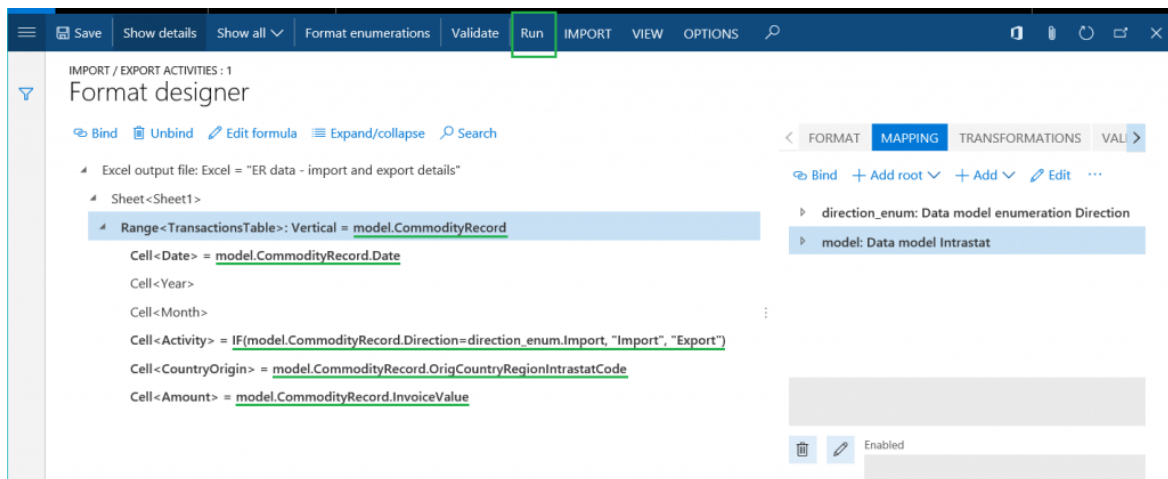




4. Add a new data source item for this format. (This enumeration will be required for further data binding.)
  - a. Name the data source `direction_enum`.
  - b. Select **Data model enumeration** as the data source type.
  - c. Refer to the **Direction** data model enumeration.



5. Complete the binding of elements of the **Intrastat** data model and elements of the designed format, as shown in the following illustration.



After it's run, the ER format generates the output result in Excel format. It sends the details of the Intrastat transactions to the output result, and separates them as transactions that describe either import activities or export activities. Click **Run** to test the new ER format for the list of Intrastat transactions on the **Intrastat** page (**Tax > Declarations > Foreign trade > Intrastat**).

Click the edit button to make changes.

Intrastat

OVERVIEW GENERAL

✓ Date	Direction	Correction	Item number	Category	Commodity	Weight	Invoice amount	Statistical amount	Is...	Country/region of origin
1/31/2016	Arrivals		D0003		920 20 34	4.00	1,000.00	1,000.00	✓	AUT
1/31/2016	Arrivals		T0006		900 22 33	5.00	700.00	700.00	✓	CHE

The following output result is generated. The file is named **Import and export details.xlsx**, as you specified in the format settings.

Import and export details.xlsx [Read-Only] - Excel

A1	Date	Year	Month	Activity	CountryOrigin	Amount
1	31.01.2016	2016	1	Import	AT	1 000,00
2	31.01.2016	2016	1	Import	CH	700,00
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

## Configure the ER destination

You must configure the ER framework to send the output result of the new ER format configuration in a special way.

- The output result must be sent to the folder of the selected SharePoint Server.
- Each execution of the format configuration must create a new version of same Excel file.

On the **Electronic reporting** page (**Organization administration** > **Electronic reporting**), click the **Electronic reporting destination** item, and add a new destination. In the **Reference** field, select the **Import / export activities** format configuration that you created earlier. Follow these steps to add a new file destination record for the reference.

1. In the **Name** field, enter the title of the file destination.
2. In the **File name** field, select the name **Excel output file** for the Excel file format component.

Click the **Settings** button for the new destination record. Then, in the **Destination settings** dialog box, follow these steps.

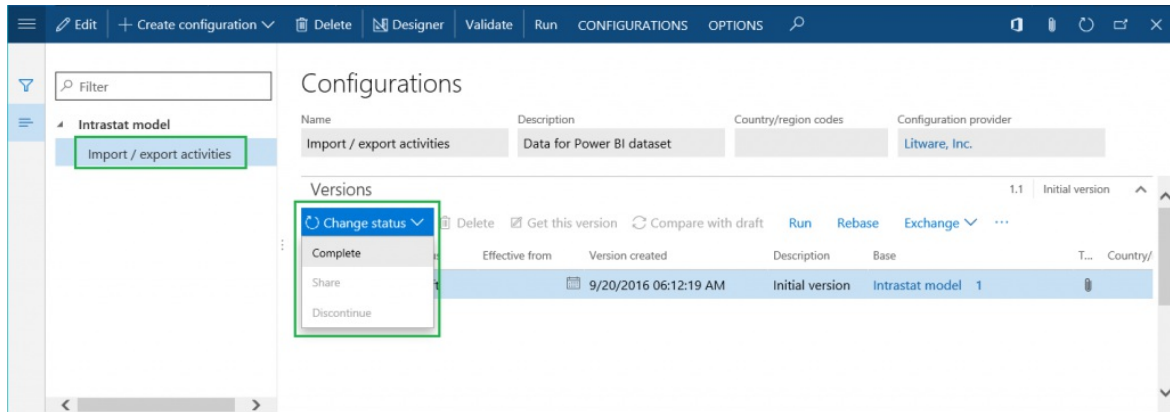
1. On the **Power BI** tab, set the **Enabled** option to **Yes**.
2. In the **SharePoint** field, select the **Shared** document type that you created earlier.

## Schedule execution of the configured ER format

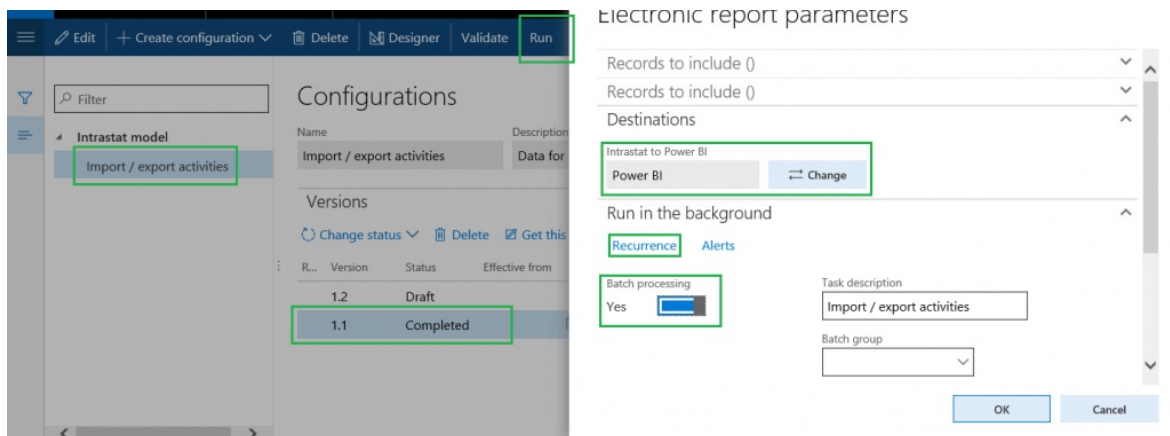
1. On the **Configurations** page (**Organization administration** > **Electronic reporting** >

Configurations), in the configurations tree, select the **Import / export activities** configuration that you created earlier.

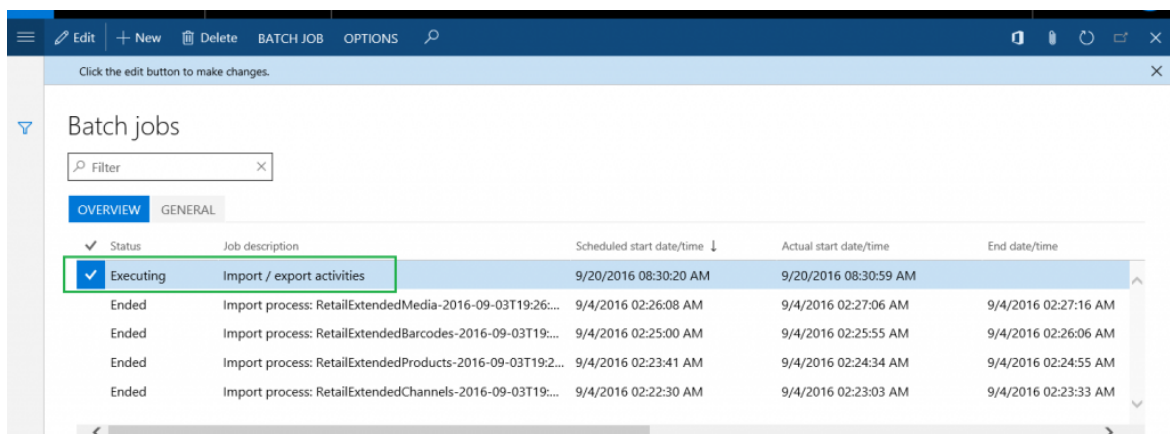
2. Change the status of version 1.1 from **Draft** to **Complete** to make this format available for use.



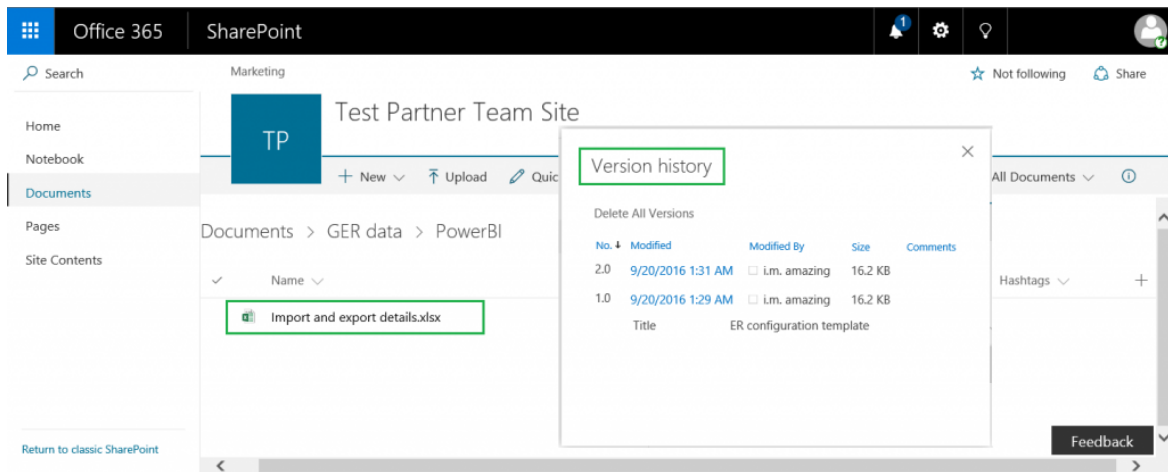
3. Select the completed version of the **Import / export activities** configuration, and then click **Run**. Note that the configured destination is applied to the output result that is generated in Excel format.
4. Set the **Batch processing** option to **Yes** to run this report in unattended mode.
5. Click **Recurrence** to schedule the required recurrence of this batch execution. The recurrence defines how often the updated data will be transferred to Power BI.



6. After it's configured, you can find the ER report execution job on the **Batch jobs** page (System administration > Inquiries > Batch jobs).

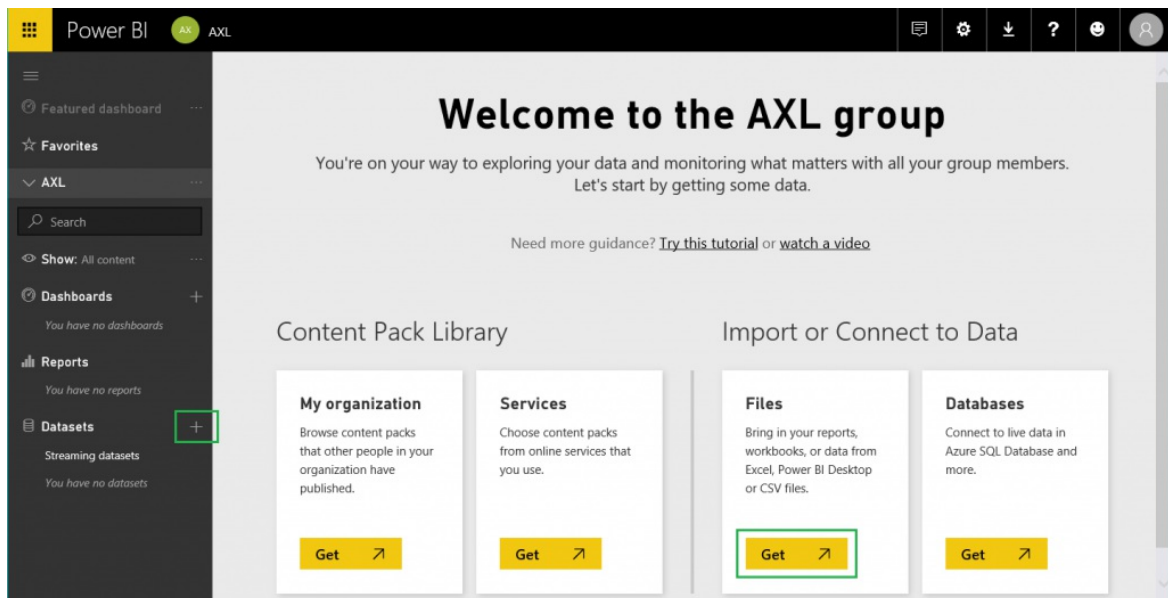


7. When this job is run for the first time, the destination creates a new Excel file that has the configured name in the selected SharePoint folder. Every subsequent time that the job is run, the destination creates a new version of this Excel file.

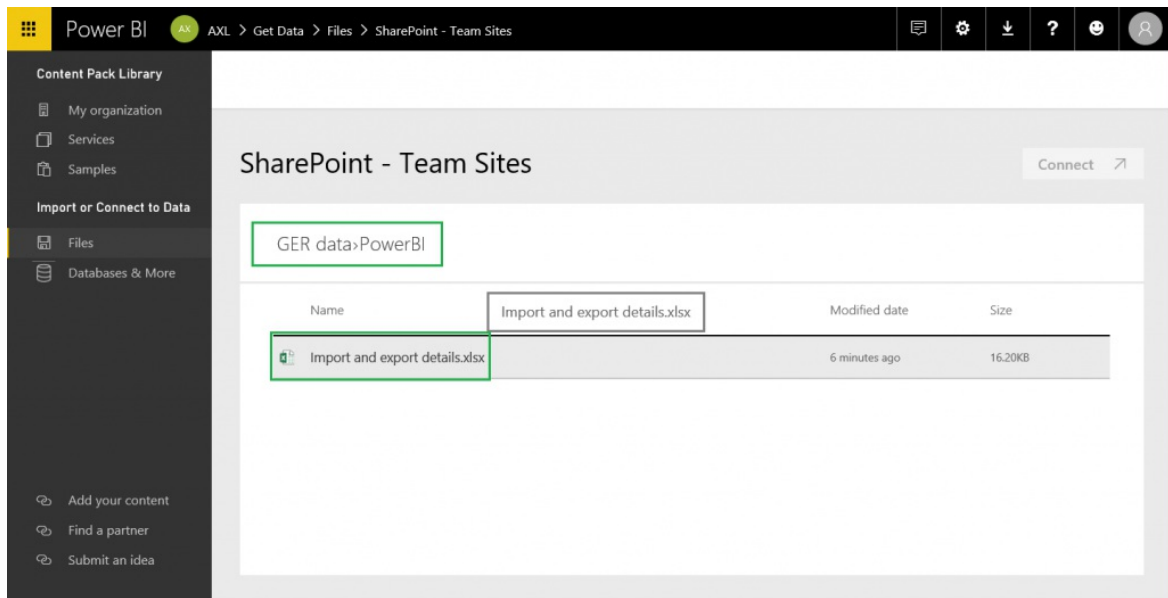


## Create a Power BI dataset by using the output result of the ER format

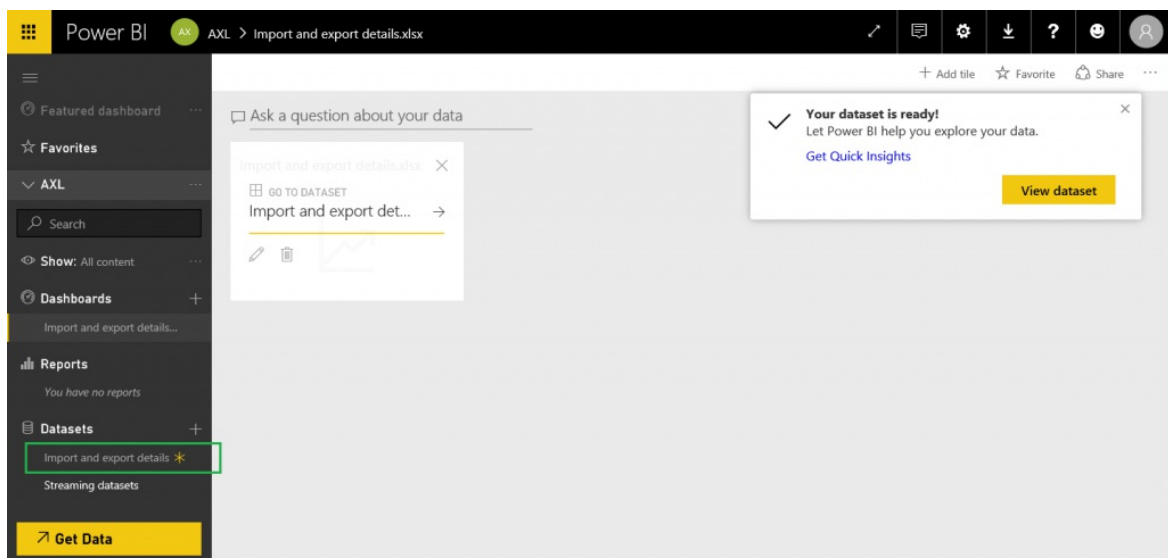
1. Sign in to Power BI, and either open an existing Power BI group (workspace) or create a new group. Either click **Add** under **Files** in the **Import or Connect to Data** section, or click the plus sign (+) next to **Datasets** in the left pane.



2. Select the **SharePoint – Team sites** option, and then enter the path of SharePoint Server that you're using ( `https://ax7partner.litware.com` in our example).
3. Browse to the `/Shared Documents/GER data/PowerBI` folder, and select the Excel file that you created as the source of data for the new Power BI dataset.



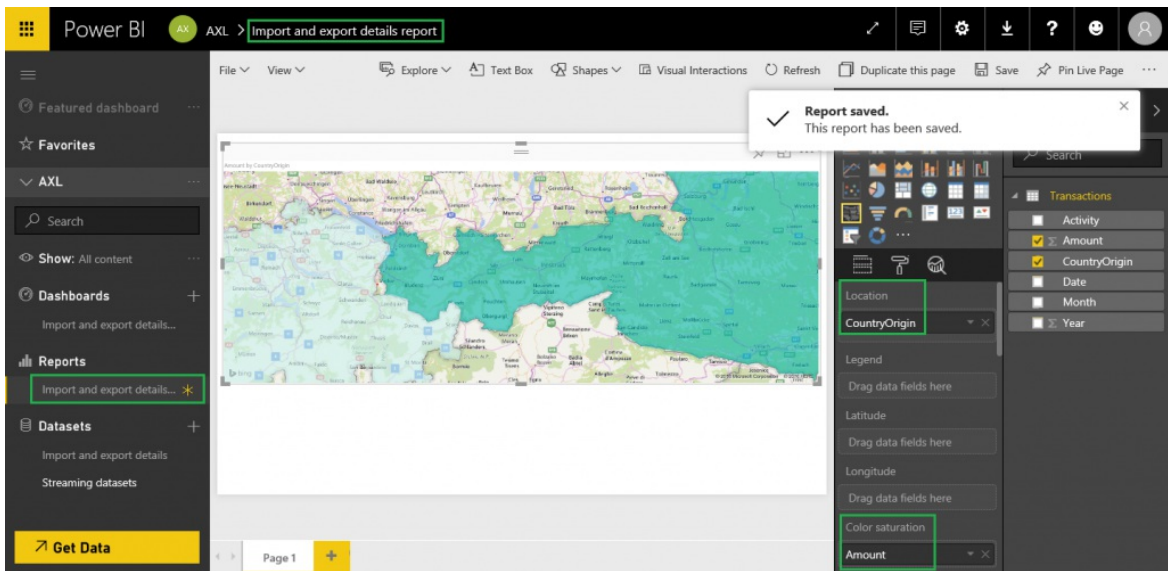
4. Click **Connect**, and then click **Import**. A new dataset is created that is based on the selected Excel file. The dataset can also be added automatically to the newly created dashboard.



5. Configure the refresh schedule for this dataset to force a periodic update. Periodic updates enable the consumption of new business data that comes via periodic execution of the ER report through new versions of the Excel file that are created on the SharePoint Server.

## Create a Power BI report by using the new dataset

1. Click the **Import and export details** Power BI dataset that you created.
2. Configure the visualization. For example, select the **Filled map** visualization, and configure it as follows:
  - Assign the **CountryOrigin** dataset field to the **Location** field of the map visualization.
  - Assign the **Amount** dataset field to the **Color saturation** field of the map visualization.
  - Add the **Activity** and **Year** dataset fields to the **Filters** fields collection of the map visualization.
3. Save the Power BI report as **Import and export details report**.

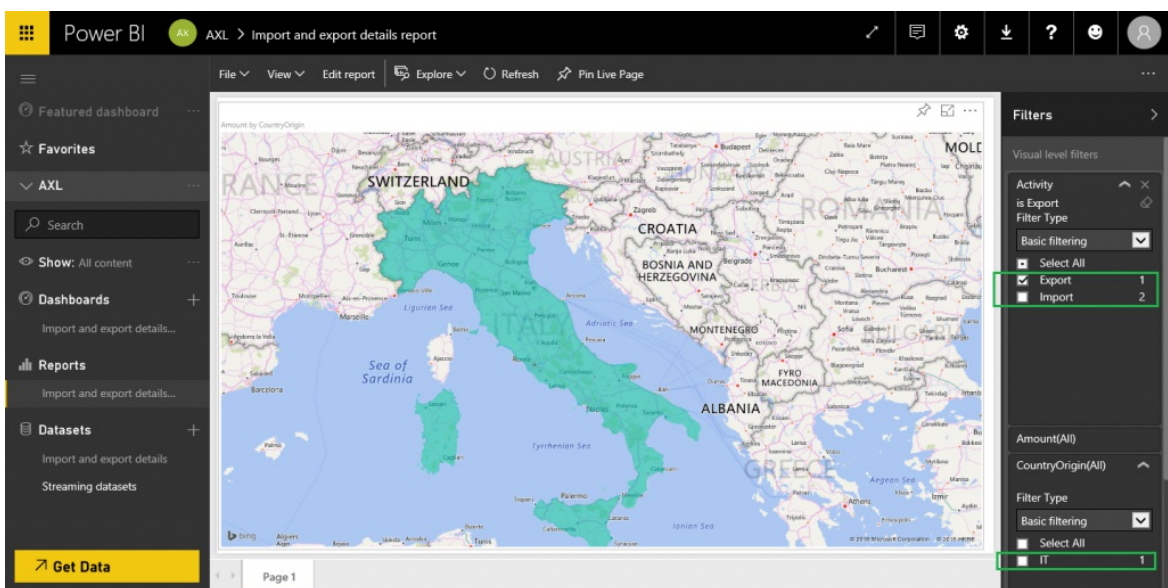


Note that the map shows the countries/regions that are mentioned in the Excel file (Austria and Switzerland in this example). These countries/regions are colored to show the proportion of invoiced amounts for each.

- Update the list of Intrastat transactions. The export transaction that originated from Italy is added.

Date	Direction	Correction	Item number	Category	Commodity	Weight	Invoice amount	Statistical amount	Is...	Country/region of origin
1/31/2016	Arrivals	<input type="checkbox"/>	D0003		920 20 34	4.00	1,000.00	1,000.00	✓	AUT
1/31/2016	Arrivals	<input type="checkbox"/>	T0006		900 22 33	5.00	700.00	700.00	✓	CHE
1/31/2016	Dispatches	<input type="checkbox"/>	D0006		900 30 10	1.00	1,500.00	500.00	✓	ITA

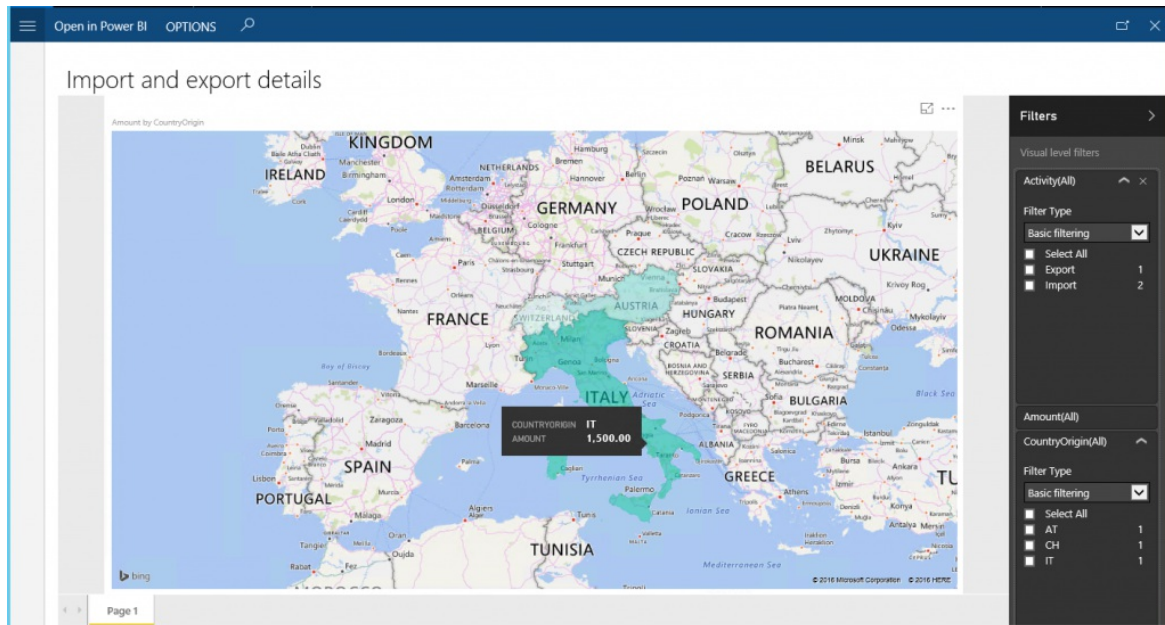
- Wait for the next scheduled execution of the ER report and the next scheduled update of the Power BI dataset. Then review the Power BI report (select to show import transactions only). The updated map now shows Italy.



## Access Power BI report in Finance

Set up the integration with Power BI. For more information, see [Configure Power BI integration for workspaces](#).

1. On the **Electronic reporting** workspace page that supports Power BI integration (**Organization administration > Workspaces > Electronic reporting workspace**), click **Options > Open report catalog**.
2. Select the **Import and export details** Power BI report that you created, to show that report as an action item on the selected page.
3. Click the action item to open the page that shows the report that you designed in Power BI.



## Additional resources

[Electronic reporting \(ER\) destinations](#)

[Electronic reporting \(ER\) overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Generate printable FTI forms

2/18/2021 • 10 minutes to read • [Edit Online](#)

The Electronic reporting (ER) framework lets you generate printable free text invoice (FTI) forms as Microsoft Office documents. This topic provides information about how to build your own configurations as well as details of available configuration templates.

## Overview

In addition to the existing capability of generating printable FTI forms by using Microsoft SQL Server Reporting Services (SSRS), you can now use the ER framework. You can manage printable FTI forms in Microsoft Office Excel and Word. You can also modify the layout, data flow, and formatting to meet specific requirements without making code changes.

### NOTE

If you want to start with an overview of existing ER configurations for this sample of the printable FTI forms solution, you can go directly to section **Download sample ER configurations to generate printable FTI forms** later in this topic.

## Create customized configurations for FTI printable forms

As part of your customized solution for printable FTI forms, you must create a set of ER configurations.

### Configure the ER data model

Your application must include the ER data model configuration that contains a data model that describes the customer invoicing business domain. As a requirement, the name of the data model must be **CustomersInvoicing**. For information about how to design ER data models, see [ER Design domain specific data model](#).

### Configure the ER model mapping

Your application must include the ER model mapping for the CustomersInvoicing data model. The model mapping can be in either the ER data model configuration or the ER model mapping configuration. However, the name of the root descriptor of the model mapping must be **FreeTextInvoice**.

The mapping must contain the following data sources:

- Data source type: **Table records**
  - This data source must be named **CustInvoiceJour**.
  - It must refer to the CustInvoiceJour application table.
  - It's used at runtime to pass from the application to the ER model mapping the list of invoices that have been selected for printing.
- Data source type: **Object**
  - This data source must be named **PrintMgmtPrintSettingDetail**.
  - It must refer to the **PrintMgmtPrintSettingDetail** application class.
  - It's used at runtime to pass from the application to the ER model mapping details of the print management settings for the ER format that is running.

The details of the application integration with the ER framework can be found in the



**ERPrintMgmtReportFormatSubscriber** class (ER Application Suite integration model) in the source code of the application.

For more information about the design of ER model mappings, see [Define ER model mappings and select data sources for them](#).

### Configure the ER format

In your application instance, you must have the ER format configuration that will be used to generate FTI forms.

#### NOTE

This format configuration must be created for the CustomersInvoicing data model, and it must use the model mapping that has the **FreeTextInvoice** root descriptor.

For information about how to configure ER formats, see [ER Create a format configuration \(November 2016\)](#). For information about how to design ER formats to generate reports in OpenXML format, see [ER Design a configuration for generating reports in OPENXML format \(November 2016\)](#).

## Configure print management

To generate FTI forms by using the ER framework, you can assign ER formats in the same way that you assign SSRS reports. To associate the ER format with all Accounts receivable FTIs, go to **Accounts receivable > Setup > Forms > Form setup > General > Print management > Free text invoice > Original**. To associate the ER format with a specific customer or invoice, follow these steps.

1. Go to **Accounts receivable > Invoices > All free text invoices**.
2. Select the FTI to associate the ER format with, and open the **Print management setup** page.
3. Select the document level to specify the scope of invoices for processing.
4. Select the ER format for the specified document level.

The screenshot shows the Dynamics 365 interface for configuring print management. The breadcrumb trail is "Accounts receivable > Invoices > All free text invoices". The page title is "Print management setup" for "US-001". The left navigation pane shows a tree view with "Free text invoice" selected, and "Copy <Default>" highlighted. The main content area is titled "ORIGINAL OR COPY IDENTIFICATION" and contains several fields: "Original / copy" (Copy), "Name" (Copy), "Sus" (All countries/regions), "No" (Report format), "Default country/region", a checked checkbox for "Customer FTI report (GER)", "FreeTextInvoice.Report", a dropdown menu showing "FreeTextInvoice.Report", "Number of copies" (1), and "Footer text".

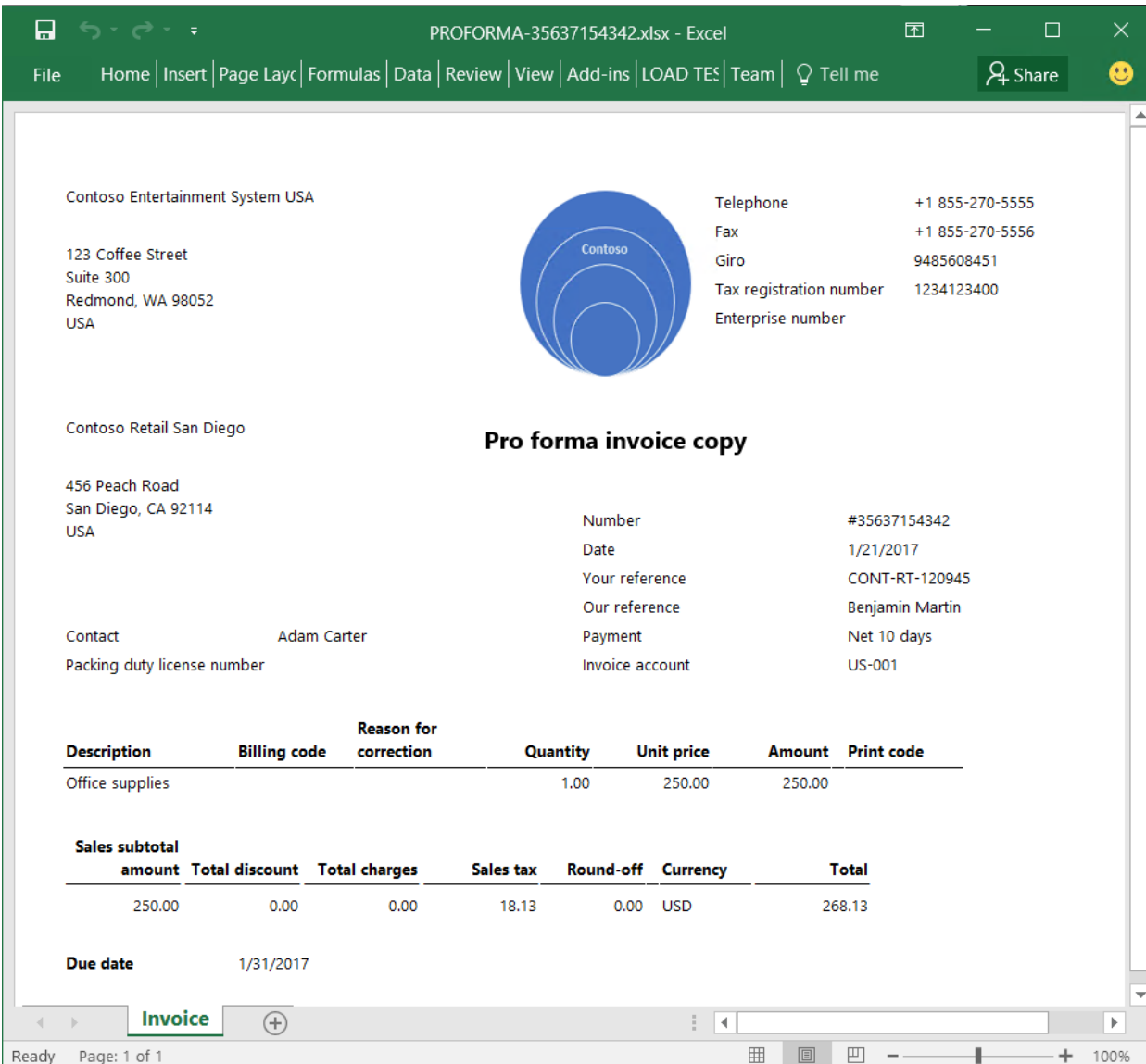
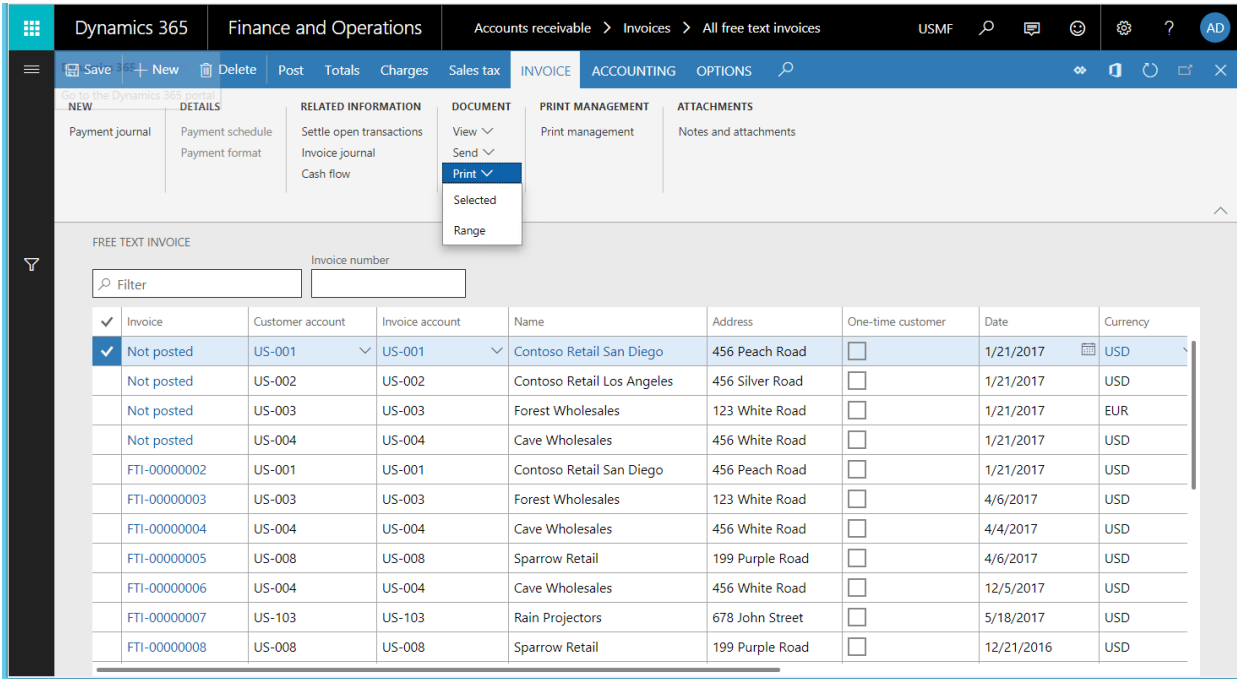
#### NOTE

Only ER formats that use the **FreeTextInvoice** root descriptor of the CustomersInvoicing data model appear in the **Report format lookup** field for the selected format.

# Generate FTI forms

FTI forms are generated in the ER framework in the same way that SSRS reports are generated.

To generate FTI forms, you can select invoices either by range or by selection.



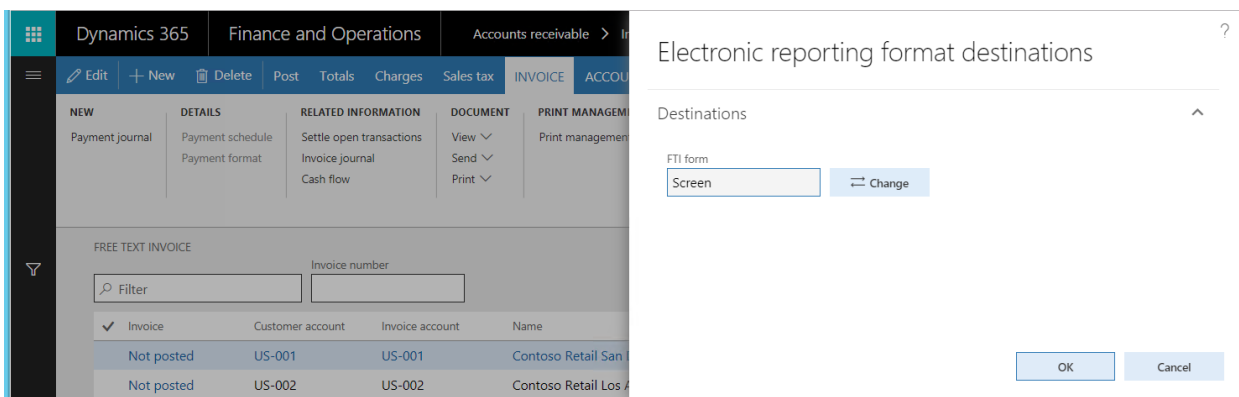
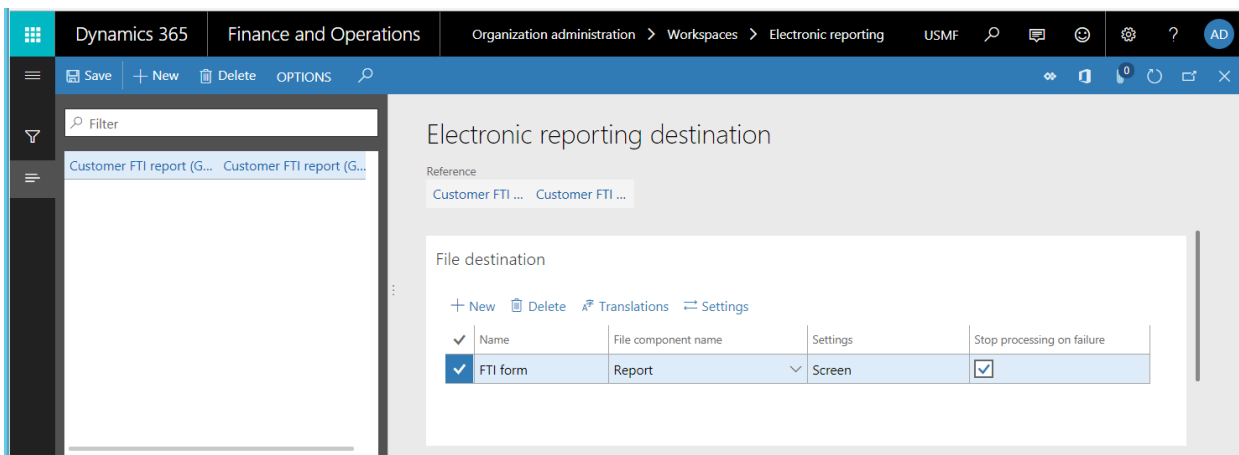
When you use ER formats to print FTI forms in this way, the default ER file destinations are used. You can't change the destination. For more information about how to configure the ER destinations for ER formats, see [Electronic reporting \(ER\) destinations](#).

You can also generate FTI forms when you post an FTI, by turning **Print invoice** on and turning **Use print management destinations** off.

**NOTE**

When you use ER formats to print FTI forms in this way, the default ER file destinations are used. You can change the default destination at runtime if the destination has already been configured. To change the destination, you must have the following security privilege:

- **Name:** ERFormatDestinationRuntimeMaintain
- **Label:** Maintain electronic reporting format destination during runtime



The ER framework currently supports the following destinations for generated documents:

- **Downloaded file** – Generated forms are offered as downloads that you can save by using the browser.
- **Screen** – Microsoft 365 Excel is used to preview generated FTI forms in Excel format.
- **SharePoint folder** – Generated forms are stored based on the settings of the Document management framework.
- **Application archive** – Generated forms are stored as attachments of execution log records in the Microsoft Azure Storage.
- **Email** – Generated forms are sent as email attachments.

**NOTE**

You can't send the FTI forms that are generated directly to the printer, because direct printing that uses the Dynamics Printer Routing Agent isn't currently supported.

# Download sample ER configurations to generate printable FTI forms

You can download sample ER configurations to use as a template for your FTI solution. The configurations are stored in the Shared asset library in Microsoft Dynamics Lifecycle Services (LCS). The configurations include:

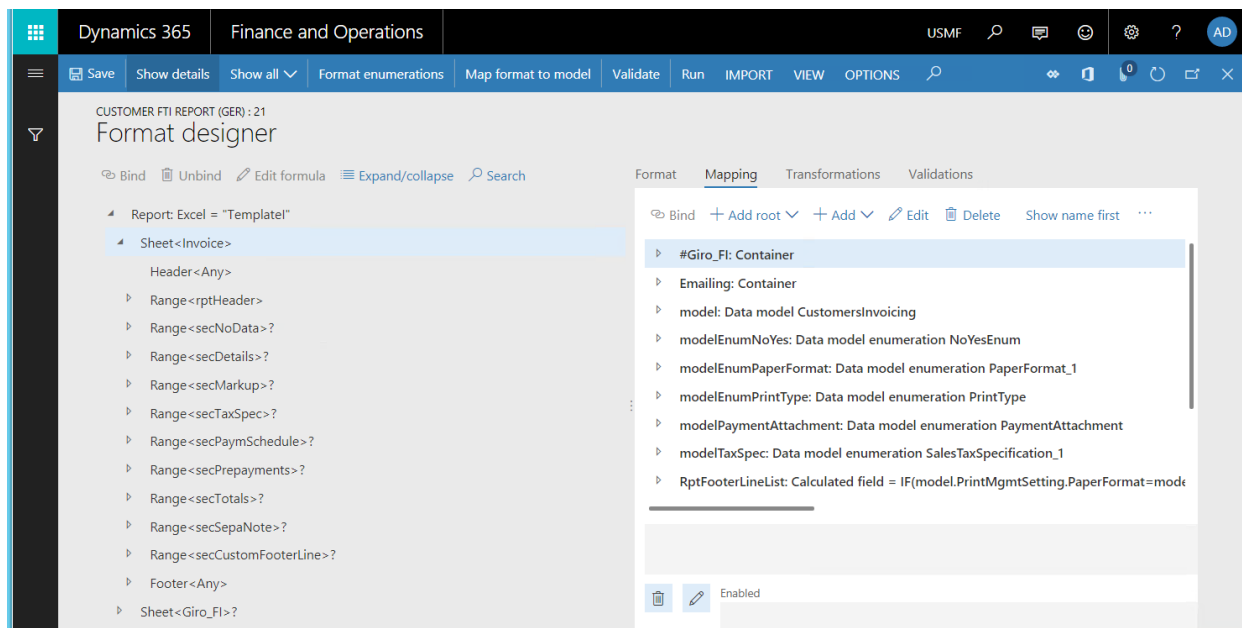
- The **Customer invoicing model** configuration contains the required data model and model mapping.
- The **Customer FTI report (GER)** configuration contains the sample format.

## NOTE

These configurations have been created as samples to help clarify possible scenarios. The future of these configurations depends on the results of this evaluation and any feedback that is received.

## Features that are implemented in the sample ER format

In the sample ER format configuration, an Excel file is used as a template to generate FTI forms.



Currently, this sample ER format supports the following features to generate FTI forms:

- FTI forms are generated for both original invoices that have been posted and original invoices that haven't yet been posted. Corrected invoices and credit notes aren't supported.
- FTI forms are generated in the invoice language. The format of values and dates in the generated forms is based on the settings of the user's client locale.
- Generated invoices show data unavailability notifications if there are no lines in the invoices that are processed.
- Generated invoice headers are based on the paper format that has been selected for the FTI form on the **Accounts receivable parameters** page. Company details appear in the header of the generated invoice form only if the paper format is blank.
- Generated invoice forms show company and customer tax exempt numbers when the appropriate option has been selected for the FTI form on the **Accounts receivable parameters** page.
- The generated invoice lines and invoice totals sections show the default invoice's monetary details in the invoice registration currency.
- The generated invoice totals section can show monetary details in the euro currency and the invoice registration currency when the **Print amount in currency representing the euro** option is enabled on the **Accounts receivable parameters** page.
- Generated invoice forms show any process invoice notes that are available, based on settings on the **Accounts receivable parameters** page. Notes are included for both the whole invoice and each invoice

line.

- Generated invoice forms include notes for the customer FTI form and the processing invoice language when they have been configured in the AR form notes list.
- Depending on the Print management settings, generated invoices include custom footer text when it has been configured for the invoice language, the ER format, and the FTI document scope.
- The totals section of generated invoice forms includes any cash discount information that is available.
- The payment schedule section of generated invoice forms includes any payment schedule details that are available.
- The markup section of generated invoice forms includes any charges transactions that are available.
- Generated invoice forms include sales tax details, based on the **Sales tax specification** setting on the **Accounts receivable parameters** page. This section can show tax details either in the invoice registration currency only, or in the invoice registration currency and the company accounting currency at the same time.
- Generated invoice forms show direct debit notification details. For example, they show when the method of payment that has the mandatory direct debit mandate ID was selected for the invoice, when the processing invoice was registered in the euro currency, and when the direct debit mandate ID was defined for the invoice.
- Generated invoices show any prepayment details that are available for posted invoices.
- Generated invoice forms can be sent to an invoice customer as an email attachment. The appropriate ER file destination should be configured for the ER format that is being used.

### **Country/region-specific features**

The following country/region-specific features are included in the sample ER format to show how specific requirements can be handled in ER configurations.

#### **Norway**

The Enterprise register term is put on the header of the generated invoice form when the invoice is processed for a legal entity that is configured in the following manner:

- The country/region context for Norway is used.
- The **Print Foretaksregisteret** parameter is active on sales documents.

#### **Spain**

The **Special regime for cash accounting method** term is put on the header of the generated invoice form when the invoice is processed for a legal entity that is configured in the following manner:

- The country/region context for Spain is used.
- The special regime for the cash accounting method is enabled on the invoice processing date.

When cash discount details, such as the cash discount amount and invoice line net amount, are available, they are presented in the invoice totals section of the generated invoice form when it has been processed for a legal entity that is configured in the following manner:

- The country/region context for Spain is used.
- **Cash discount is applied in the invoice** is turned on in the invoice option (**General ledger parameters > Sales tax section**).

#### **Italy**

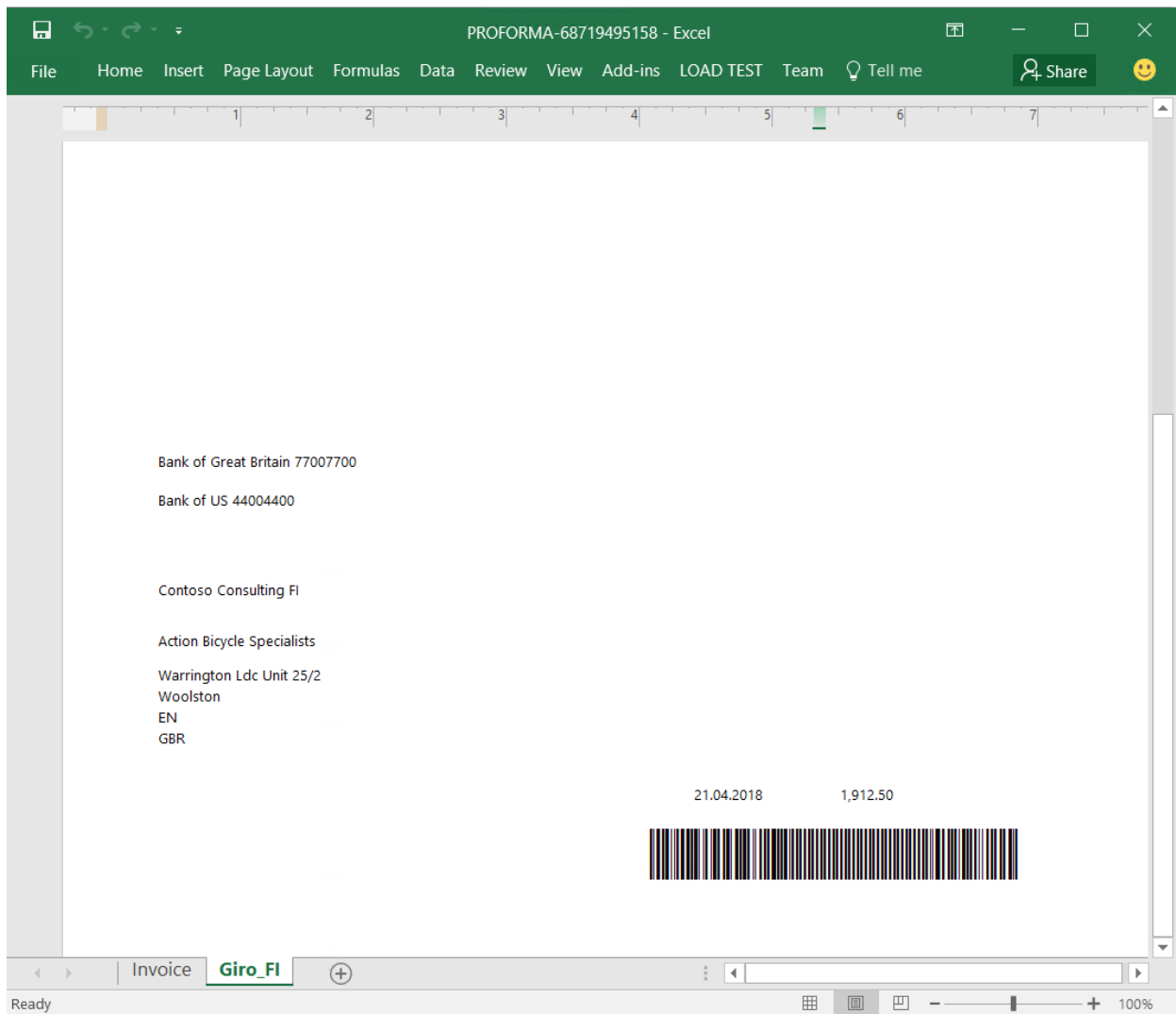
The goods discount mark is included on the invoice lines of the generated invoice when it's being processed for a legal entity that is configured using the country/region context for Italy.

#### **Finland**

In addition to the generated invoice form, Giro money transfer slips can be generated as follows:

- For the legal entity that uses the country/region context for Finland, and that has at least one bank account that is marked as **Giro account** and **Bank bar code**.

- For an invoice that is marked as required for the **Finnish** associated payment attachment.



#### NOTE

The sample ER format has been configured to optionally generate the Giro money transfer slips in the separate worksheet.

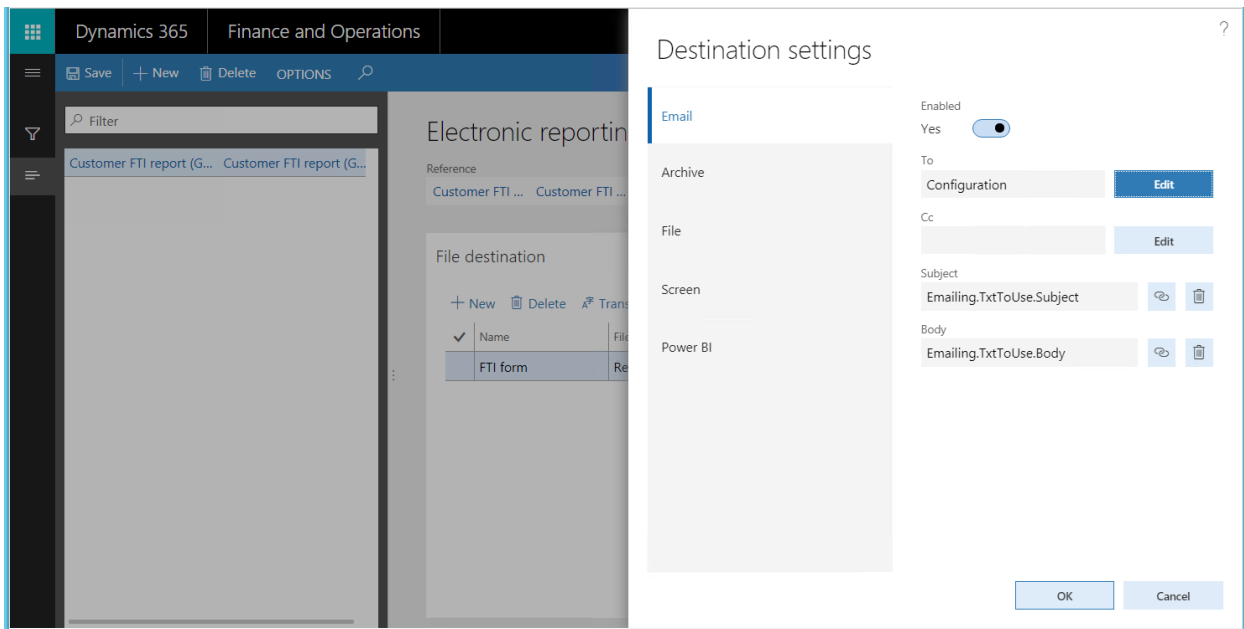
#### NOTE

You must first install the font that is used to generate the bar code on the local machine where the generated invoice form in Excel format will be previewed.

### Use the sample ER format to configure email destinations

Use the following elements of the sample ER format to configure email destinations:

- The email address of a customer contact can be accessed through the following ER expression: **model.InvoiceBase.Contact.ElectronicMail**.
- The email subject text can be accessed through the following ER expression: **Emailing.TxtToUse.Subject**.
- The email body text can be accessed through the following ER expression: **Emailing.TxtToUse.Body**.

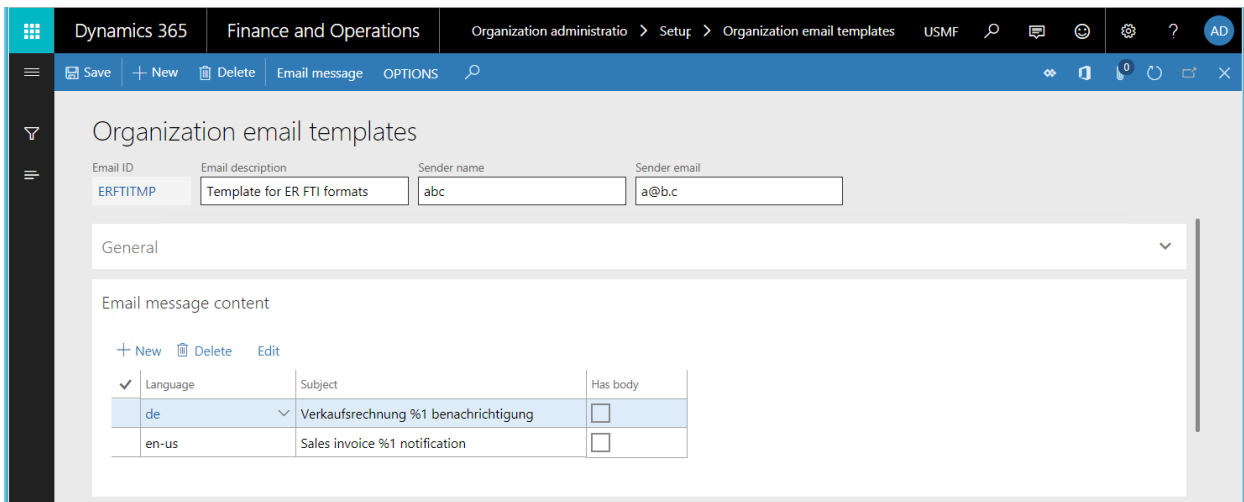


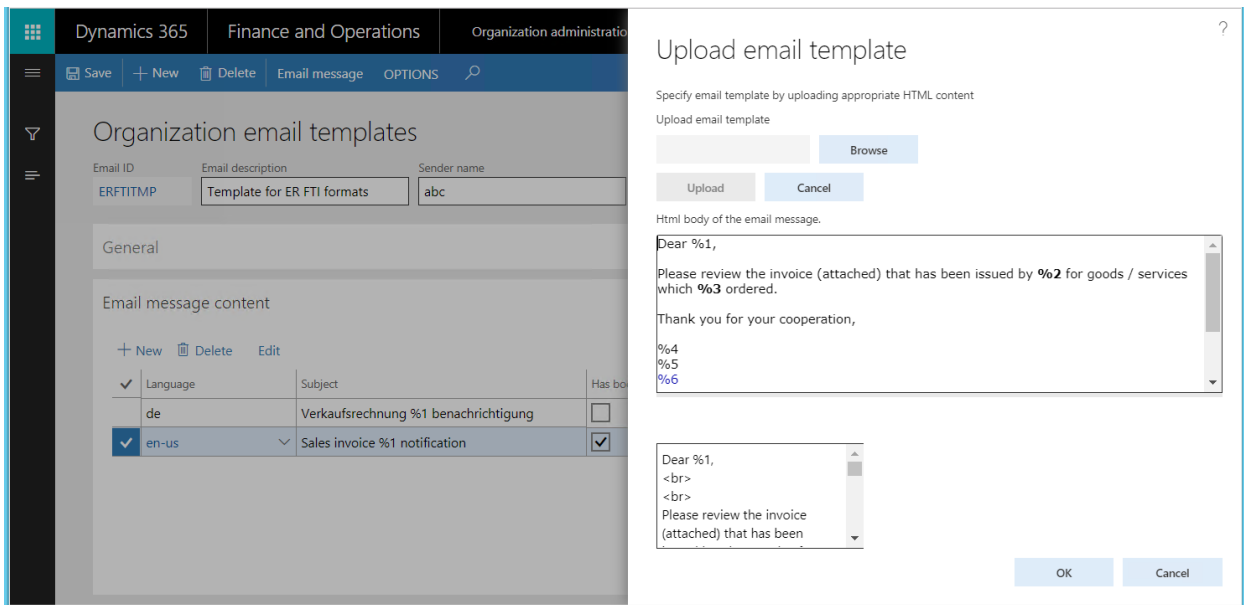
The default text of the email's subject and body is defined in the sample ER format. The language depends on the format's labels. This default text will be used for emails if a custom organization email template that has the predefined **ERFTITMP** ID hasn't been added.

**NOTE**

The **ERFTITMP** email template ID has been defined in the sample ER format. It can be changed as required in a new ER format that is created from this sample format.

If the organization email template that has the predefined **ERFTITMP** ID has been added for the legal entity that you're processing the invoice for, the template for the email subject and body text will be used to generate the email.

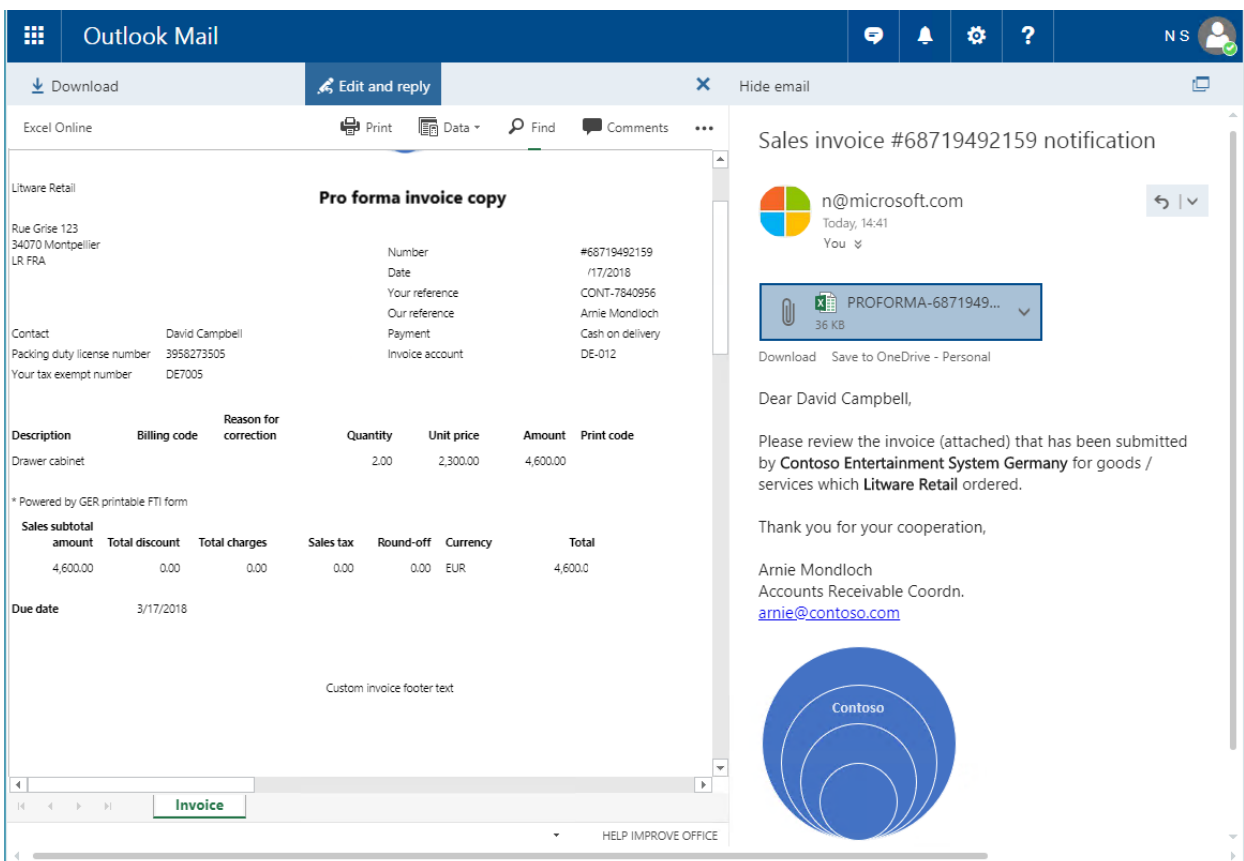




The `Emailing.TxtToUse.Subject` ER expression of the sample ER format is configured to replace any occurrences of the placeholder `%1` by the processing invoice ID.

The `Emailing.TxtToUse.Body` expression of the sample format is configured for the following substitutions for placeholders:

- `"%1"` is replaced with the name of the customer's contact person.
- `"%2"` is replaced with the company name.
- `"%3"` is replaced with the customer name.
- `"%4"` is replaced with the name of the company's contact person.
- `"%5"` is replaced with the job title of the company's contact person.
- `"%6"` is replaced with the email address of the company's contact person.



## Additional resources



## Electronic reporting (ER) overview

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Trace the execution of ER formats to troubleshoot performance issues

2/18/2021 • 15 minutes to read • [Edit Online](#)

As part of the process of designing Electronic reporting (ER) configurations to generate electronic documents, you define the method that is used to get data out of the application and enter it in the output that is generated. The ER performance trace feature helps significantly reduce the time and cost that are involved in collecting the details of ER format execution and using them to troubleshoot performance issues. This tutorial provides guidelines about how to take performance traces for executed ER formats, and how to use the information from these traces to help improve performance.

## Prerequisites

To complete the examples in this tutorial, you must have the following access:

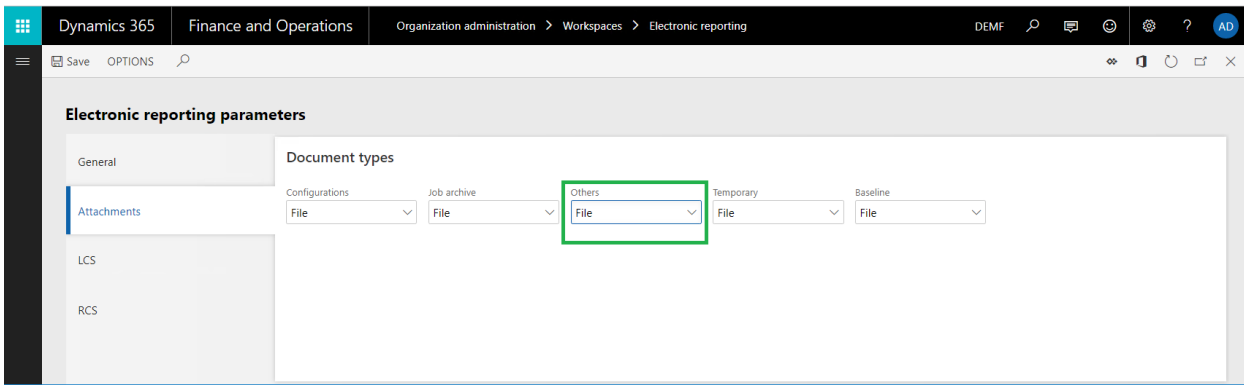
- Access to one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator
- Access to the instance of Regulatory Configuration Services (RCS) that has been provisioned for the same tenant as the application, for one of the following roles:
  - Electronic reporting developer
  - Electronic reporting functional consultant
  - System administrator

You must also download and locally store the following files.

FILE	CONTENT
Performance trace model.version.1	<a href="#">Sample ER data model configuration</a>
Performance trace metadata.version.1	<a href="#">Sample ER metadata configuration</a>
Performance trace mapping.version.1.1	<a href="#">Sample ER model mapping configuration</a>
Performance trace format.version.1.1	<a href="#">Sample ER format configuration</a>

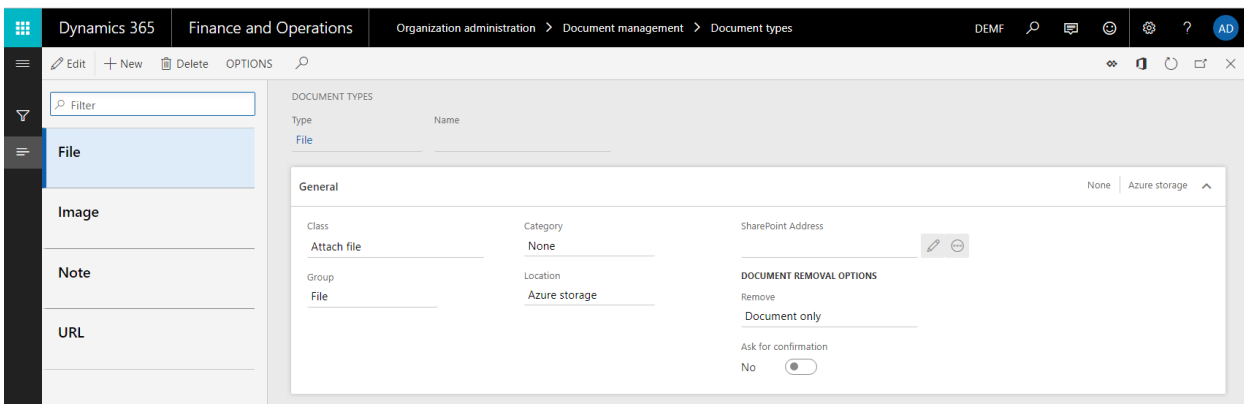
### Configure ER parameters

Each ER performance trace that is generated in the application is stored as an attachment of the execution log record. The Document management (DM) framework is used to manage these attachments. You must configure ER parameters in advance, to specify the DM document type that should be used to attach performance traces. In the **Electronic reporting** workspace, select **Electronic reporting parameters**. Then, on the **Electronic reporting parameters** page, on the **Attachments** tab, in the **Others** field, select the DM document type to use for performance traces.



To be available in the **Others** lookup field, a DM document type must be configured in the following manner on the **Document types** page (**Organization administration > Document management > Document types**):

- **Class:** Attach file
- **Group:** File

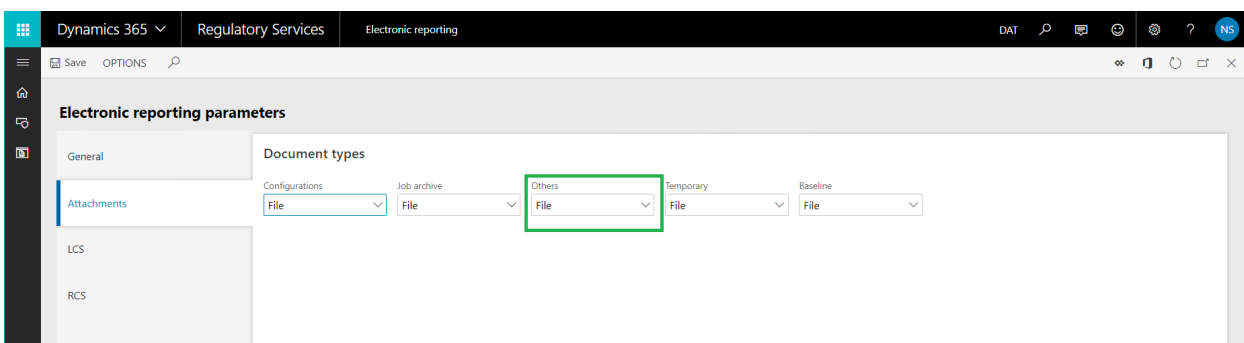


#### NOTE

The selected document type must be available in every company of the current instance, because DM attachments are company-specific.

### Configure RCS parameters

ER performance traces that are generated will be imported into RCS for analysis by using the ER format designer and the ER mapping designer. Because ER performance traces are stored as attachments of the execution log record that is related to the ER format, you must configure RCS parameters in advance, to specify the DM document type that should be used to attach performance traces. In the instance of RCS that has been provisioned for your company, in the **Electronic reporting** workspace, select **Electronic reporting parameters**. Then, on the **Electronic reporting parameters** page, on the **Attachments** tab, in the **Others** field, select the DM document type to use for performance traces.



To be available in the **Others** lookup field, a DM document type must be configured in the following manner on

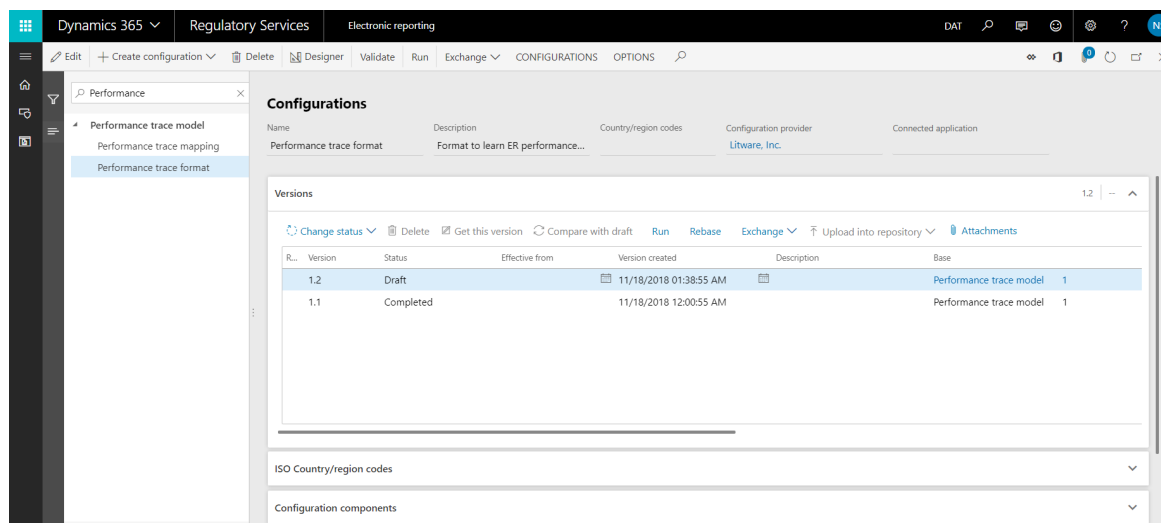
the Document types page (Organization administration > Document management > Document types):

- Class: Attach file
- Group: File

## Design an ER solution

Assume that you've started to design a new ER solution to generate a new report that presents vendor transactions. Currently, you can find the transactions for a selected vendor on the **Vendor transactions** page (go to **Account payable > Vendors > All vendors**, select a vendor, and then, on the Action Pane, on the **Vendor** tab, in the **Transactions** group, select **Transactions**). However, you want to have all vendor transaction at the same time in one electronic document in XML format. This solution will consist of several ER configurations that contain the required data model, metadata, model mapping, and format components.

1. Sign in to the instance of RCS that has been provisioned for your company.
2. In this tutorial, you will create and modify configurations for the **Litware, Inc.** sample company. Therefore, make sure that this configuration provider has been added to RCS and selected as active. For instructions, see the [Create configuration providers and mark them as active](#) procedure.
3. In the **Electronic reporting** workspace, select the **Reporting configurations** tile.
4. On the **Configurations** page, import the ER configurations that you downloaded as a prerequisite into RCS, in the following order: data model, metadata, model mapping, format. For each configuration, follow these steps:
  - a. On the Action Pane, select **Exchange > Load from XML file**.
  - b. Select **Browse** to select the appropriate file for the required ER configuration in XML format.
  - c. Select **OK**.



## Run the ER solution to trace execution

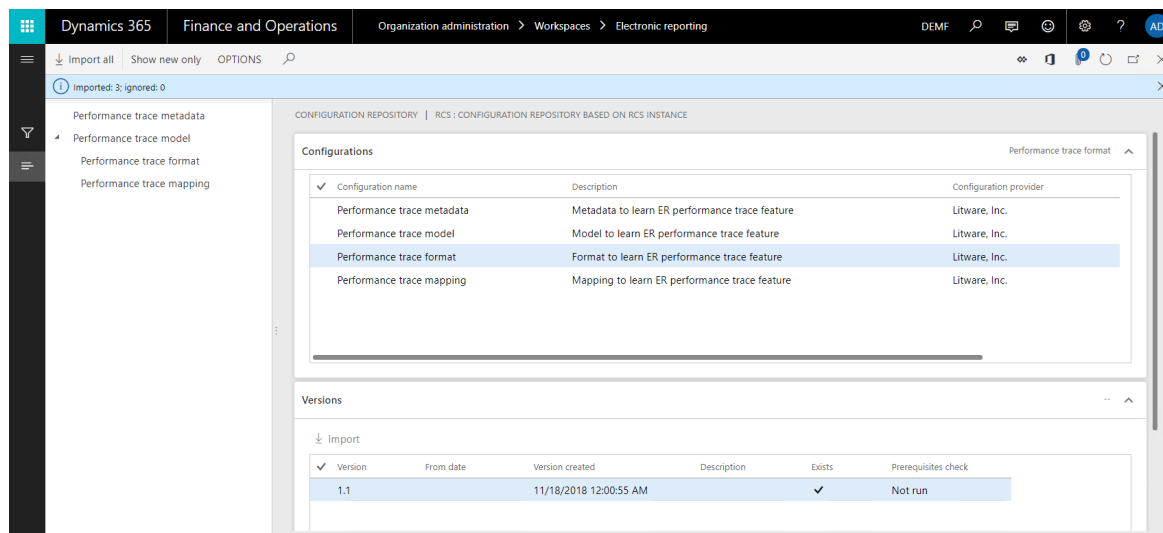
Assume that you've finished designing the first version of the ER solution. You now want to test it in your instance and analyze execution performance.

### Import an ER configuration from RCS into Finance and Operations

1. Sign in to your application instance.
2. For this tutorial, you will import configurations from your RCS instance (where you design your ER components) into your instance (where you test and finally use them). Therefore, you must make sure that all the required artifacts have been prepared. For instructions, see the [Import Electronic reporting](#)

(ER) configurations from Regulatory Configuration Services (RCS) procedure.

3. Follow these steps to import the configurations from RCS into the application:
  - a. In the **Electronic reporting** workspace, on the tile for the **Litware, Inc.** configuration provider, select **Repositories**.
  - b. On the **Configuration repository** page, select the repository of the **RCS** type, and then select **Open**.
  - c. On the **Configurations** FastTab, select the **Performance trace format** configuration.
  - d. On the **Versions** FastTab, select version **1.1** of the selected configuration, and then select **Import**.



The corresponding versions of the data model and model mapping configurations are automatically imported as prerequisites for the imported ER format configuration.

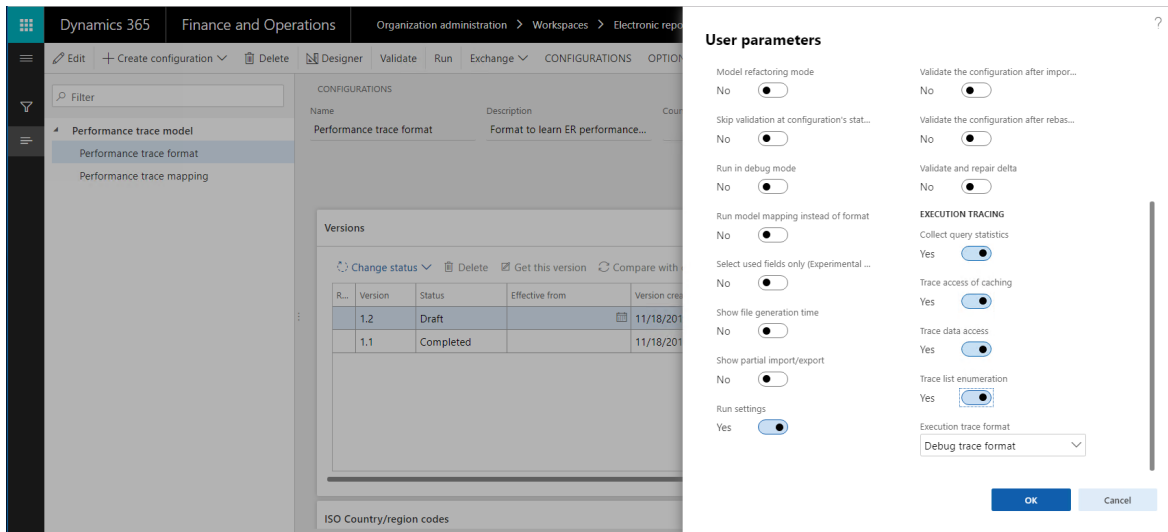
### Turn on the ER performance trace

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
3. In the **User parameters** dialog box, in the **Execution tracing** section, follow these steps:
  - a. In the **Execution trace format** field, select **Debug trace format** to start to collect the details of ER format execution. When this value is selected, the performance trace will collect information about the time that is spent on the following actions:
    - Running each data source in the model mapping that is called to get data
    - Processing each format item to enter data in the output that is generatedYou use the **Execution trace format** field to specify the format of the generated performance trace that the execution details are stored in for ER format and mapping elements. By selecting **Debug trace format** as the value, you will be able to analyze the content of the trace in ER Operation designer, and see the ER format or mapping elements that are mentioned in the trace.
  - b. Set the following options to **Yes** to collect specific details of the execution of the ER model mapping and ER format components:
    - **Collect query statistics** – When this option is turned on, the performance trace will collect the following information:
      - The number of database calls that were made by data sources
      - The number of duplicate calls to the database
      - Details of the SQL statements that were used to make database calls

- **Trace access of caching** – When this option is turned on, the performance trace will collect information about the ER model mapping's cache usage.
- **Trace data access** – When this option is turned on, the performance trace will collect information about the number of calls to the database for executed data sources of the record list type.
- **Trace list enumeration** – When this option is turned on, the performance trace will collect information about the number of records that are requested from data sources of the record list type.

#### NOTE

The parameters in the **User parameters** dialog box are specific to the user and the current company.



### Run the ER format

1. Select the DEMF company.
2. Go to **Organization administration > Electronic reporting > Configurations**.
3. On the **Configurations** page, in the configuration tree, select the **Performance trace format** item.
4. On the Action Pane, select **Run**.

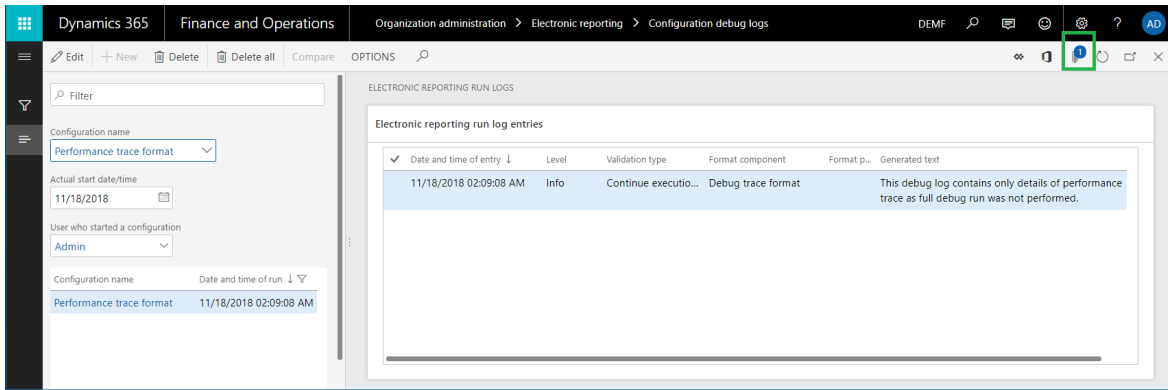
Notice that the file that is generated presents information about 265 transactions for six vendors.

## Review the execution trace

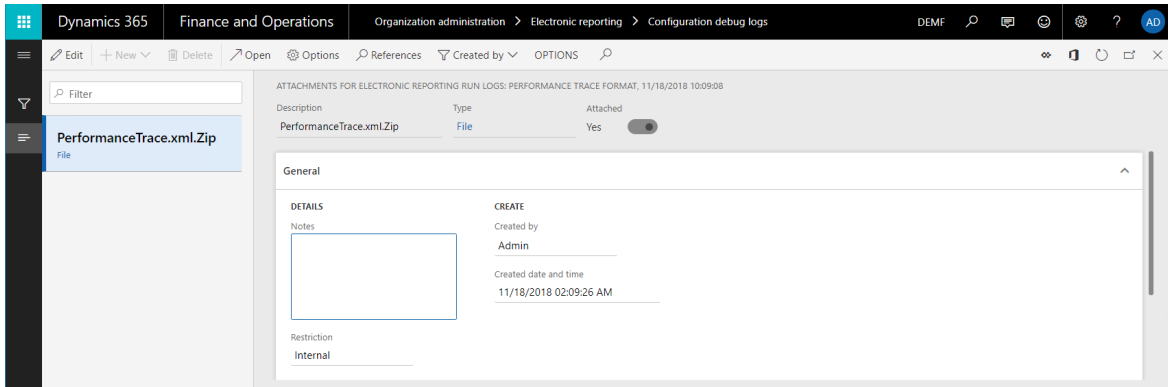
### Export the generated trace from the application

Performance traces are decoupled from the source ER format and can be serialized to an external zip file.

1. Go to **Organization administration > Electronic reporting > Configuration debug logs**.
2. On the **Electronic reporting run logs** page, in the left pane, in the **Configuration name** field, select **Performance trace format** to find the log records that have been generated by the execution of the **Performance trace format** configuration.
3. Select the **Attachments** button (the paper clip symbol) in the upper-right corner of the page, or press **Ctrl+Shift+A**.



4. On the **Attachments for Electronic reporting run logs** page, on the Action Pane, select **Open** to get the performance trace as a zip file and store it locally.



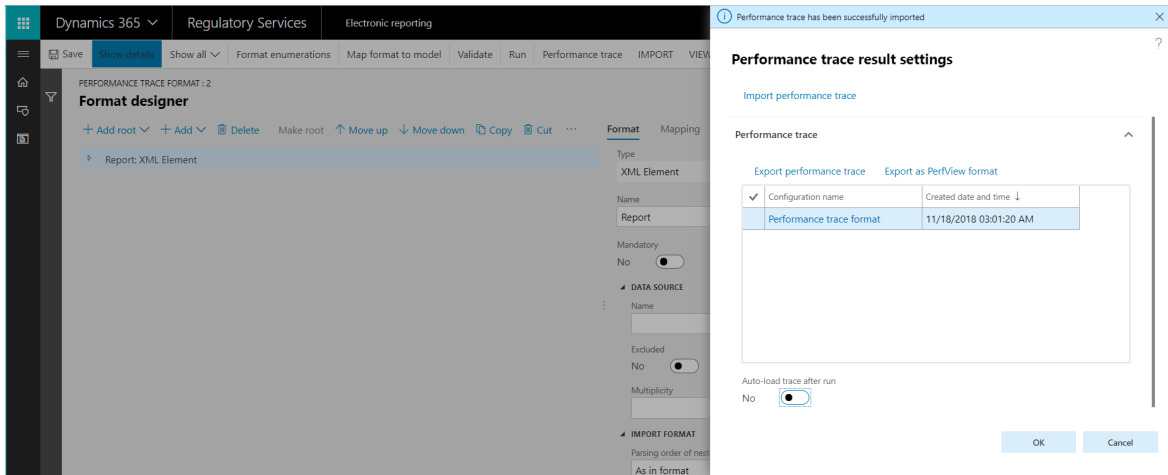
#### NOTE

The trace that is generated has a reference to the source ER report via a unique report identifier in **GUID** format only. The version numbering of the format isn't considered.

Notice that the association between the performance trace that has been generated for the executed ER format and the ER model mapping is based on the root descriptor that was used and the common data model. The version numbering of the format and model mapping isn't considered. The setting of the **Default for model mapping** flag for the model mapping also isn't considered.

#### Import the generated trace into RCS

1. In RCS, in the **Electronic reporting** workspace, select the **Reporting configurations** tile.
2. On the **Configurations** page, in the configuration tree, expand the **Performance trace model** item, and select the **Performance trace format** item.
3. On the Action Pane, select **Designer**.
4. On the **Format designer** page, on the Action Pane, select **Performance trace**.
5. In the **Performance trace result settings** dialog box, select **Import performance trace**.
6. Select **Browse** to select the zip file that you exported earlier.
7. Select **OK**.

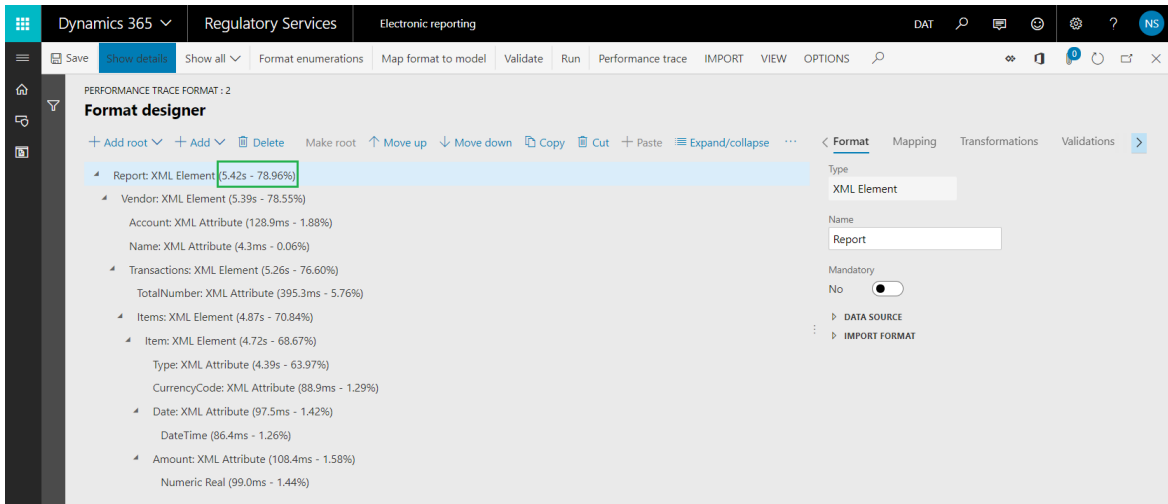


## Use the performance trace for analysis in RCS – Format execution

1. In RCS, on the **Format designer** page, select **Expand/collapse** to expand the content of all format items.

Notice that additional information is shown for some items of the current format:

- The actual time that was spent entering data in the generated output by using the format item
- The same time expressed as a percentage of the total time that was spent generating the whole output



2. Close **Format designer** page.

## Use the performance trace for analysis in RCS – Model mapping

1. In RCS, on the **Configurations** page, in the configuration tree, select the **Performance trace mapping** item.
2. On the Action Pane, select **Designer**.
3. Select **Designer**.
4. On the **Model mapping designer** page, on the Action Pane, select **Performance trace**.
5. Select the trace that you imported earlier.
6. Select **OK**.

Notice that new information becomes available for some data source items of the current model mapping:

- The actual time that was spent getting data by using the data source
- The same time expressed as a percentage of the total time that was spent running the whole model mapping

Notice that ER informs you that the current model mapping duplicates database requests while the `VendTable/<Relations/VendTrans.VendTable_AccountNum` data source is run. This duplication occurs because the list of vendor transactions is called two times for each iterated vendor record:



- One call is made to enter details of each transaction in the data model, based on configured bindings.
- One call is made to enter the calculated number of transactions per vendor in the data model.

The screenshot shows the 'Model mapping designer' interface. A yellow warning banner at the top states: 'Detected number of duplicate queries to database - 6. Consider caching them.' The 'DATA SOURCE TYPES' pane on the left shows 'Data model' selected. The 'DATA SOURCES' pane shows a tree structure with 'VendTable (3.04s - 76.16%)[12][Q:6]' selected. Below this, a table shows performance statistics:

Path	Queries	Duplicated queries	Description
VendTable/<Relations/VendTrans.VendTable_AccountNum	12	6	
VendTable	1	0	

The 'DETAILS' pane shows 'Performance statistics' with a 'Find in tree' button. A SQL query is displayed in the right pane, starting with 'SELECT T1.TAXWITHHOLDGROUP,T1.RESIDENCEFOREIGNCOUNTRYREGIO'.

The value [Q:530] indicates that the VendTrans table was called 530 times to return a record from that table to the VendTable/<Relations/VendTrans.VendTable\_AccountNum data source. The value [530] indicates that the VendTable/<Relations/VendTrans.VendTable\_AccountNum data source was called 530 times to return a record from that data source and enter the details from it in the data model.

We recommend that you use caching for the VendTable/<Relations/VendTrans.VendTable\_AccountNum data source, to reduce the number of calls that are made to get the details for 265 transactions and help improve the performance of the model mapping.

It can also be useful to reduce the number of calls that are made to the LedgerTransTypeList data source. This data source is used to associate each value of the LedgerTransType enumeration with its label. By using this data source, you can find an appropriate label and enter it in the data model for each vendor transaction. The current number of calls to this data source (9,027) is quite high for 265 transactions.

The screenshot shows the 'Model mapping designer' interface. A yellow warning banner at the top states: 'Detected number of duplicate queries to database - 6. Consider caching them.' The 'DATA SOURCE TYPES' pane on the left shows 'Data model' selected. The 'DATA SOURCES' pane shows a tree structure with 'LedgerTransTypeList (28.4ms - 0.71%)[9027]' selected. The 'DATA MODEL' pane on the right shows the root definition with the following structure:

```

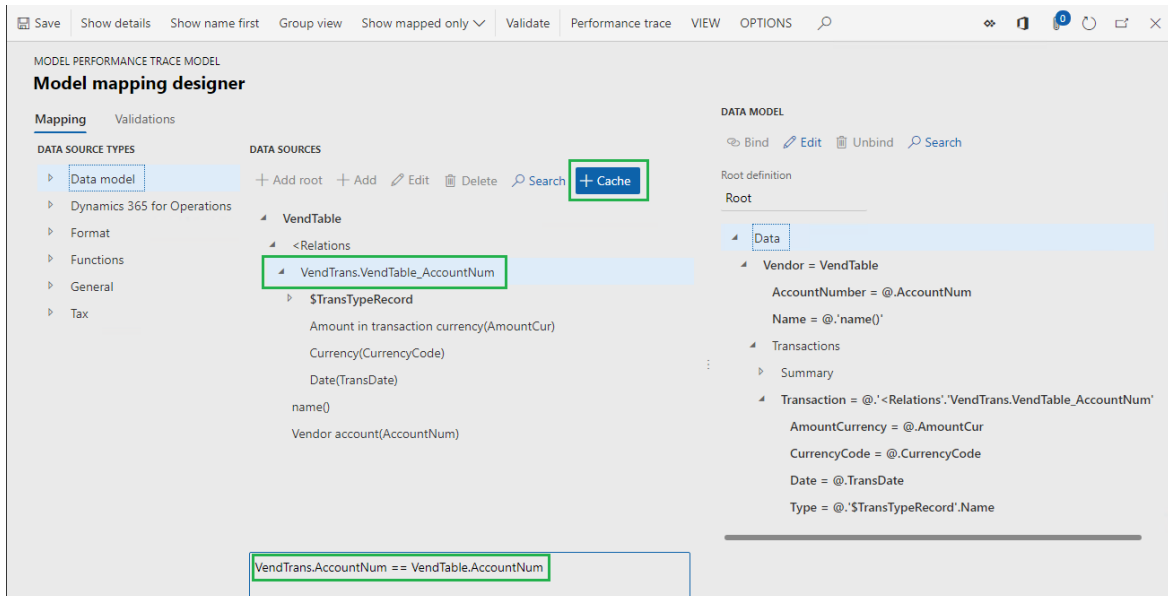
Root
├── Data
│   ├── Vendor = VendTable
│   │   ├── AccountNumber = @AccountNum
│   │   └── Name = @.name()
│   ├── Transactions
│   └── Summary
│       ├── TotalTransactionsNumber = COUNT(@.<Relations'.VendTrans.VendTable_AccountNum')
│       └── Transaction = @.<Relations'.VendTrans.VendTable_AccountNum'

```

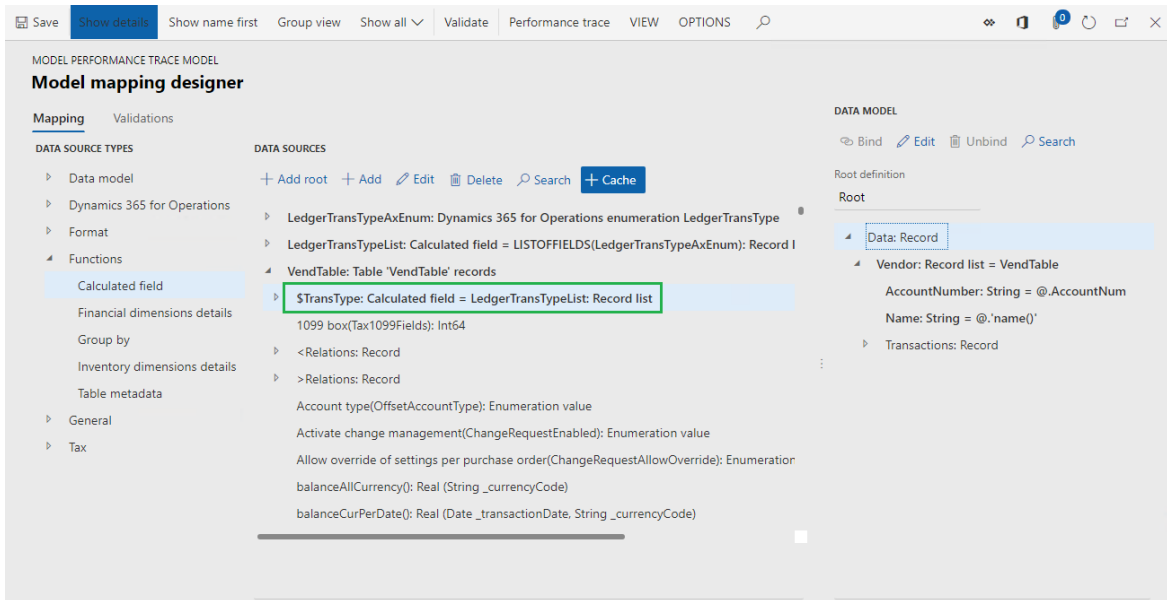
# Improve the model mapping based on information from the execution trace

## Modify the logic of the model mapping

1. Follow these steps to use caching, to help prevent duplicate calls to the database:
  - a. In RCS, on the **Model mapping designer** page, in the **Data sources** pane, select the **VendTable** item.
  - b. Select **Cache**.
  - c. Expand the **VendTable** item, expand the list of one-to-many relations for the VendTable data source (the **<Relations** item), and select the **VendTrans.VendTable\_AccountNum** item.
  - d. Select **Cache**.



2. Follow these steps to bring the LedgerTransTypeList data source into the scope of the VendTable data source:
  - a. In the **Data source types** pane, expand the **Functions** item, and select the **Calculated field** item.
  - b. In the **Data sources** pane, select the **VendTable** item.
  - c. Select **Add**.
  - d. In the **Name** field, enter **\$TransType**.
  - e. Select **Edit formula**.
  - f. In the **Formula** field, enter **LedgerTransTypeList**.
  - g. Select **Save**.
  - h. Close the **Formula editor** page.
  - i. Click **OK**.
3. Follow these steps to do caching of the **\$TransType** field:
  - a. Select the **LedgerTransTypeList** item.
  - b. Select **Cache**.
  - c. Select the **VendTable.\$TransType** item.
  - d. Select **Cache**.



4. Follow these steps to change the **\$TransTypeRecord** field so that it starts to use the cached **\$TransType** field:

a. In the **Data sources** pane, expand the **VendTable** item, expand the **<Relations** item, expand the **VendTrans.VendTable\_AccountNum** item, and select the **VendTable.VendTrans.VendTable\_AccountNum.\$TransTypeRecord** item.

b. Select **Edit**.

c. Select **Edit formula**.

d. In the **Formula** field, find the following expression:

```
FIRSTORNUL (WHERE (LedgerTransTypeList, LedgerTransTypeList.Enum = @.TransType))
```

e. In the **Formula** field, enter the following expression:

```
FIRSTORNUL (WHERE (VendTable.'$TransType', VendTable.'$TransType'.Enum = @.TransType)).
```

f. Select **Save**.

g. Close the **Formula editor** page.

h. Select **OK**.

5. Select **Save**.

6. Close the **Model mapping designer** page.

7. Close the **Model mappings** page.

### Complete the modified version of the ER model mapping

1. In RCS, on the **Configurations** page, on the **Versions** FastTab, select version 1.2 of the **Performance trace mapping** configuration.

2. Select **Change status**.

3. Select **Complete**.

### Import the modified ER model mapping configuration from RCS into the application

Repeat the steps in the [Import an ER configuration from RCS into Finance and Operations](#) section earlier in this topic to import version 1.2 of the **Performance trace mapping** configuration.

## Run the modified ER solution to trace execution

## Run the ER format

Repeat the steps in the [Run the ER format](#) section earlier in this topic to generate a new performance trace.

## Work with the execution trace

### Export the generated trace from the application

Repeat the steps in the [Export the generated trace from the application](#) section earlier in this topic to save a new performance trace locally.

### Import the generated trace into RCS

Repeat the steps in the [Import the generated trace into RCS](#) section earlier in this topic to import the new performance trace into RCS.

### Use the performance trace for analysis in RCS – Model mapping

Repeat the steps in the [Use the performance trace for analysis in RCS – Model mapping](#) section earlier in this topic to analyze the latest performance trace.

Notice that the adjustments that you made to the model mapping have eliminated duplicate queries to database. The number of calls to database tables and data sources for this model mapping has been also reduced. Therefore, the performance of the whole ER solution has improved.

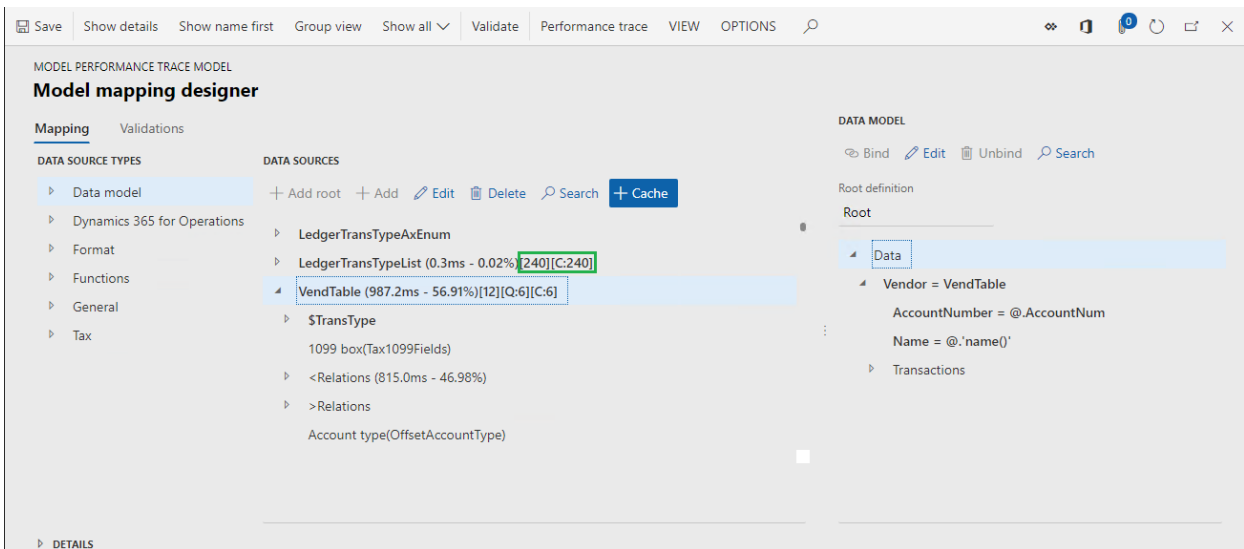
The screenshot shows the 'Model mapping designer' interface. The 'Performance statistics' table is highlighted with a green border. The table has the following data:

Path	Queries	Duplicated queries	Description
VendTable/<Relations/VendTrans.VendTable_AccountNum	6	0	
VendTable	1	0	

Below the table, a query snippet is visible, showing a complex SQL query with various table names and conditions, including 'VENDTABLE T1'.

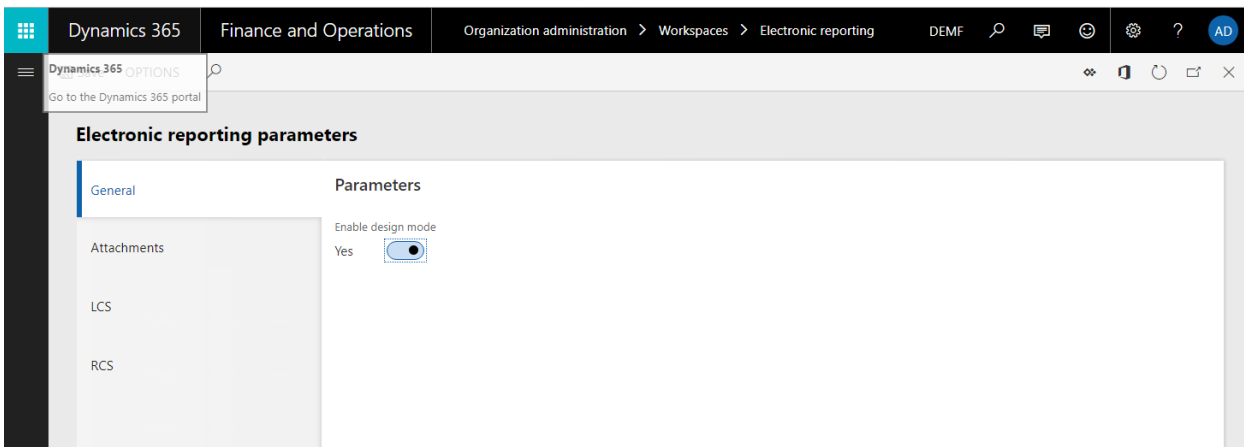
In the trace information, the value [12] for the VendTable data source indicates that this data source was called 12 times. The value [Q:6] indicates that six calls were translated to database calls to the VendTable table. The value [C:6] indicates that the records that were fetched from the database were cached, and six other calls were processed by using the cache.

Notice that the number of calls to the LedgerTransTypeList data source has been reduced from 9,027 to 240.



## Review the execution trace in the application

In addition to RCS, some versions might offer capabilities for an ER framework designer experience. These versions have an **Enable design mode** option that can be turned on. You can find this option on the **General** tab of the **Electronic reporting parameters** page, which you can open from the **Electronic reporting** workspace.



If you use one of these versions, you can analyze the details of generated performance traces directly in the application. You don't have to export them from the application and import them into RCS.

## Review the execution trace by using external tools

### Configure user parameters

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
3. In the **User parameters** dialog box, in the **Execution tracing** section, in the **Execution trace format** field, select **PerfView XML**.

### Run the ER format

Repeat the steps in the [Run the ER format](#) section earlier in this topic to generate a new performance trace.

Notice that the web browser offers a zip file for download. This file contains the performance trace in PerfView format. You can then use the PerfView performance analysis tool to analyze the details of ER format execution.

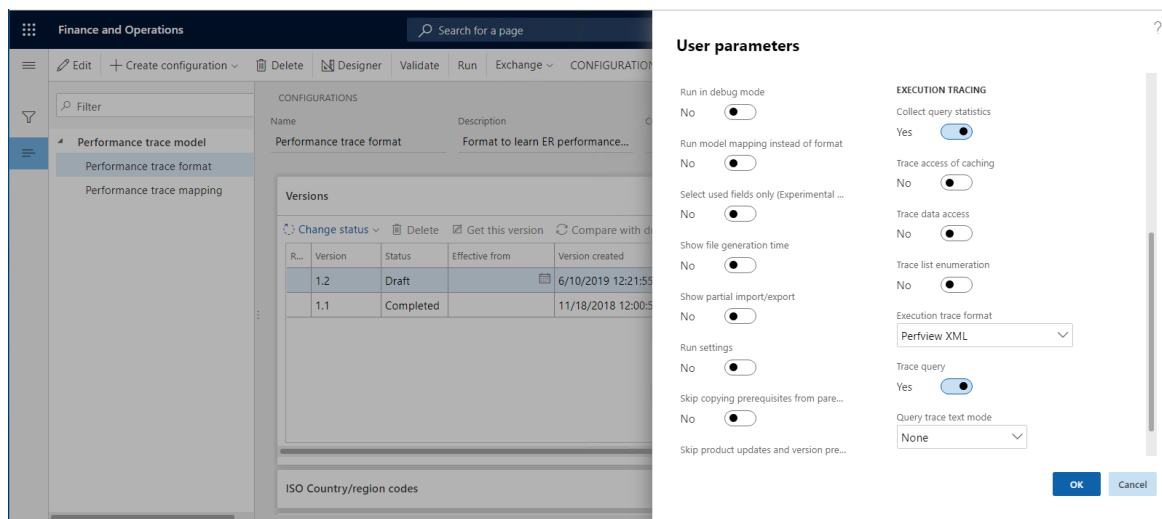
Name	Exc %	Exc	Exc Ct	Inc %	Inc	Ii	Li	F	F	V	V
Format:Report/Vendor/Transactions/Items/Item/Date/DateTime	0.3	29,103	530	1.9	188,945.0	116	1.5	0	0	—	—
Format:Report/Vendor/Transactions/Items/Item/CurrencyCode	0.3	29,009	530	1.8	176,177.0	110	1.5	0	0	—	—
Format:Report/Vendor/Transactions/Items/Item/Type	0.3	26,467	530	37.5	3,637,389.0	178	20.0	0	0	—	—
ModelMapping:VendTable/name()	0.2	21,010	12	0.2	21,010.0	1.7	12.0	0	0	—	—
FormatMapping:model/Data/Vendor	0.2	15,659	2	0.2	23,775.0	5.9	4.0	0	0	—	—
FormatMapping:model/Data/Vendor/AccountNumber	0.1	14,097	12	11.5	1,110,588.0	46.0	24.0	0	0	—	—
ModelMapping:VendTable	0.1	8,116	2	0.1	8,116.0	4.0	2.0	0	0	—	—
ModelMapping:LedgerTransTypeAxEnum	0.1	5,453	92	0.1	5,453.0	59.0	92.0	0	0	—	—
ModelMapping:VendTable/\$TransType	0.0	4,256	10	2.0	192,529.0	16.0	12.0	0	0	—	—
ModelMapping:LedgerTransTypeList/Name	0.0	3,629	92	0.0	3,629.0	39.0	92.0	0	0	—	—
Format:Report/Vendor	0.0	1,869	12	67.4	6,531,173.0	238	27.0	0	0	—	—
Format:Report/Vendor/Transactions	0.0	1,683	12	55.7	5,394,750.0	197	27.0	0	0	—	—
FormatMapping:model/Data/Vendor/Transactions/Transaction	0.0	1,666	12	0.0	1,666.0	13.0	12.0	0	0	—	—
Format:Report/Vendor/Transactions/TotalNumber	0.0	1,218	12	10.3	1,000,553.0	1.7	57.0	0	0	—	—
FormatMapping:model/Data/Vendor/Name	0.0	1,144	12	0.2	22,154.0	92.0	24.0	0	0	—	—
Format:Report/Vendor/Name	0.0	1,027	12	0.2	23,181.0	64.0	36.0	0	0	—	—
Format:Report/Vendor/Account	0.0	785	12	11.5	1,111,373.0	30.0	36.0	0	0	—	—
ROOT	0.0	0	0	100.0	9,692,330.0	353	27.0	0	0	—	—

# Use external tools to review an execution trace that includes database queries

Because of improvements that have been made to the ER framework, the performance trace that is generated in PerfView format now offers more details about ER format execution. In Microsoft Dynamics 365 for Finance and Operations version 10.0.4 (July 2019), this trace can also include details of executed SQL queries to the application database.

### Configure user parameters

1. Go to **Organization administration > Electronic reporting > Configurations**.
2. On the **Configurations** page, on the Action Pane, on the **Configurations** tab, in the **Advanced settings** group, select **User parameters**.
3. In the **User parameters** dialog box, in the **Execution tracing** section, set the following parameters:
  - In the **Execution trace format** field, select **PerfView XML**.
  - Set the **Collect query statistics** option to **Yes**.
  - Set the **Trace query** option to **Yes**.



## Run the ER format

Repeat the steps in the [Run the ER format](#) section earlier in this topic to generate a new performance trace.

Notice that the web browser offers a zip file for download. This file contains the performance trace in PerfView format. You can then use the PerfView performance analysis tool to analyze the details of ER format execution. This trace now includes the details of SQL database access during the execution of the ER format.

Name	E	E	E	I	I	I	I	F	F	V
ModelMapping:VendTable/name()	0.1	19,	12	0.1	19,	1,6	12	0	0	0
Query::SELECT T1.POSTINGPROFILE,T1.ACCOUNTINGEVENT,T1.ACCOUNTNUM,T1.AMOUNTCUR,T1.AMOUNTMST,T1.APPROVE	0.1	18,	542	24,	3,9	3,6	1,0	0	0	0
FormatMapping:model/Data/Vendor	0.1	14,	2	0.1	23,	5,7	4	0	0	0
FormatMapping:model/Data/Vendor/AccountNumber	0.1	12,	12	6,9	1,1	4,7	24	0	0	0
ModelMapping:VendTable	0.1	8,2	2	0.1	8,2	4,1	2	0	0	0
ModelMapping:LedgerTransTypeAxEnum	0.0	5,2	92	0.0	5,2	5,6	92	0	0	0
ModelMapping:VendTable/\$TransType	0.0	4,2	10	1.1	18,5	15,	12	0	0	0
ModelMapping:LedgerTransTypeList/Name	0.0	3,7	92	0.0	3,7	4,0	92	0	0	0
FormatMapping:model/Data/Vendor/Transactions/Transaction	0.0	2,8	12	0.0	2,8	24,0	12	0	0	0
Format:Report/Vendor	0.0	1,9	12	7,7	12,	4,5	2,7	0	0	0
Format:Report/Vendor/Transactions	0.0	1,7	12	7,0	11,	4,1	2,7	0	0	0
Format:Report/Vendor/Transactions/TotalNumber	0.0	1,6	12	5,0	8,2	7,2	1,1	0	0	0
FormatMapping:model/Data/Vendor/Name	0.0	1,1	12	0.1	21,	8,7	24	0	0	0
Format:Report/Vendor/Name	0.0	1,0	12	0.1	22,	6,1	3,6	0	0	0
Format:Report/Vendor/Account	0.0	8,6	12	6,9	1,1	3,1	3,6	0	0	0
Query::SELECT T1.TAXWITHHOLDGROUP,T1.RESIDENCEFOREIGNCOUNTRYREGIONID,T1.PAYMTERMID,T1.LINEDISC,T1.ACCOUN	0.0	4,8	14	1.4	2,2	7,9	2,8	0	0	0
QueryFirst::SELECT T1.POSTINGPROFILE,T1.ACCOUNTINGEVENT,T1.ACCOUNTNUM,T1.AMOUNTCUR,T1.AMOUNTMST,T1.APPR	0.0	3,9	12	1.2	1,9	5,5	3,6	0	0	0
QueryFirst::SELECT T1.TAXWITHHOLDGROUP,T1.RESIDENCEFOREIGNCOUNTRYREGIONID,T1.PAYMTERMID,T1.LINEDISC,T1.ACC	0.0	5,8	2	0.3	4,2	7,1	6	0	0	0
ROOT	0.0	0	0	10,0	16,	5,8	2,8	0	0	0

## Additional resources

- [Electronic Reporting overview](#)
- [Improve performance of ER solutions by adding parameterized CALCULATED FIELD data sources](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Business document management overview

2/18/2021 • 25 minutes to read • [Edit Online](#)

Business users use the [Electronic reporting \(ER\)](#) framework to configure formats for outbound documents in accordance with the legal requirements of various countries/regions. Users can also define the dataflow to specify what application data is placed in generated documents. The ER framework generates outbound documents in Microsoft Office formats (Excel workbooks or Word documents) by using predefined templates. The templates are populated with required data in accordance to configured dataflow while required documents are generated. Each configured format can be published as part of an ER solution to generate specific outbound documents. This is represented by an ER format configuration that can contain templates you can use to generate different outbound documents. Business users can use this framework to manage required business documents.

**Business document management** is built on top of the ER framework and enables business users to edit business document templates by using Microsoft 365 service or appropriate Microsoft Office desktop application. Edits to the documents might include changing business document designs and adding placeholders for additional data without source code changes and new deployments. No knowledge of the ER framework is required to update templates of business documents.

## NOTE

Be aware that Business document management allows you to modify templates that are used to produce business documents such as orders, invoices, etc. While a template has been modified and a new version of it has been published, this version is used to generate required business documents. Business document management cannot be used to modify already generated business documents.

## Supported deployments

Currently, the Business document management feature is implemented only for cloud deployments. If this feature is critical to your on-premises deployment, let us know by providing feedback on the [Ideas](#) site.

## Supported Microsoft Office applications

To use Business document management for editing templates in Excel or Word formats by using Microsoft Office desktop applications, you must have Microsoft Office 2010 or later installed. This is supported in cloud and on-premises deployments.

To use Business document management for editing templates in Excel or Word formats by using Microsoft 365 applications, you must have Microsoft 365 Office for the web subscription. This is supported in cloud deployment.

## Business document availability

For a complete list of all the reports planned for the October 2019 release, see [Configurable business documents reporting in Word and Excel](#).

For a complete list of all the reports planned for the October 2020 release, see [Configurable business documents – Word templates](#).

More reports will become available in future releases. Special notifications about additional reports will be sent

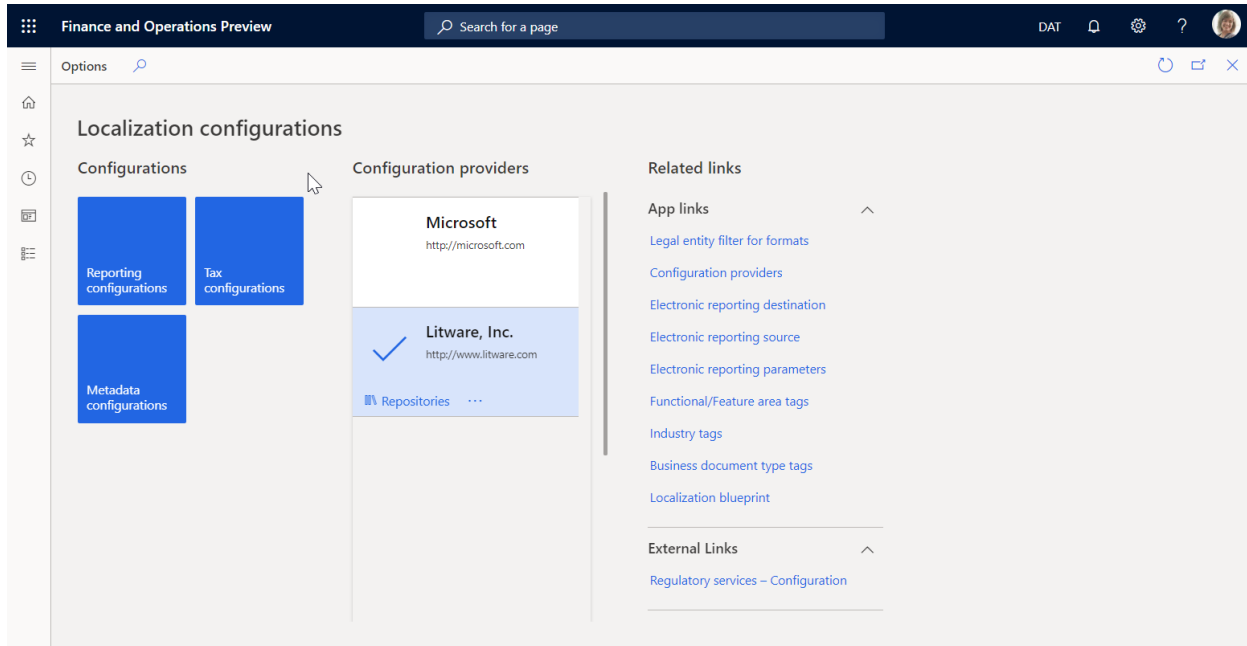


separately. To learn how to review the list of currently available reports, see the section [List of ER configurations that have been released in Finance to support configurable business documents](#) below.

To learn more about this feature, complete the example in this topic.

## Configure ER parameters

Because Business document management is built on top of the ER framework, you must configure the ER parameters to start working with Business document management. To do this, you need to set up the ER parameters as described in [Configure the Electronic reporting \(ER\) framework](#). You also need to add a new configuration provider as described in [Create configuration providers and mark them as active](#).



## Import ER solutions

Sample ER configurations are used in the example of this procedure. You must import, into your current instance of Dynamics 365 Finance, the ER configurations that contain the business document templates that can be edited by using Business document management. Download, and then locally store the following files to complete this procedure.

### Sample ER customer invoicing solution

FILE	CONTENT
Customer invoicing model.version.2.xml	<a href="#">ER data model configuration</a>
Customer FTI report (GER).version.2.3.xml	<a href="#">Free text invoice ER format configuration</a>

### Sample ER payment checks solution

FILE	CONTENT
Model for cheques.version.10.xml	<a href="#">ER data model configuration</a>
Cheques printing format.version.10.9.xml	<a href="#">Payment cheque ER format configuration</a>

### Sample ER foreign trade solution

FILE	CONTENT
Intrastat model.version.1.xml	<a href="#">ER data model configuration</a>
Intrastat report.version.1.9.xml	<a href="#">Intrastat control report ER format configuration</a>

Use the following procedure to import each file. Import the ER *data model* configuration of each ER solution in the tables above before you import the corresponding ER *format* configuration.

1. Open the **Organization administration > Electronic reporting > Configurations** page.
2. On the top of the page, select **Exchange**.
3. Select **Load from XML file**.
4. Select **Browse** to load the required XML file.
5. Select **OK** to confirm configuration's import.

The screenshot shows the 'Customer invoicing model' configuration page in Dynamics 365 Finance and Operations. The left sidebar shows a tree view with 'Customer invoicing model' selected. The main area displays configuration details for 'Customer invoicing model', including a description and a 'Microsoft' configuration provider. Below this, a 'Versions' table is shown with one entry: Version 2, Status Completed, Effective from (blank), Version created 1/15/2019 07:54:01 AM, and Description (blank). The interface also includes a search bar at the top and various action buttons like 'Change status', 'Delete', and 'Exchange'.

Alternatively, you can import the officially published ER format configurations from Microsoft Dynamics Lifecycle Service (LCS). For example, to complete this procedure you can import the latest version of the **Free text invoice (Excel)** ER format configuration. The corresponding ER data model and ER model mapping configurations will be imported automatically.

The screenshot shows the 'Shared asset library' in Dynamics 365 Lifecycle Services. The left sidebar lists various asset types, with 'GER Configuration (1705)' selected. The main area displays a table of 'GER Configuration files'. The table has columns for Name, Valid, Version, Scope, Status, Modified date, and Size. The row for 'Free text invoice (Excel).version.155.77' is highlighted. To the right, 'Additional details' for the selected configuration are shown, including the description 'Free text invoice (Excel)', Asset ID '027e4cd3-1d40-4aa0-9195-bc', Validation status 'Not validated', and creation/modification information.

For more information about importing ER configurations, see [Manage the ER configuration lifecycle](#).

# Enable Business document management

To start Business document management, you need to open the **Feature management** workspace and enable the **Business document management** feature.

Use the following procedure to enable Business document management functionality for all legal entities.

1. Open the **Feature management** workspace.
2. On the **New** tab, select the **Business document management** feature in the list.
3. Select **Enable now** to turn on the selected feature.
4. Refresh the page to access the new feature.

## NOTE

For more information about using the new document user interface in Business document management, see [New document user interface in Business document management](#).

The screenshot displays the 'Feature management' workspace. At the top, there are statistics: New (19), Not enabled (33), and Scheduled (0). Below these are buttons for 'Enable all' and 'Check for updates'. The main area shows a list of features with columns: Feature name, Enable date, Feature added, and Module. The 'Business document management' feature is selected and highlighted. A right-hand pane shows details for this feature, including a description: 'This feature provides a new Business document management workspace, a new form for visibility of pre-loaded business document templates, and redesigned business document templates.' There is also a 'Comments' section and a 'Disable' button at the bottom right of the pane.

Feature name	Enable date	Feature added	Module
Attachment recovery	8/1/2019	8/1/2019	System administration
Feature management enablement enhancements	8/1/2019	8/1/2019	System administration
Enable master plan setup wizard features.	8/1/2019	8/1/2019	Master planning
Keep GST tax document for confirmation journal	8/1/2019	8/1/2019	Tax
<b>Business document management</b>	8/2/2019	6/3/2019	Organization administration
Enable fixed exchange rate for reporting currency on sales ...	8/1/2019	5/9/2019	Shared AP and AR

For more information about activating new features, see [Feature management overview](#).

## Configure parameters

Use the information in the following sections to set up the basic parameters for Business document management.

### Prerequisites for parameter setup

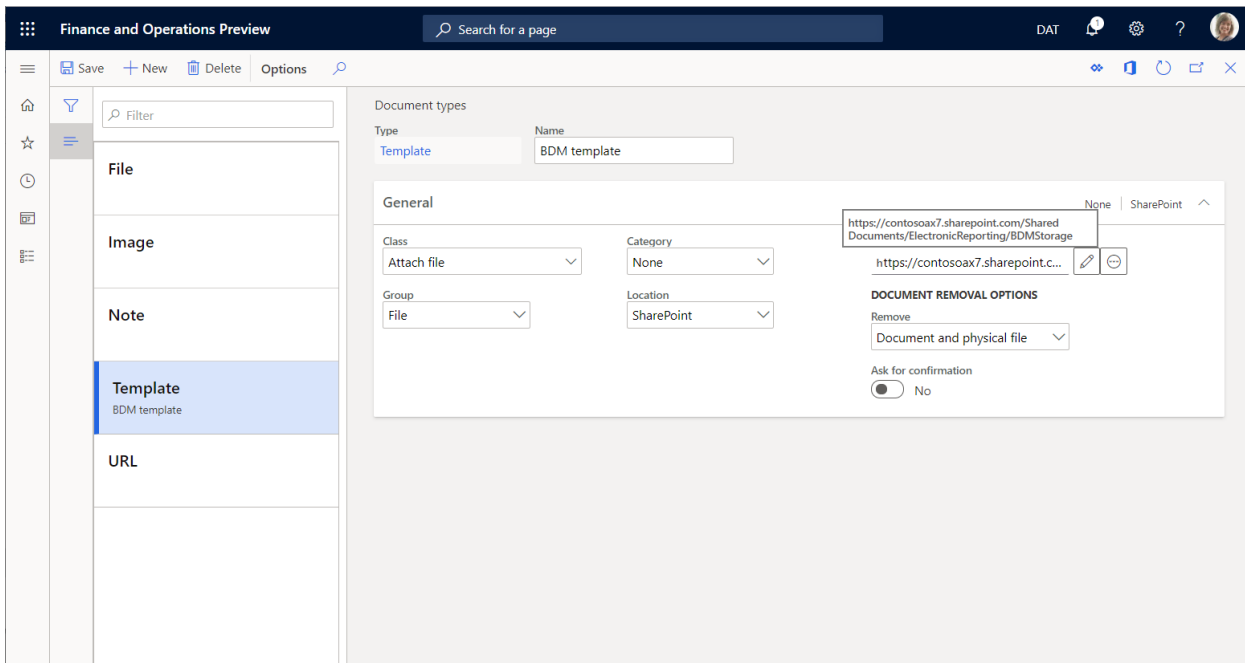
Before you can set up Business document management, you must set up the required document type in the Document management framework. This document type is used to specify a temporary storage of documents in Office formats (Excel and Word) that are used as templates for ER reports. The temporary storage template can be edited by using the Office desktop applications.

For this document type, the following attribute values must be selected.

ATTRIBUTE NAME	ATTRIBUTE VALUE
Class	Attach file

ATTRIBUTE NAME	ATTRIBUTE VALUE
Group	File
Location	SharePoint

For information about how to set up the required document management parameters and document types, see [Configure document management](#).



### Set up parameters

Basic Business document management parameters can be set up on the **Business document parameters** page. Only specific users can access the page. This includes:

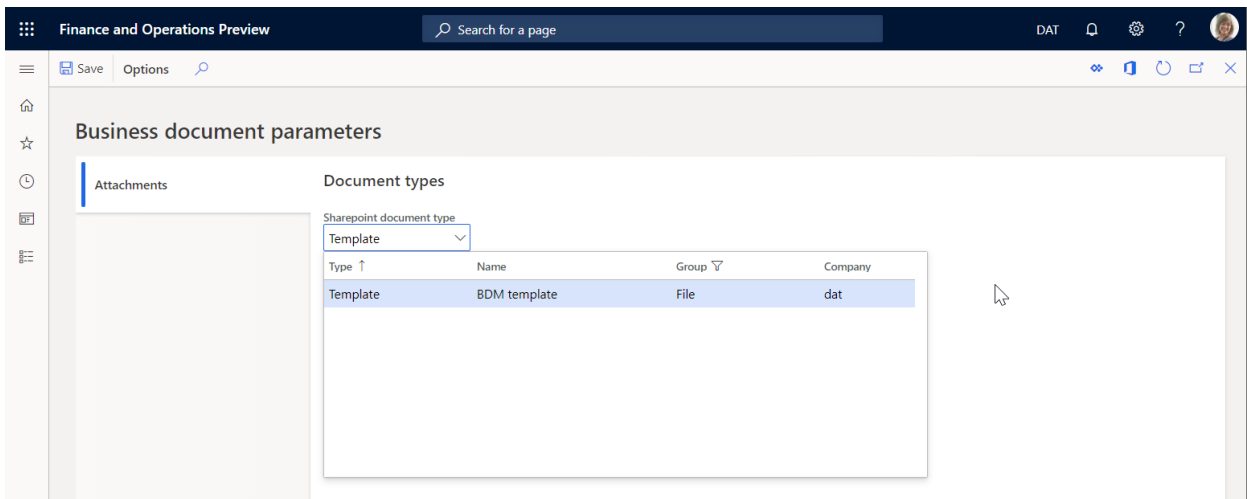
- Users who are assigned to the **System administrator** role.
- Users who are assigned to any role that is configured to perform the duty, **Maintain Business document management parameters** (AOT name **ERBDMaintainParameters**).

Use the following procedure to set up the basic parameters for all legal entities.

1. Sign in as a user with access to the **Business document parameters** page.
2. Go to **Organization administration > Electronic reporting > Business document management > Business document parameters**.
3. On the **Business document parameters** page, on the **Attachments** tab, in the **SharePoint document type** field, define the document type that should be used to temporarily store templates in Office formats while they are edited using the Office desktop applications.

#### NOTE

Only document types that are configured using a SharePoint location are available for this parameter.



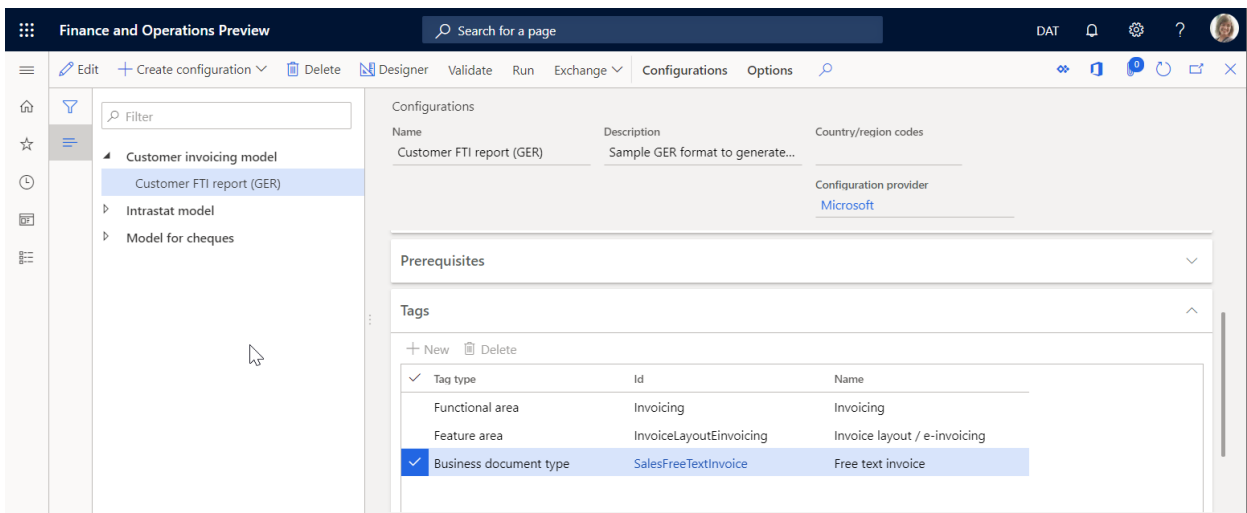
The selected document type is company-specific and will be used when the user is working with Business document management in the company for which the selected document type is configured. When the user is working with Business document management in another company, the same selected document type will be used if one has not been configured for this company. When a document type has been configured, it will be used instead of the one selected in the **SharePoint document type** field.

**NOTE**

The **SharePoint document type** parameter defines a SharePoint folder as temporary storage for templates that are editable using either Microsoft Excel or Word. You need to set up this parameter if you plan to use these Office desktop applications for editing templates. For more information, see [Edit a template in the Office desktop application](#). You can keep this parameter blank if you plan to modify the template by only using the functionality in Microsoft 365. For more information, see [Edit a template in Microsoft 365](#).

## Configure access permissions

By default, when access to Business document management permissions is not enabled, every user with access to the Business document management workspace will see all of the ER solution templates that are available. The Business document management workspace will show only those templates that reside in ER format configurations and that are marked by a **Business document type** tag.



The list of templates available in the Business document management workspace can be restricted by configuring access permissions. This may be important when different templates are used to produce business documents for different business domains (functional areas), and you want to allow specific users access to different templates for editing in the Business document management workspace.

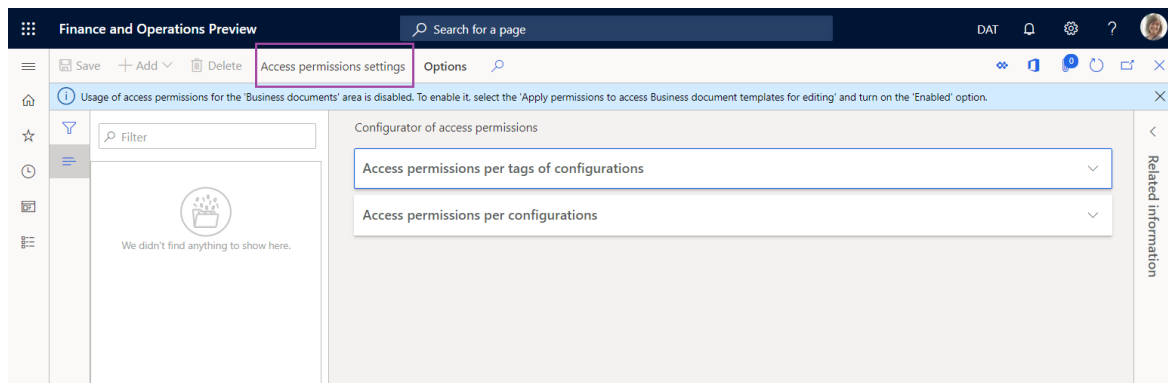
Business document management access permissions can be set on the **Configurator of access permissions**. Only the following users can access the page:

- Users assigned to the **System administrator** role.
- Users assigned to any other role that is configured to perform the duty, **Configure permissions to access Business document templates for editing** (AOT name ERBDTemplatesSecurity).

Use the following procedure to set up the access Business document management permissions for all legal entities.

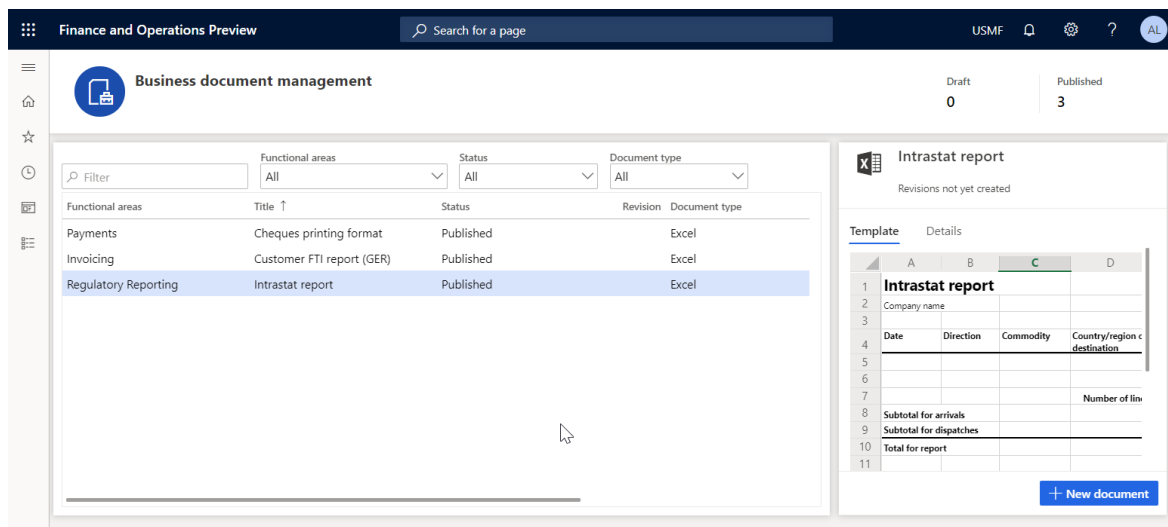
1. Sign in as a user with access to the **Configurator of access permissions** page.
2. Go to **Organization administration > Electronic reporting > Business document management > Manage access permissions**.

Pay attention to the notification informing you that the usage of access permissions for Business document management is currently not enabled.



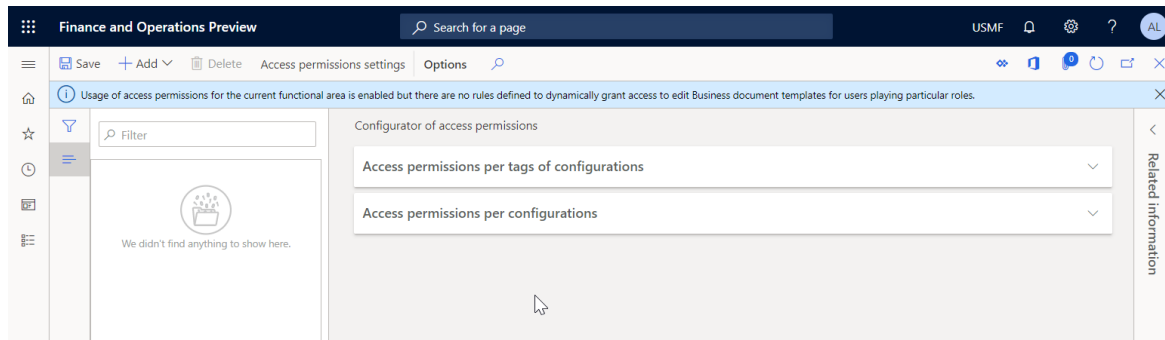
With this setting, every user assigned to any security role that is configured to perform the **Manage Business document templates** (AOT name ERBDManageTemplates) duty is able to open the Business document management workspace and can edit any template that is available.

The following graphic shows what is available in the Business document management workspace for users assigned to the **Accounts receivable clerk** role. With the current access permissions setting, the user can edit business document templates from different functional areas including invoicing, regulatory reporting, and payments.



3. On the **Configurator of access permissions** page, select **Access permissions setting**.
4. In the **Settings of access permissions to edit templates** dialog box, enable the **Apply configured access permissions** option.

5. Select **OK** to confirm that Business document management access permissions have been enabled.



6. Select **Add** to enter a new business role for which permissions to access Business document management templates must be configured.

7. In the **Security roles** dialog box, select the **Accounts receivable clerk** role and then select **OK** to confirm the role selection.

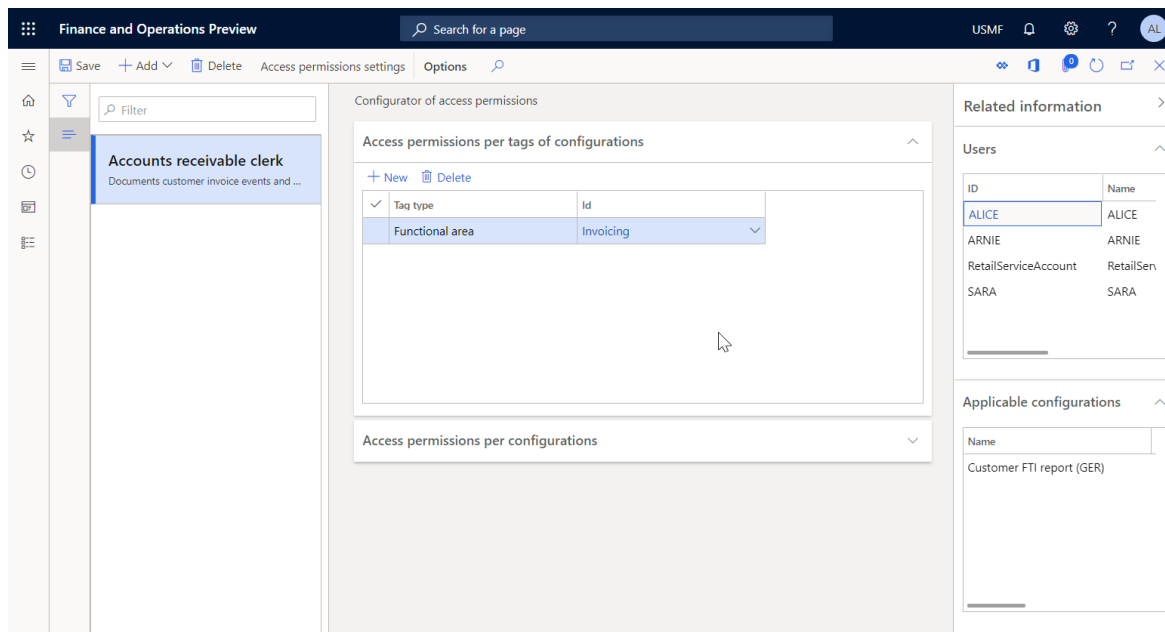
8. On the **Access permissions per tags of configurations** tab, select **New**.

9. In the **Tag type** field, select **Functional area**, and in the **ID** field, select **Invoicing**.

10. Select **Save** to store configured access permissions for the selected role.

The current setting means that for any user who is assigned to the **Accounts receivable clerk** role and performing the duty, **Manage Business document templates** (AOT name **ERBDManageTemplates**), ER format configuration templates that have the **Invoicing** value for the **Functional area** tag will be available to edit in the Business document management workspace.

11. Switch the **Related information** pane from the right side of the current page. The **Related information** pane shows how the configured access permissions will be applied, including what ER configuration templates will be available for users that are assigned to the **Accounts receivable clerk** role.



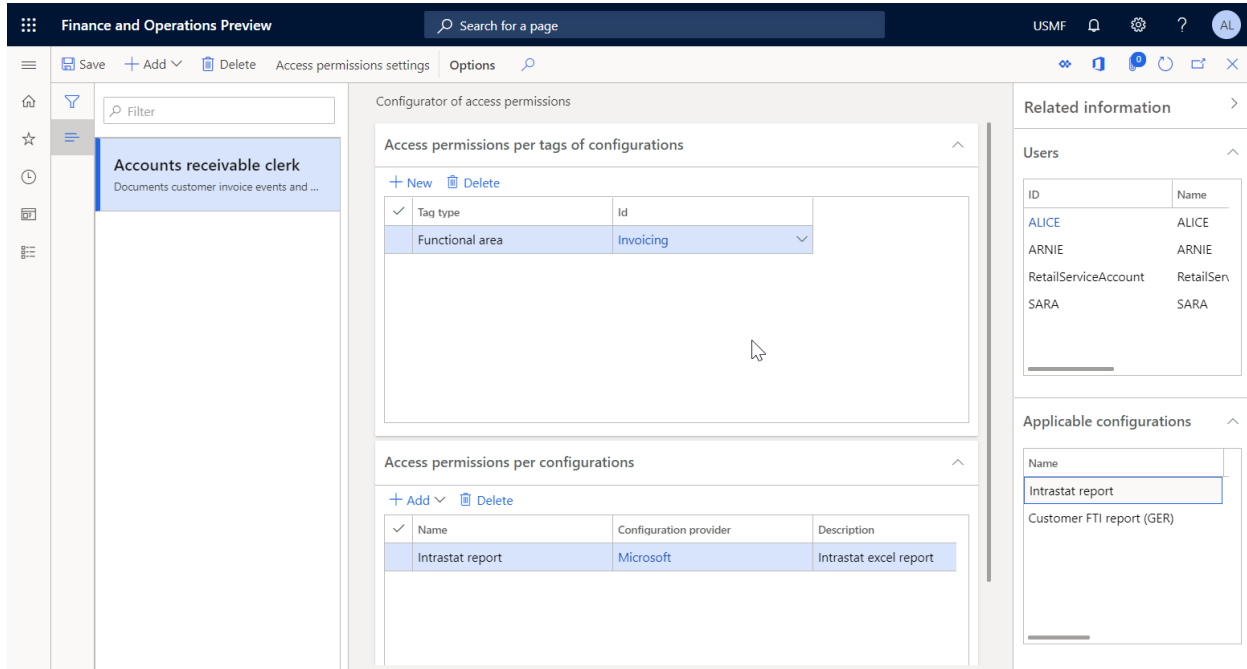
12. On the **Access permissions per configurations** tab, select the **Add** option.

13. In the **Select configuration** dialog box, mark the **Intrastat report** ER format configuration.

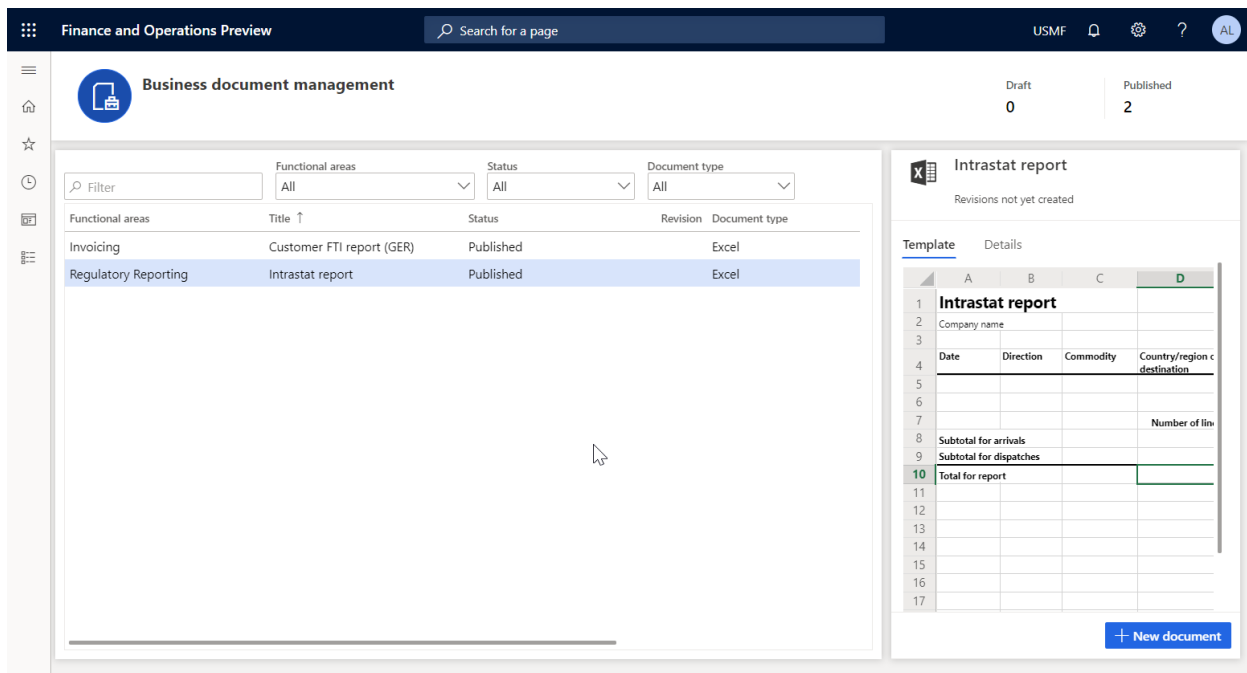
14. Select **OK** to confirm the entry of the selected configurations and then select **Save** to store the configured access permissions for the selected role.

The current setting means that for any user assigned to the **Accounts receivable clerk** role and performing the duty, **Manage Business document templates** (AOT name **ERBDManageTemplates**), the following templates will be available to edit in the Business document management workspace:

- Templates that have the value, **Invoicing** for the **Functional area** tag.
- Templates from ER format configurations that are listed on the **Access permissions per configurations** tab (templates from the **Intrastat report** format configuration of the **Statutory reporting** domain in this example).



The following graphic shows what the Business document management workspace provides to a user assigned to the **Accounts receivable clerk** role. With the current Business document management access permissions setting, the user can edit business document templates from the **Invoicing** domain and the **Intrastat report** ER format configuration. Templates from the **Payments** domain are not accessible for the **Accounts receivable clerk** role.





## NOTE

The **Access permissions per configurations** rules are stored by using the unique identification ID of an ER format configuration. This means that these rules will not be deleted when an ER configuration that refers to them are deleted. When you import deleted configurations back to this instance, these rules will refer to them again. There is no need to set up the rules again after the deleted configurations are imported again.

## Use Business document management to edit templates

Business users can access business document templates for editing in the Business document management workspace. Only the following users can access the Business document management workspace:

- Users assigned to the role, **System administrator**.
- Users assigned to any role that is configured to perform the duty, **Manage Business document templates** (AOT name **ERBDManageTemplates**).

Use the following procedure to edit free text invoice templates in the Business document management workspace. Before you complete this procedure, you must have completed all of the preceding procedures in this topic.

1. Sign in as a user with access to the Business document management workspace.
2. Open the Business document management workspace.

When the **Office-like UI experience for Business document management** feature is turned off in the **Feature management** workspace, the main grid in the **Business document management** workspace shows the following templates:

- Templates that are owned by your ER configuration provider (that is, the provider that is currently marked as active in the **Electronic reporting** workspace). After you select one of these templates, you can select **Edit template** to start or continue to edit it.
- Templates that are owned by other ER configuration providers. After you select one of these templates, you can select **New document** to create a copy of it that is owned by your ER configuration provider, and then start to edit the copy.

The screenshot shows the Business document management workspace interface. At the top, there is a navigation bar with 'Finance and Operations Preview', a search bar, and user information 'USMF'. Below the navigation bar, the main workspace is titled 'Business document management'. On the left, there is a sidebar with navigation icons. The main area contains a table of templates with filters for Functional areas, Status, and Document type. The table lists three templates: 'Cheques printing format', 'Customer FTI report (GER)', and 'Intrastat report'. The 'Intrastat report' template is selected and highlighted. To the right of the table, there is a preview of the 'Intrastat report' template, showing a grid with columns for Date, Direction, Commodity, and Country/region of destination. The preview also shows a 'Number of lines' column and a 'New document' button at the bottom.

Functional areas	Title	Status	Revision	Document type	Modified date and time	Modified by
Payments	Cheques printing format	Published		Excel	8/2/2019 07:50:56 AM	Admin
Invoicing	Customer FTI report (GER)	Published		Excel	8/2/2019 06:21:34 AM	Admin
Regulatory Reporting	Intrastat report	Published		Excel	8/2/2019 07:47:23 AM	Admin

**Intrastat report**  
Revisions not yet created

	A	B	C	D
1	<b>Intrastat report</b>			
2	Company name			
3				
4	Date	Direction	Commodity	Country/region of destination
5				
6				
7				Number of lines
8	Subtotal for arrivals			
9	Subtotal for dispatches			
10	Total for report			
11				
12				

[+ New document](#)

The **Template** tab presents the content of the selected template. Select the **Details** tab to review details of the selected template as well as details of an ER format configuration this template resides in. Notice that all of the templates have a status of **Published**, and contain no details in the **Revision** column. This means that these templates are not currently being edited.

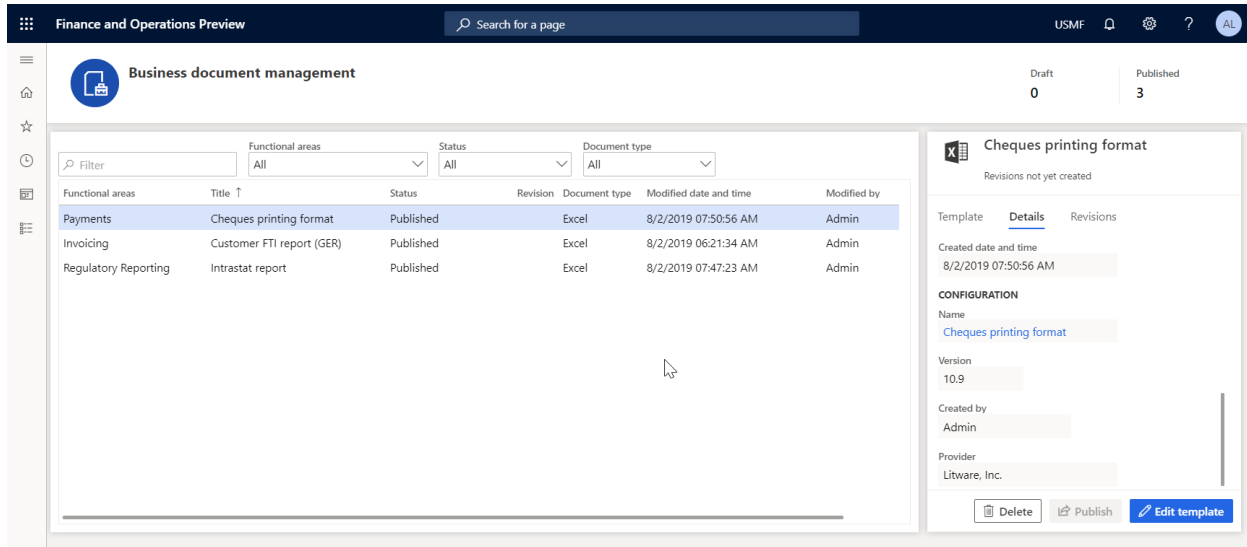
When the **Office-like UI experience for Business document management** feature is turned on in the

**Feature management** workspace, the main grid in the **Business document management** workspace shows templates that are owned by your ER configuration provider (that is, the provider that is currently marked as active in the **Electronic reporting** workspace). After you select one of these templates, you can select **Edit template** to start or continue to edit it.

To work with templates that are owned by other ER configuration providers, select **New document** to create a copy of the template that is owned by your ER provider. You can then start to edit the copy. For more information, see [New document user interface in Business document management](#).

### Initiate editing templates owned by your configuration provider

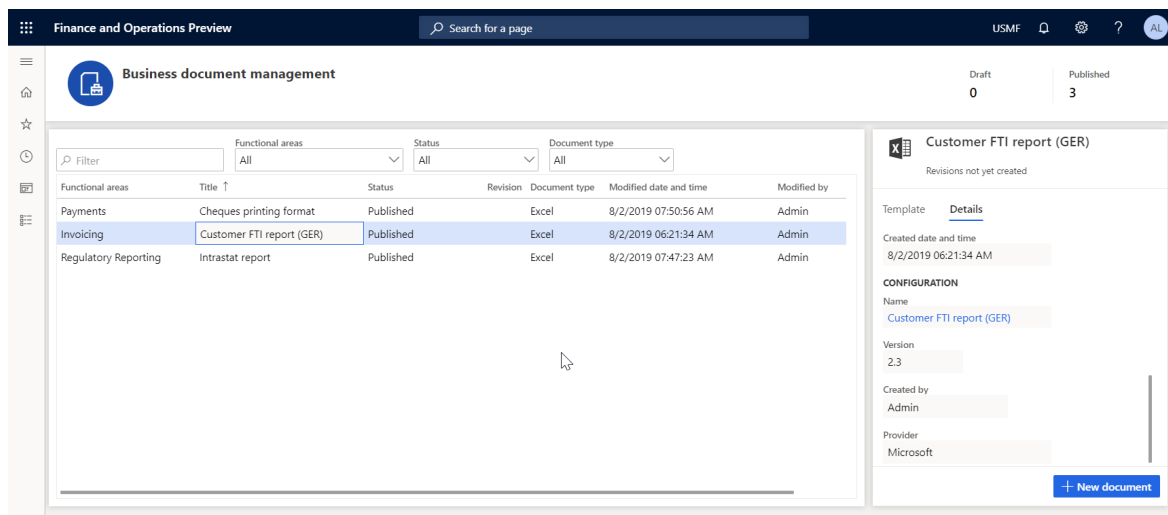
1. In the Business document management workspace, select the **Cheques printing format** template in the list.
2. Select the **Details** tab.



The **Edit template** option is available for the selected template. This option is always available for a template in an ER format configuration that is owned by the active ER configuration provider (**Litware, Inc.** in this example). When **Edit template** is selected, the existing template from the draft version of the underlying ER format configuration will be available to edit.

### Initiate editing templates owned by other providers

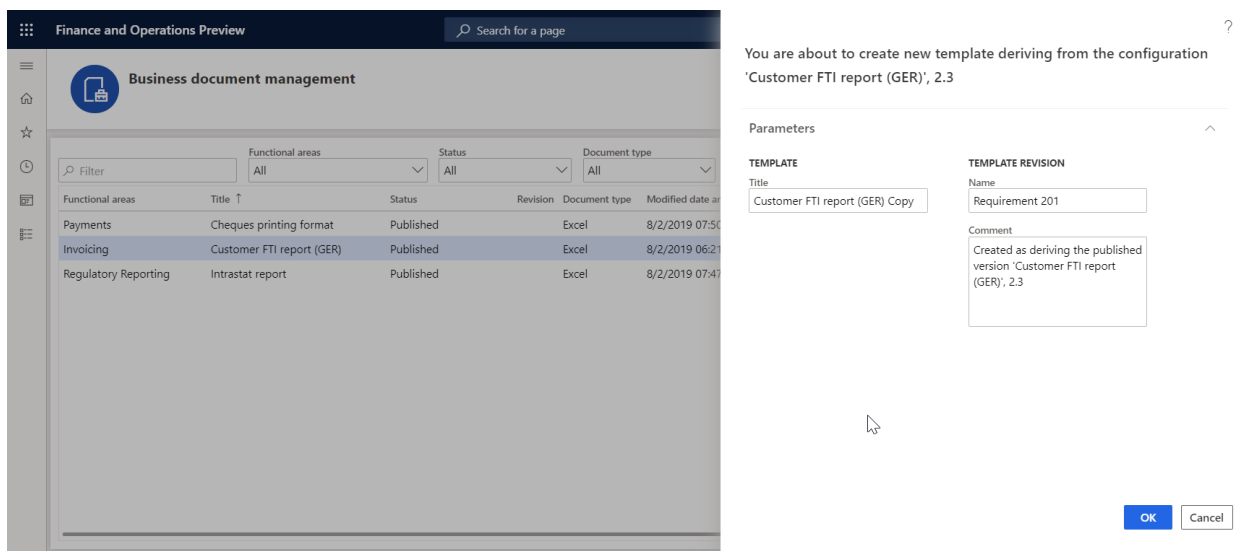
1. In the Business document management workspace, select the document that you want to use as a template.



2. Select **New document**, and in the **Title** field, change the title of the editable template if needed. The text will be used to name the ER format configuration that is automatically created. Note that the draft version of this configuration (**Customer FTI report (GER) Copy**) that will contain the edited template will

automatically be marked to run this ER format for the current user. At the same time, the non-modified original template from the base ER format configuration will be used to run this ER format for any other user.

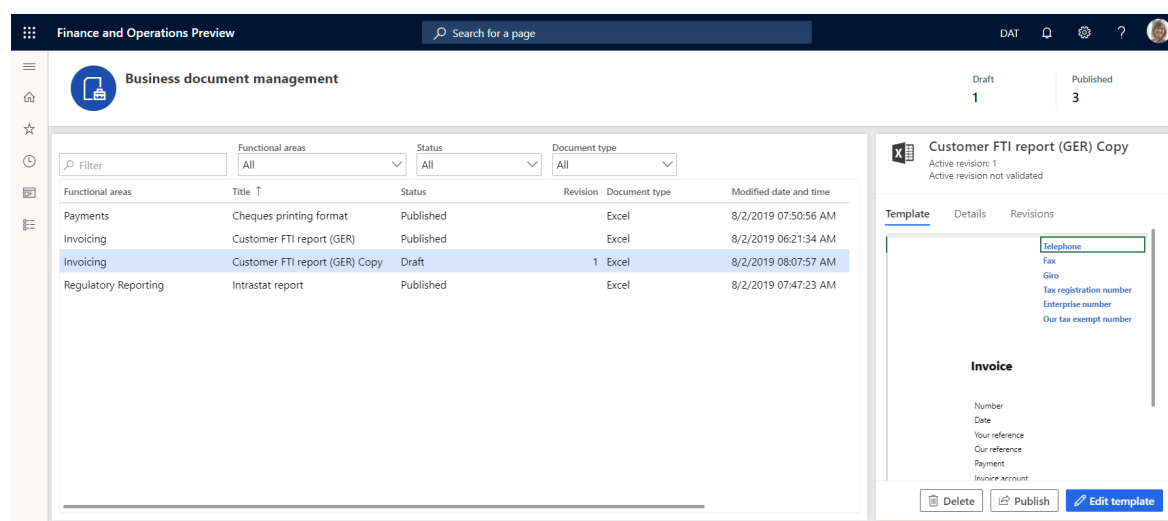
3. In the **Name** field, change the name of the first revision of the editable template that will be created automatically.
4. In the **Comment** field, change the comment for the automatically created revision of the editable template.
5. Select **OK** to confirm the start of the editing process.



The **New document** option is always available for a template in an ER format configuration provided by current and another provider (Microsoft in this example) that doesn't have any revision. The edited template will then be stored in a new ER format configuration that is automatically generated.

### Start editing a template

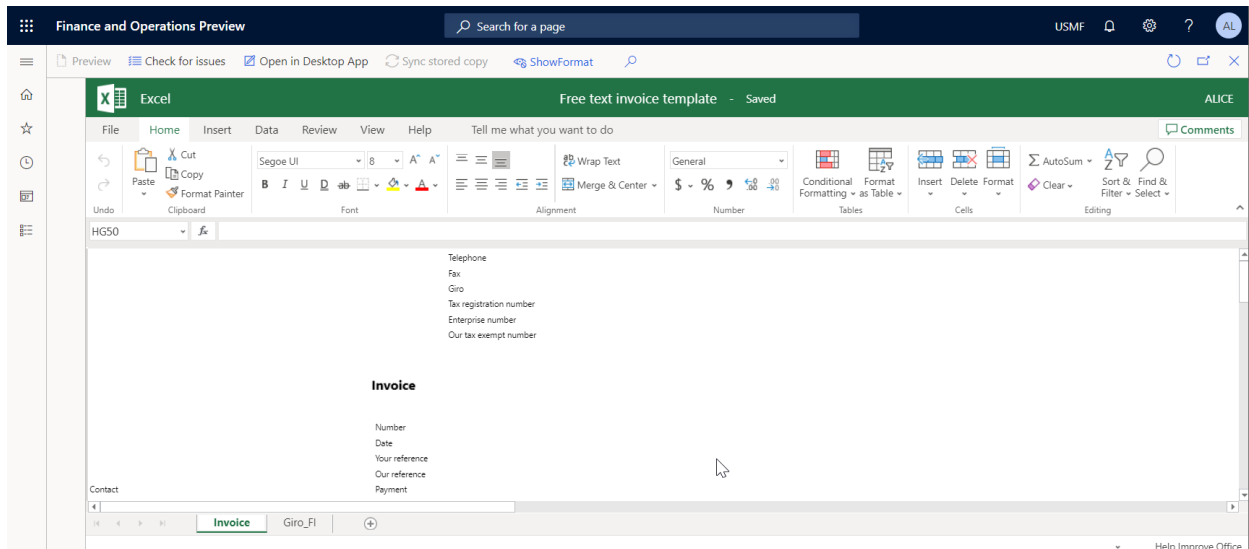
1. From the selected template, select **Edit document**.
2. In the **Name** field, change the name of the first revision of the editable template that will be created automatically.
3. In the **Comment** field, change the remark for the automatically created revision of the editable template.



4. Select **OK** to confirm the start of the editing process.

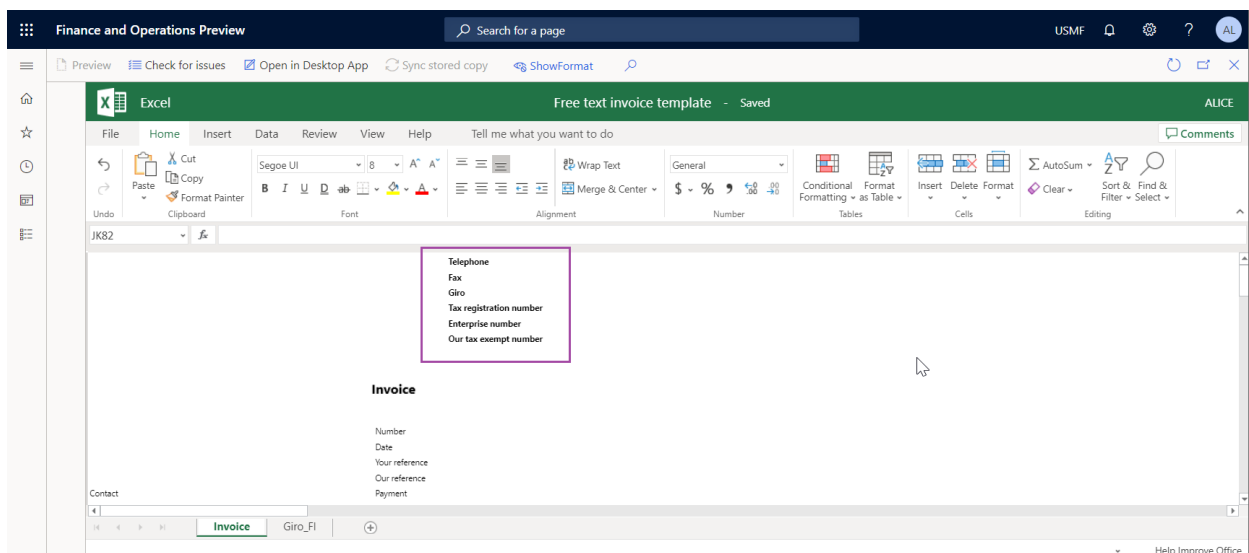
The **BDM template editor** page will open. The selected template will be available for online editing by using

## Microsoft 365.



### Edit a template in Microsoft 365

You can modify the template using Microsoft 365. For example, in Office online, change the font of the field prompts in the template header from **Regular** to **Bold**. These changes are automatically stored in the editable template that is stored in the primary template's storage (by default, the Azure blob storage). This is configured for the ER framework.

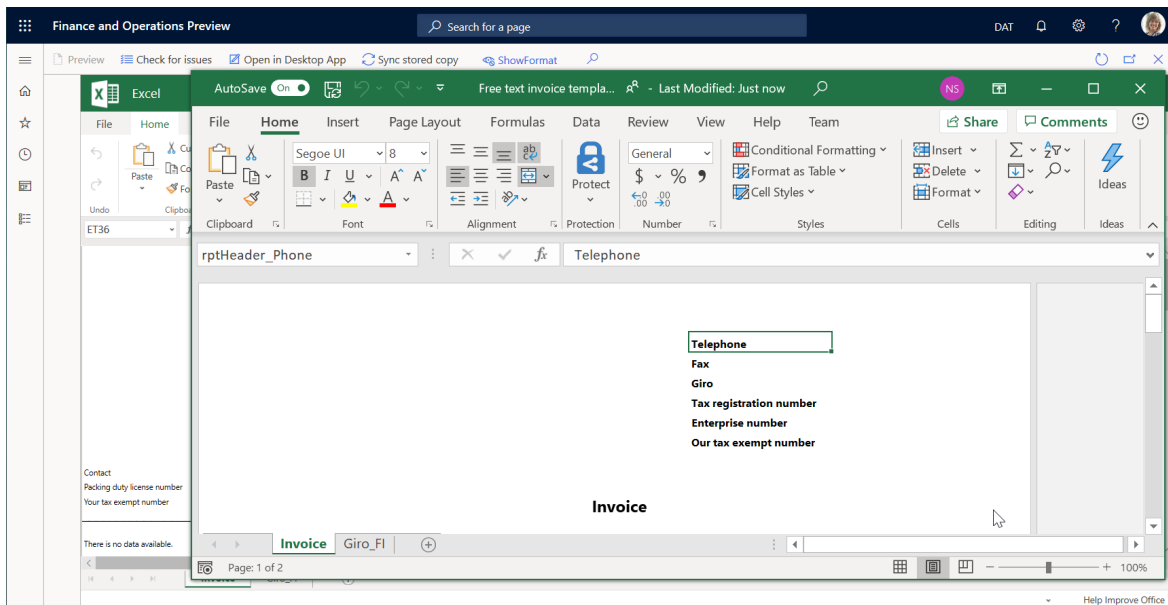


### Edit a template in the Office desktop application

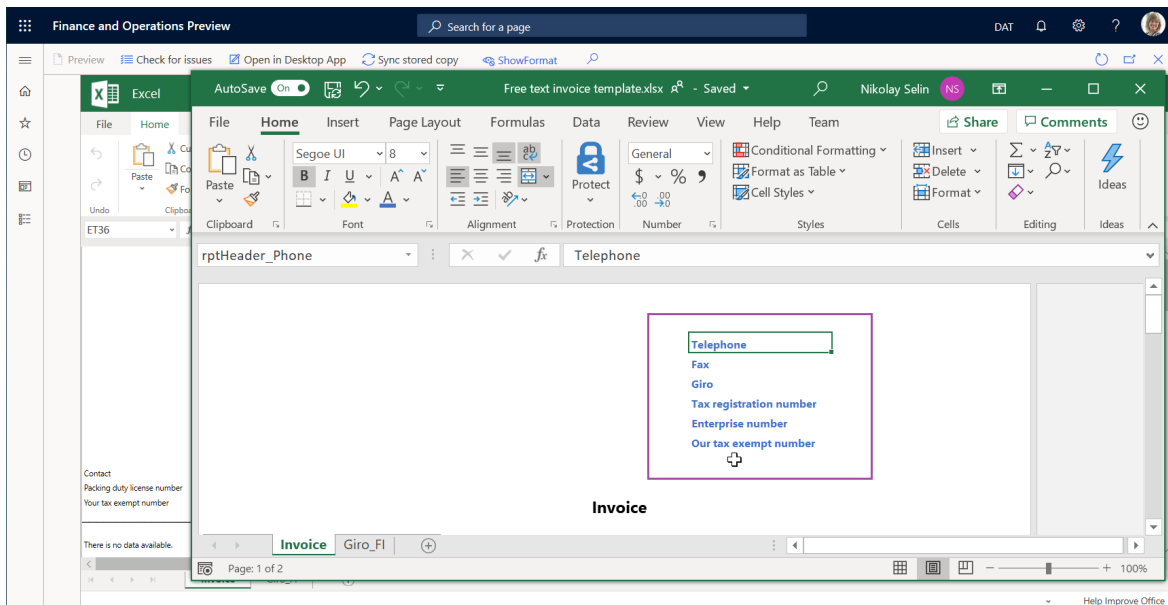
#### NOTE

This function is only available when the **SharePoint document type** parameter is properly configured. For more information, see [Configure parameters](#).

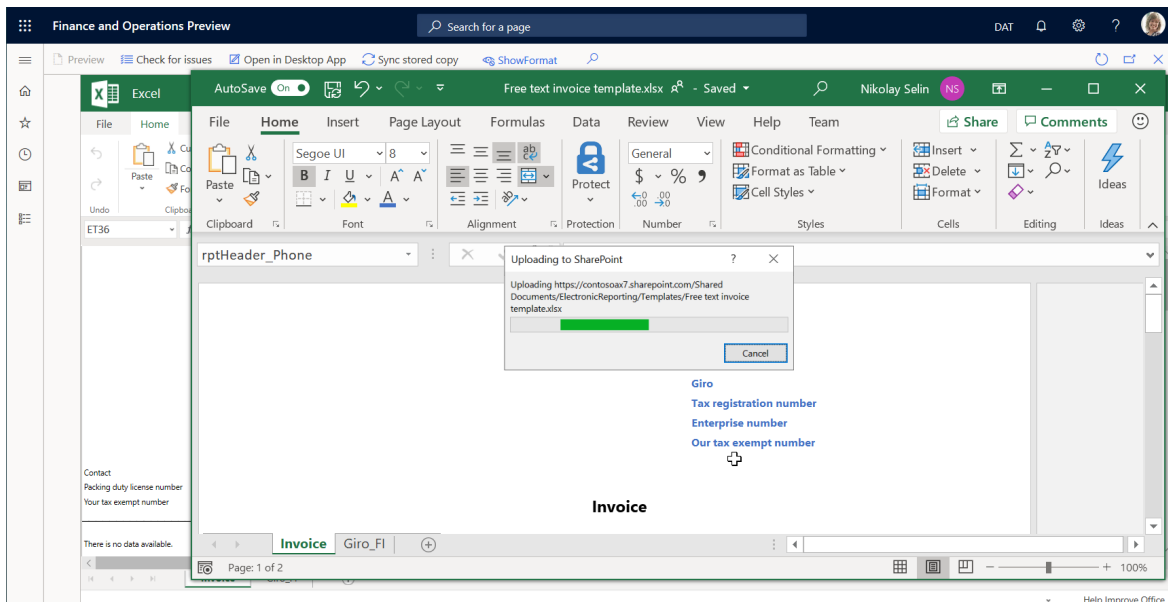
1. Select the **Open in Desktop App** option to modify the template by using the functionality of the Office desktop application (Excel in this example). The editable template is copied from the permanent storage to the temporary storage configured in the Business document management parameters as a SharePoint folder.
2. Confirm that you want to open the template from the temporary file storage in the Office desktop Excel application.



3. Modify the template. For example, change the font of the fields prompts in the template header by updating color from Black to Blue.



4. Select Save in the Excel desktop application to store the template changes in the temporary storage.



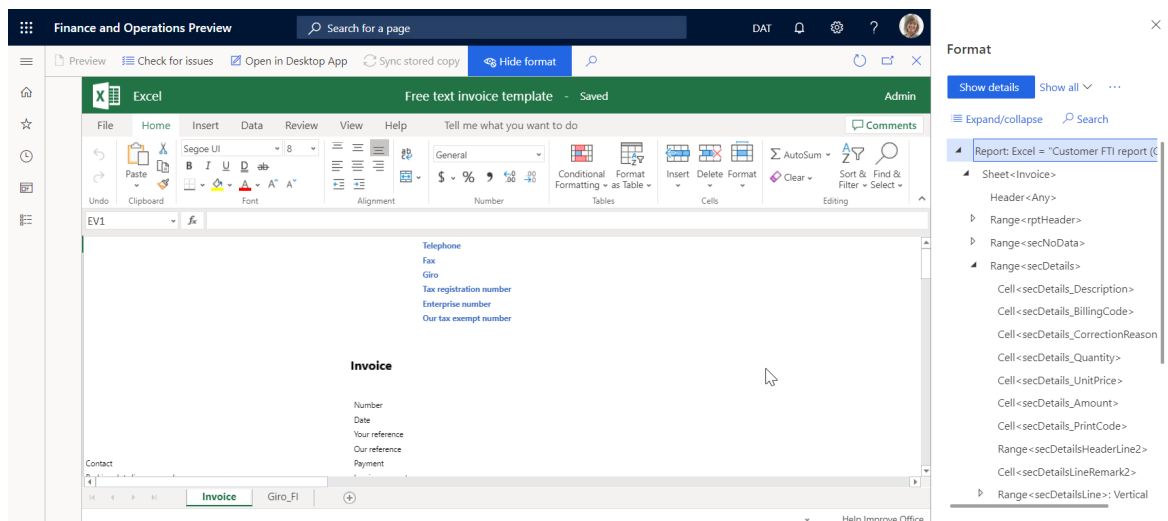
- Close the Excel desktop application.
- Select **Sync stored copy** to synchronize the temporary template storage to the permanent template storage.

#### NOTE

The copy of the editable template in the temporary file storage is kept for only the current session of template editing. When you finish this session by closing the **BDM template editor** page, this copy will be deleted. If you adjusted the template in the temporary file storage and did not select **Sync stored copy**, when you try to close the **BDM template editor** page, a message will ask whether you want to store introduced changes. Select **Yes** if you want to save your changes to the template in the permanent file location.

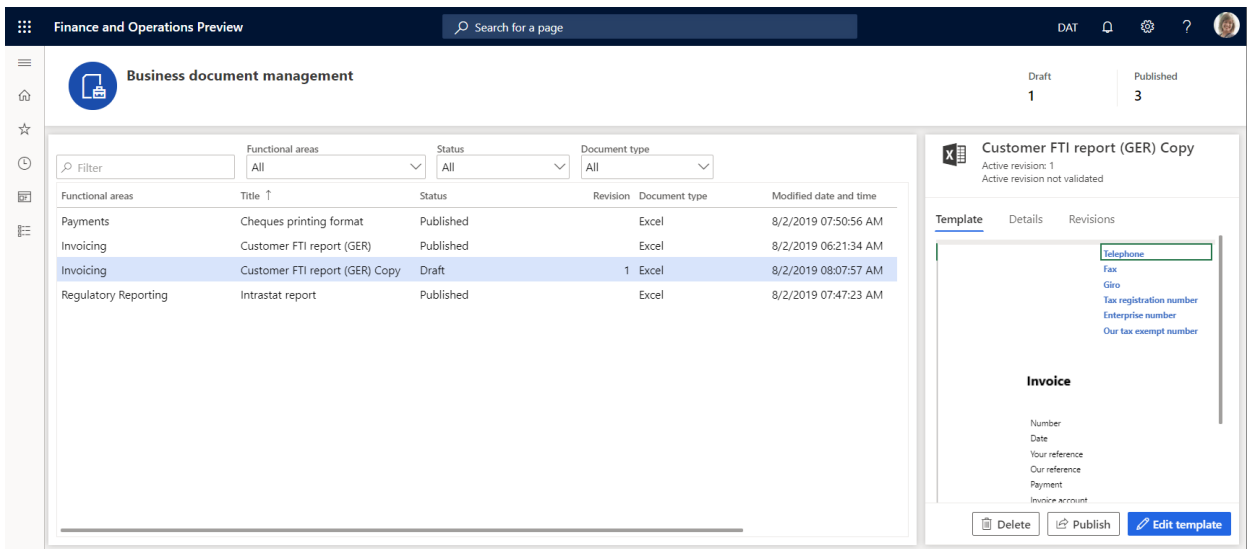
#### Validate a template

- On the **BDM template editor** page, select **Check for issues** to validate the modified template against the underlying ER format configuration. Follow the recommendations (if any) to align the template with the structure of the format from the base ER format configuration.
- Select **Show format** to view the current structure of the format from the base ER format configuration that must be aligned with the editable template.
- Select **Hide format** to close the pane.



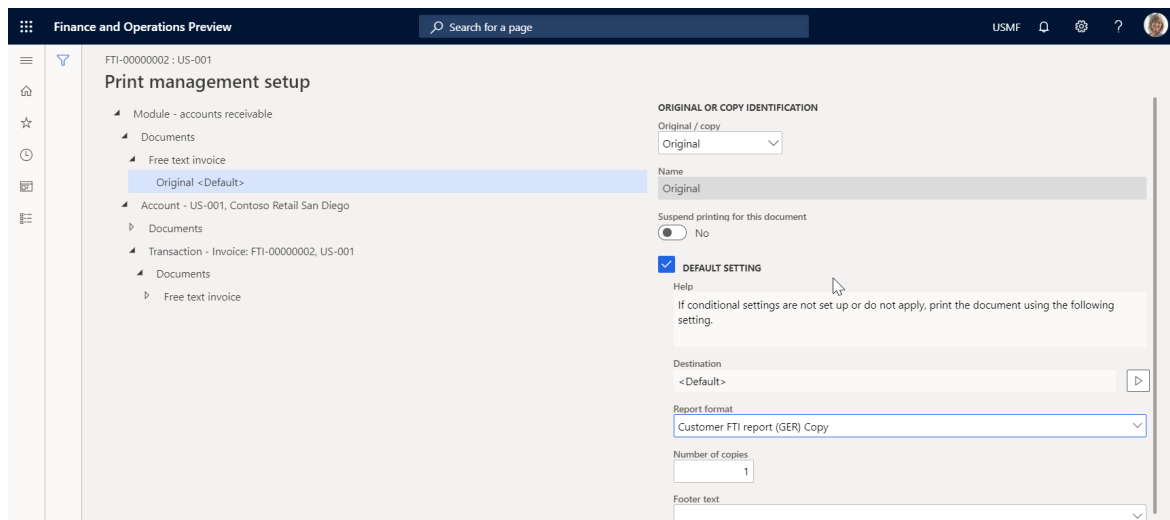
- Close the **BDM template editor** page.

The updated template is shown on the **Template** tab. Notice that the status of the edited template is now **Draft** and the current revision is no longer empty. This means that the process of this template's editing has been started.

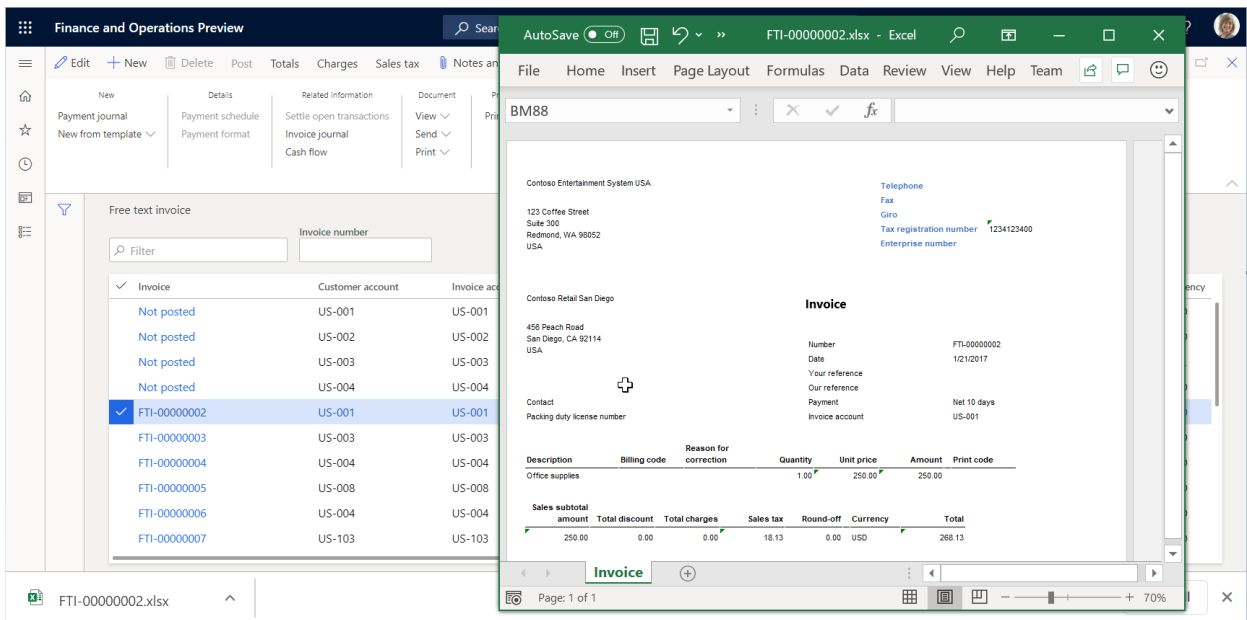


### Test the modified template

1. In the application, change to the company, USMF.
2. Go to **Accounts receivable > Invoices > All free text invoices**.
3. Select the FTI-00000002 invoice, and then select **Print management**.
4. Select the **Module - accounts receivable > Documents > Free text invoice > Original document** level to specify the scope of invoices for processing.
5. In the **Report format** field, select the **Customer FTI report (GER) Copy** ER format for the specified document level.



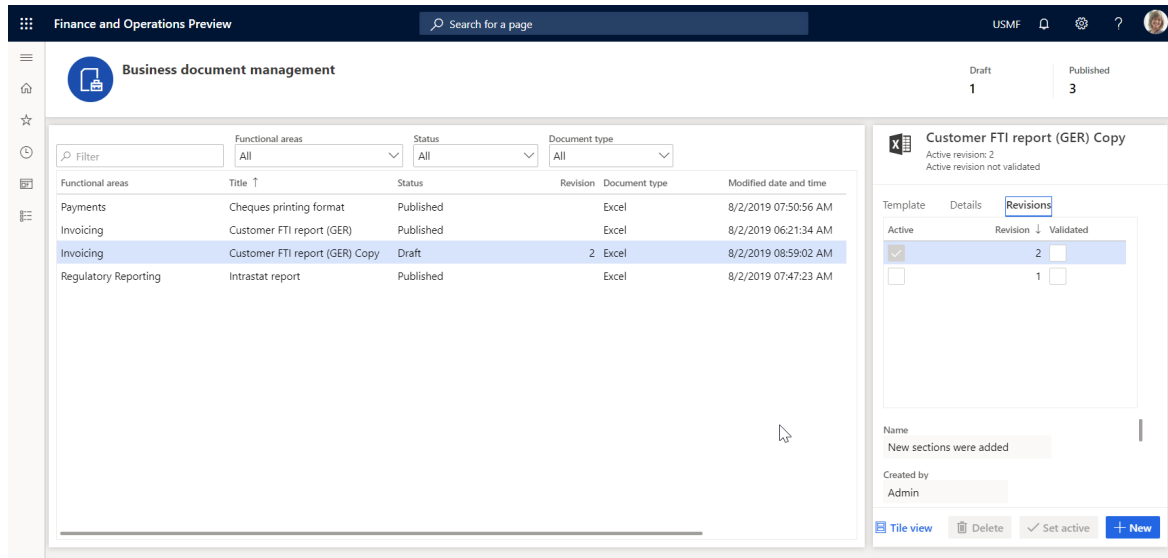
6. Press **Escape** to close the current page.
7. Select **Print**, and then select **Selected**.
8. Download the document and open it using the Excel desktop application.



The modified template is used to generate the free text invoice report for the selected item. To analyze how this report is affected by the changes that you introduced to the template, you can run this report in one application session right after you modified the template in another application session.

### Create an alternative template revision

1. Open the **BDM template editor** page and select the **Customer FTI report (GER) Copy** template.
2. On the **Revisions** tab, select **New**.
3. If needed, in the **Name** field, change the name of the second revision and base it on the currently active first revision.
4. If needed, in the **Comment** field, change the remark for the automatically created revision of the editable template.



You created a new revision of your template that has been stored in the permanent template's storage. Now you can continue editing the template of the second revision that is currently selected as active.

5. Select the first revision and then select **Set active**. You can select another revision as active if at any time you want to return to that revision of the template.
6. Select the second revision, and then select **Delete**.
7. Select **OK** to confirm that you want to delete the selected revision. You can delete any of the non-active



revisions when they are no longer needed.

### Delete a modified template

1. On the **BDM template editor** page, select the **Template** tab.
2. Select **Delete**.
3. If you select **OK** to confirm deletion, the **Customer FTI report (GER) Copy ER** format with the modified template will be deleted. Select **Cancel** to explore other options.

### Revoke changes of template

When you edit the template from an ER format that is owned by the current active provider, you will be offered the option to revoke changes introduced for the template.

The screenshot shows the 'Business document management' interface. On the left, there is a navigation pane with icons for home, star, clock, and list. The main area displays a table of templates with filters for Functional areas, Status, and Document type. The table lists several templates, including 'Cheques printing format' which is currently in Draft status. On the right, a preview window shows the 'Cheques printing format' template with a grid and a preview of the generated document.

Functional areas	Title	Status	Revision	Document type	Modified date and time
Payments	Cheques printing format	Draft	1	Excel	8/2/2019 09:02:04 AM
Invoicing	Customer FTI report (GER)	Published		Excel	8/2/2019 06:21:34 AM
Invoicing	Customer FTI report (GER) Copy	Draft	1	Excel	8/2/2019 09:02:28 AM
Regulatory Reporting	Intrastat report	Published		Excel	8/2/2019 07:47:23 AM

1. On the **BDM template editor** page, select the **Template** tab.
2. Select **Undo**.
3. If you select **OK** to revoke the changes introduced for the template, the modified template will be replaced by the original template and all changes will be removed. When you revoke changes to the template, you will be able to delete the template. Select **Cancel** to explore other options.

### Publish a modified template

1. On the **BDM template editor** page, on the **Template** tab, select **Publish**.
2. If you select **OK** to confirm publishing, the draft version of the derived **Customer FTI report (GER) Copy ER** format that contains the modified template will be marked as completed. The modified template becomes available for other users. The completed versions of this ER format will keep only the last active revision of your template. Other revisions will be deleted. Select **Cancel** to explore other options.

## Frequently asked questions

### I selected **Edit document**, but instead of going to the **BDM template editor** page in **Finance**, I was sent to the **Microsoft 365** webpage.

This issue is a known issue that involves Microsoft 365 redirection. It occurs when you sign to Microsoft 365 for the first time. To work around this issue, select **Back** in your browser to return to the previous page.

### I understand how to edit a template by using **Microsoft 365** in the first application session, and how to use the template in the second application session and adjust the template to see how my changes affect the generated business document. Can I use the **Office desktop** application in the same way?

Yes, you can. In the first application session, select **Open in Desktop App**. Your template will be stored in the temporary file storage and opened in the Office desktop application. Next, complete the following steps to preview your template changes in the generated business document:

1. Make changes in the template by using the Office desktop application.
2. Select **Save** in the Office desktop application.
3. On the **BDM template editor** page of the first application session, select **Sync stored copy**.
4. Execute this template ER format in the second application session.

**When I select Open in Desktop App, I receive the following error message: "Value cannot be null. Parameter name: externalId." How do I work around this issue?**

Most likely you signed in to the current instance of the app of the Azure AD domain which differs from the Azure AD domain that was used to deploy this instance. Because the SharePoint service, which is used to store templates for making them available for editing by using the Office desktop applications, belongs to the same domain, we have no permissions to access the SharePoint service. To resolve this issue, sign in to the current instance using the credentials of a user with the correct Azure AD domain.

## Additional resources

[Electronic reporting \(ER\) overview](#)

[ER Design a configuration for generating reports in OPENXML format \(November 2016\)](#)

[Design ER configurations to generate reports in Word format](#)

[Embed images and shapes in documents that you generate by using ER](#)

[Configure Electronic reporting \(ER\) to pull data into Power BI](#)

## List of ER configurations that have been released in Finance to support configurable business documents

The [list](#) of ER configurations for Finance is constantly updated. Open the [Global repository](#) to review the list of ER configurations that are currently supported. You can [filter](#) the Global repository to review the list of ER configurations that are used to support configurable business documents.

The screenshot shows the 'Configuration repository' interface in Finance and Operations. The left sidebar lists various configuration categories, with 'Picking list (Excel)' selected. The main area displays a table of configurations and a section for versions.

Configuration name	Description	Configuration provider	Location
Container contents model mapping		Microsoft	Global Repository
Load list (Excel)		Microsoft	Global Repository
Load list (Word)	Initial version	Microsoft	Global Repository
Load list model mapping (RDP)		Microsoft	Global Repository
Picking list (Excel)		Microsoft	Global Repository
Picking list (CZ) (Excel)		Microsoft	Global Repository

Version	Status	From date	Supported until	Version created
9.12	Shared			11/20/2019 3:49:49 PM
9.10	Shared			10/18/2019 10:20:32 AM

The following table shows the list of ER configurations that support configurable business documents and that have been released in Finance up until December 2020.

DATA MODEL CONFIGURATION	FORMAT CONFIGURATIONS
Bill of lading model	Bill of lading (Excel)
	Bill of lading (Word)
Certificate of origin model	Certificate of origin (Excel)
	Certificate of origin (Word)
Invoice model	Customer Debit and Credit Note (Excel)
	Customer Debit and Credit Note (Word)
	Free text invoice (Excel)
	Free text invoice (Excel) (BH)
	Free text invoice (FR) (Excel)
	Free text invoice (LT) (Excel)
	Free text invoice (LV) (Excel)
	Free text invoice (PL) (Excel)
	Free text invoice (CZ) (Excel)
	Free text invoice (EE) (Excel)
	Free text invoice (HU) (Excel)
	Free text invoice (TH) (Excel)
	Free text invoice (Word)
	Project contract line items (Excel)
	Project contract line items (CZ) (Excel)
Project contract line items (Excel) (BH)	
Project contract line items (HU) (Excel)	
Project contract line items (LT) (Excel)	
Project contract line items (PL) (Excel)	
Project contract line items (Word)	
Project customer retention release (Excel)	

DATA MODEL CONFIGURATION	FORMAT CONFIGURATIONS
	Project customer retention release (CZ) (Excel)
	Project customer retention release (HU) (Excel)
	Project customer retention release (LT) (Excel)
	Project customer retention release (PL) (Excel)
	Project customer retention release (TH) (Excel)
	Project customer retention release (Word)
	Project invoice (Excel)
	Project Invoice (Word)
	Project invoice (AE) (Excel)
	Project invoice (CZ) (Excel)
	Project invoice (Excel) (BH)
	Project invoice (HU) (Excel)
	Project invoice (JP) (Excel)
	Project invoice (LT) (Excel)
	Project invoice (PL) (Excel)
	Project invoice (TH) (Excel)
	Project invoice full (MY) (Excel)
	Project invoice simple (MY) (Excel)
	Project manage invoice (Excel)
	Project manage invoice (CZ) (Excel)
	Project manage invoice (Excel) (BH)
	Project manage invoice (HU) (Excel)
	Project manage invoice (JP) (Excel)
	Project manage invoice (LT) (Excel)
	Project manage invoice (PL) (Excel)

DATA MODEL CONFIGURATION	FORMAT CONFIGURATIONS
	Project manage invoice (Word)
	Purchase advance invoice (Excel)
	Purchase advance invoice (Word)
	Sales advance invoice (Excel)
	Sales advance invoice (Word)
	Sales advance invoice (PL) (Excel)
	Sales invoice (Excel)
	Sales invoice (Excel) (BH)
	Sales invoice (Excel) (CZ)
	Sales invoice (Excel) (EE)
	Sales invoice (Excel) (FR)
	Sales invoice (Excel) (HU)
	Sales invoice (Excel) (IN)
	Sales invoice (Excel) (LT)
	Sales invoice (Excel) (LV)
	Sales invoice (Excel) (PL)
	Sales invoice (Excel) (TH)
	Sales invoice (Word)
	TMS Commercial Invoice (Excel)
	TMS Commercial Invoice (Word)
	Vendor invoice document (Excel)
	Vendor invoice document (CZ) (Excel)
	Vendor invoice document (HU) (Excel)
	Vendor invoice document (IN) (Excel)
	Vendor invoice document (LT) (Excel)

DATA MODEL CONFIGURATION	FORMAT CONFIGURATIONS
	Vendor invoice document (LV) (Excel)
	Vendor invoice document (MY) (Excel)
	Vendor invoice document (Word)
Order model	Agreement confirmation (Excel)
	Agreement confirmation (Word)
	Purchase agreement confirmation (Excel)
	Purchase agreement confirmation (Word)
	Purchase order (Excel)
	Purchase order (CZ) (Excel)
	Purchase order inquiry (CZ) (Excel)
	Purchase order (HU) (Excel)
	Purchase order inquiry (HU) (Excel)
	Purchase order (Word)
	Purchase order inquiry (Excel)
	Purchase order inquiry (Word)
	Sales order confirmation (Excel)
	Sales order confirmation (CZ) (Excel)
	Sales order confirmation (HU) (Excel)
	Sales order confirmation (Word)
Packing list model	Container contents (Excel)
	Container contents (Word)
	Load list (Excel)
	Load list (Word)
	Picking list (Excel)
	Picking list (CZ) (Excel)

DATA MODEL CONFIGURATION	FORMAT CONFIGURATIONS
	Picking list (Word)
	Production pick list (Excel)
	Production pick list (Word)
	Shipping pick list for load (Excel)
	Shipping pick list for load (Word)
	Shipping pick list for shipment (Excel)
	Shipping pick list for shipment (Word)
	Shipping pick list for wave (Excel)
	Shipping pick list for wave (Word)
Payment model	Customer payment advice (Excel)
	Customer payment advice (Word)
	Vendor payment advice (Excel)
	Vendor payment advice (Word)
Quotation model	Project quotation (Excel)
	Project quotation (Word)
	Request for quotation (Excel)
	Request for quotation (Accept) (Excel)
	Request for quotation (Accept) (Word)
	Request for quotation (Reject) (Excel)
	Request for quotation (Reject) (Word)
	Request for quotation (Return) (Excel)
	Request for quotation (Return) (Word)
	Request for quotation (Word)
	Sales quotation (Excel)
	Sales quotation (CZ) (Excel)

DATA MODEL CONFIGURATION	FORMAT CONFIGURATIONS
	Sales quotation (HU) (Excel)
	Sales quotation (Word)
	Sales quotation confirmation (Excel)
	Sales quotation confirmation (Word)
Reconciliation model	Cust account statement, Ext (Excel)
	Cust account statement, Ext (CN) (Excel)
	Cust account statement, Ext (Word)
	Cust account statement, France (Excel)
Reminder model	Collection letter note (Excel)
	Collection letter note (CN) (Excel)
	Collection letter note (Word)
	Customer interest note (Excel)
	Customer interest note (Word)
Waybill model	Load tender (Excel)
	Load tender (Word)
	Purchase order packing slip (Excel)
	Purchase order packing slip (CZ) (Excel)
	Purchase order packing slip (Word)
	Route (Excel)
	Route (Word)
	Sales order packing slip (Excel)
	Sales order packing slip (CZ) (Excel)
	Sales order packing slip (LT) (Excel)
	Sales order packing slip (PL) (Excel)
	Sales order packing slip (Word)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# New document user interface in Business document management

2/18/2021 • 2 minutes to read • [Edit Online](#)

Business document management lets business users edit business document templates by using a Microsoft 365 service or the appropriate Microsoft Office desktop application. Edits might include design changes or new deployments, or users might add placeholders to include additional data without having to change the source code. For more information about how to work with Business document management, see [Business document management overview](#).

The new document user interface (UI) is clearer and more comfortable to use. The **Business document** area shows only the templates that are available for the current provider.

The **New document** button lets users create and edit a template in an Electronic reporting (ER) format configuration that is provided by another provider. In the example in this topic, the provider is Microsoft.

## Make the new document UI in Business document management available

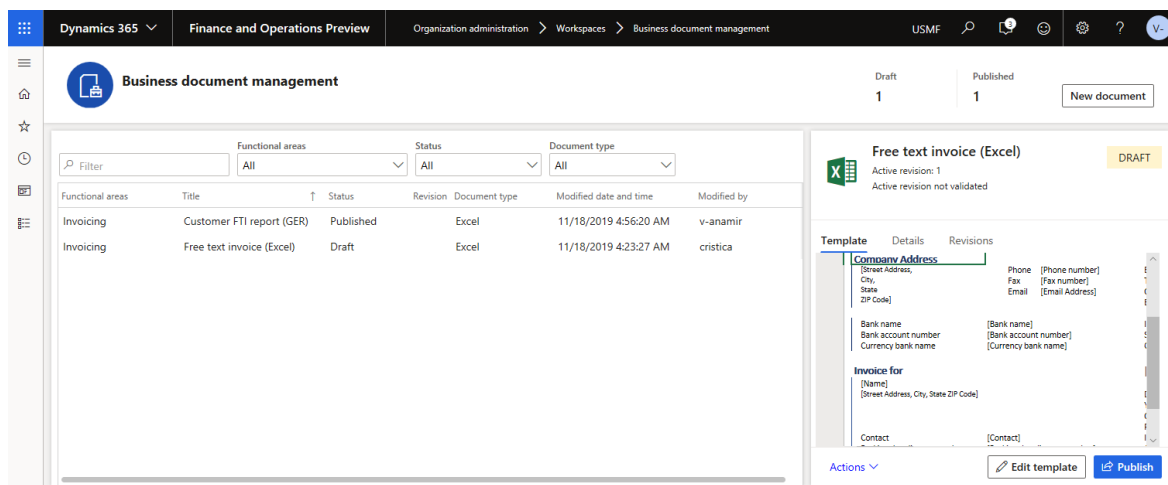
To start to use the new document UI in Business document management, you must turn on the **Office-like UI experience for Business document management** feature in the **Feature management** workspace.

Follow these steps to turn on this feature for all legal entities.

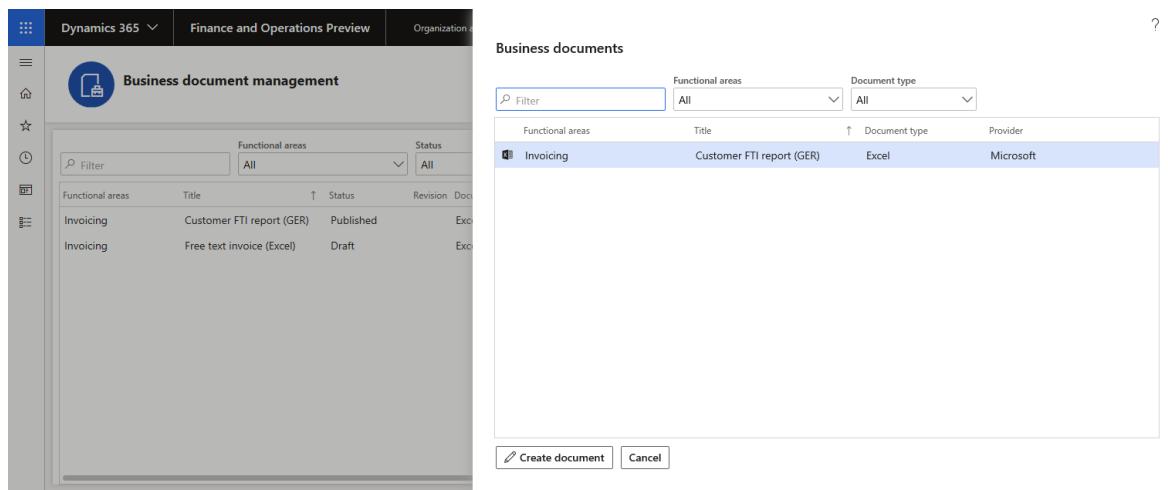
1. In the **Feature management** workspace, on the **New** tab, select the **Office-like UI experience for Business document management** feature in the list.
2. Select **Enable now** to turn on the selected feature.
3. Refresh the page to access the new feature.

### Edit templates that are owned by other providers

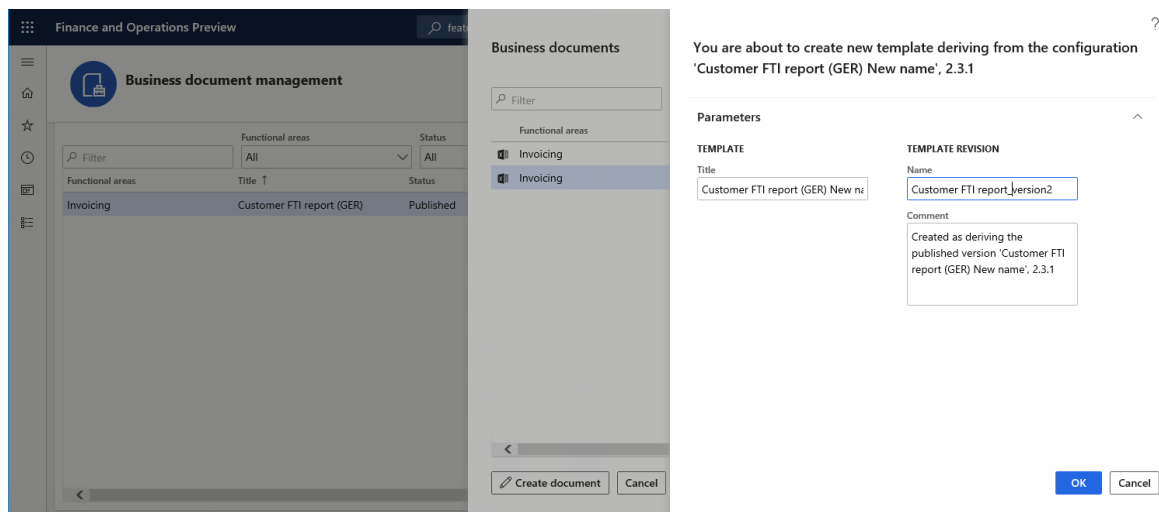
1. In the **Business document management** workspace, select **New document**.



2. In the dialog box, select the document to use as a template, and then select **Create document**.



3. In the new dialog box, in the **Title** field, change the title as you require. The title text is used to name the new ER format configuration that is automatically created. The draft version of this configuration (**Customer FTI report (GER) Copy**) will contain the edited template and will be used to run this ER format for the current user. The original template from the base ER format configuration will be used to run this ER format for every other user.
4. In the **Name** field, change the name of the first revision of the editable template that will be automatically created.
5. In the **Comment** field, update the remarks for the revision of the editable template that will be automatically created.
6. Select **OK** to confirm the start of the editing process.



The **New document** button is used to create and edit a template in an ER format configuration that is provided by another provider. In this example, the provider is Microsoft. When you select **New document**, you can view all the templates that are owned by current and other providers. After you select the template, it's opened for editing. The edited template will then be stored in a new ER format configuration that is automatically generated.

For more information, see [Business document management overview](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add new fields to a business document template in Microsoft Excel

2/18/2021 • 5 minutes to read • [Edit Online](#)

You can add new fields to a template that is used to generate business documents in Microsoft Excel format. These fields can be added as placeholders that are used to fill generated documents with required information from the application. For every field that you add, you can also specify a binding to the data sources, to specify what application data will be entered in the field when the template is used to generate business documents.

To learn more about this feature, complete the example in this topic. This example shows how to update a template to fill in the fields in free text invoice forms that are generated.

## Configure Business document management to edit templates

Because Business document management (BDM) is built on top of the [Electronic reporting \(ER\) overview](#) framework, you must configure the required ER and BDM parameters before you can start to work with BDM.

1. Sign in to the instance of Microsoft Dynamics 365 Finance as the system administrator.
2. Complete the following steps of the example in the [Business document management overview](#) topic:
  - a. Configure ER parameters.
  - b. Turn on BDM.

You can now start to use BDM to edit business document templates.

## Import ER solutions that contain a template

The example in this procedure uses the officially published ER solution. You must import the ER configurations of this solution into your current instance of Finance.

The **Free text invoice (Excel)** ER format configuration of this solution contains the business document template in Excel format that can be edited by using BDM. Import the latest version of this ER format configuration from Microsoft Dynamics Lifecycle Service (LCS). The corresponding ER data model and ER model mapping configurations will be imported automatically.

For more information about how to import ER configurations, see [Manage the ER configuration lifecycle](#).

Lifecycle Services

## Shared asset library

Select asset type

- Business database (0)
- Configuration (0)
- Cortana intelligence application (1)
- Data package (197)
- Database backup (0)
- Deployment (0)
- Downloadable VHD (229)
- Dynamics 365 Retail SDK (0)
- e-Commerce package (0)
- GER Configuration (1715)**
- Localized financial report (3)
- Localized financial report 2012 (4)
- Marketing asset (2)
- Model (24)

GER Configuration files

Asset Name	Count	Scope	Status	Created	Size
VAT Invoices (PL).version.32.22.42	1	Global	Published	11/5/2019	484 KB
Free text invoice (LT) (Excel).version.155.77.34	1	Global	Published	11/5/2019	700 KB
Cust account statement, Ext (Excel).version.24.10	1	Global	Published	11/1/2019	234 KB
Cust account statement, Ext (CN) (Excel).version.24.10.4	1	Global	Published	11/1/2019	237 KB
Reconciliation model mapping.version.24.8	1	Global	Published	11/1/2019	65 KB
Reconciliation model.version.24	1	Global	Published	11/1/2019	722 KB
Load tender model mapping.version.32.2	1	Global	Published	10/31/2019	32 KB
<b>Free text invoice (Excel).version.155.77</b>	<b>1</b>	<b>Global</b>	<b>Published</b>	<b>10/31/2019</b>	<b>869 KB</b>
Container contents model mapping.version.11.2	1	Global	Published	10/31/2019	32 KB
Container contents (Excel).version.11.2	1	Global	Published	10/31/2019	190 KB
Production pick list (Excel).version.11.5.2	1	Global	Published	10/31/2019	192 KB
Shipping pick list for load (Excel).version.11.5.3	1	Global	Published	10/31/2019	182 KB
Shipping pick list for shipment (Excel).version.11.5.3	1	Global	Published	10/31/2019	182 KB
Shipping pick list model mapping.version.11.3	1	Global	Published	10/31/2019	33 KB
Shipping pick list for wave (Excel).version.11.5.3	1	Global	Published	10/31/2019	182 KB

Additional details

**Free text invoice (Excel).**

Description

-

Asset ID  
027e4cd3-1d40-4aa0-9195-bc

Validation status  
Not validated

Created by  
System Account (Admin)

Created Date  
10/31/2019 3:08 PM

Modified by  
System Account (Admin)

Modified at  
10/31/2019 3:08 PM

## Edit the ER solution template

1. Sign in as a user who has access to the Business document management workspace.
2. Open the Business document management workspace.

Finance and Operations Preview

Search for a page

USMF

Business document management

Draft: 0, Published: 1

Filter: Functional areas: All, Status: All, Document type: All

Functional areas	Title	Status	Revision	Document type	Modified date and time
Invoicing	Free text invoice (Excel)	Published		Excel	11/6/2019 01:35:45 AM

Free text invoice (Excel) PUBLISHED

Revisions not yet created

Template Details

Free text invoice

[Your Company Name]

Company Address

[Street address] Phone [Phone number] B  
 City Fax [Fax number] T  
 State Email [Email Address] C  
 ZIP Code E

Bank name [Bank name] B  
 Bank account number [Bank account number] S  
 Currency bank name [Currency bank name] C

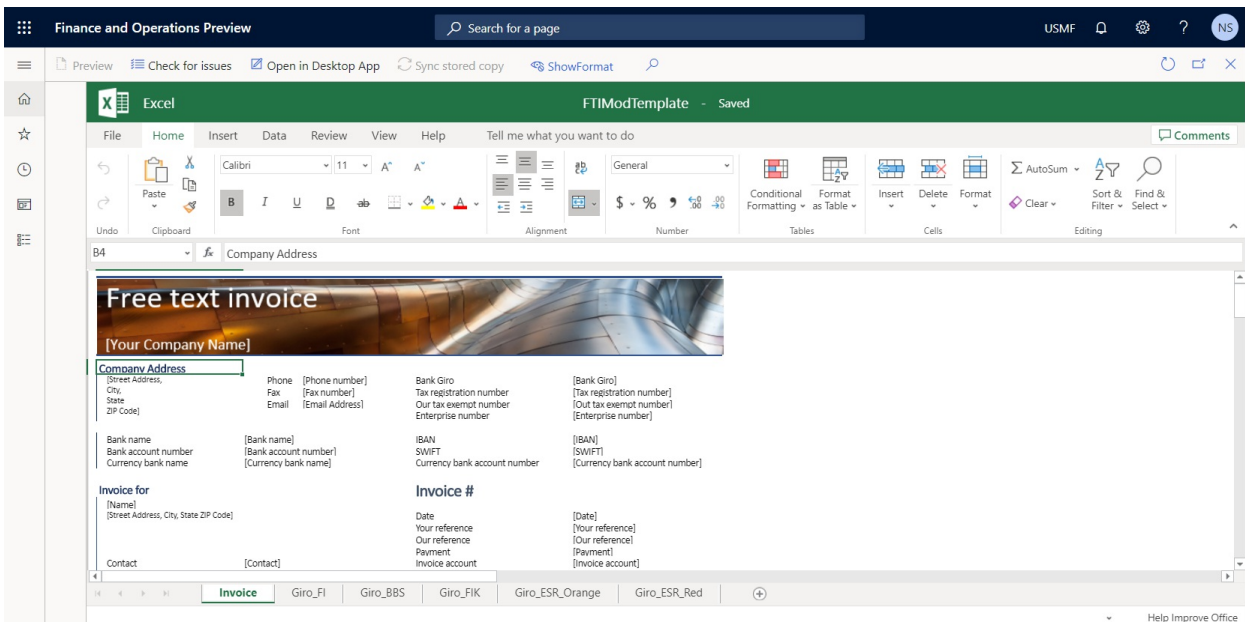
Invoice for

[Name] C  
 Street Address, City, State ZIP Code V

+ New template

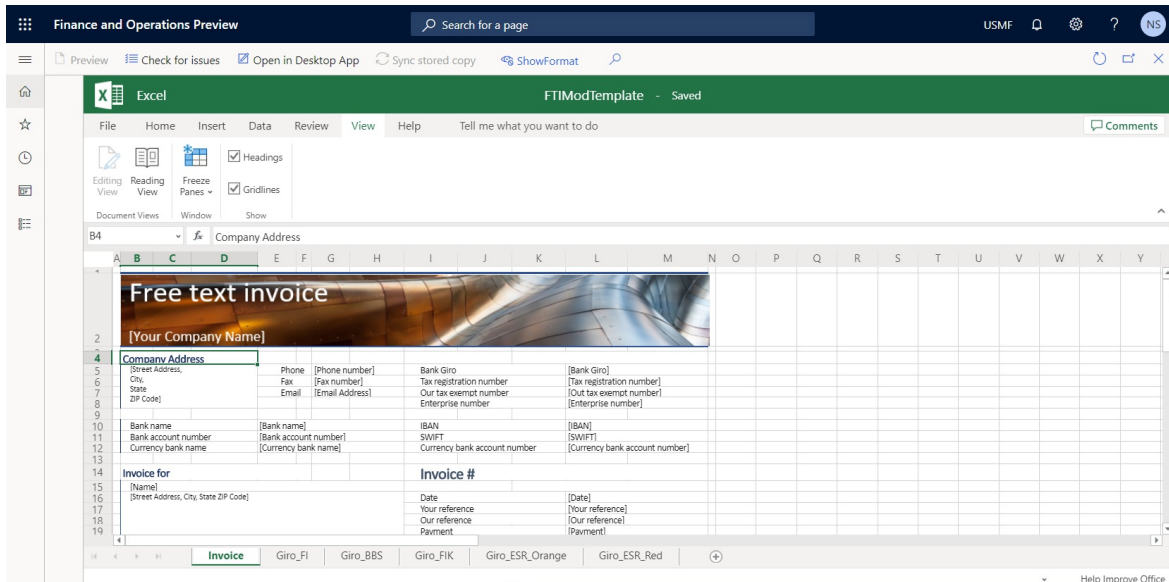
3. In the grid, select the Free text invoice (Excel) template.
4. In the right pane, select New template to create a new template that is based on the selected template.
5. In the Title field, enter Free text invoice (Excel) Contoso as the title of the new template.
6. Select OK to confirm the start of the editing process.

The BDM template editor page appears. You can use Microsoft 365 to edit the selected template online in the embedded control.

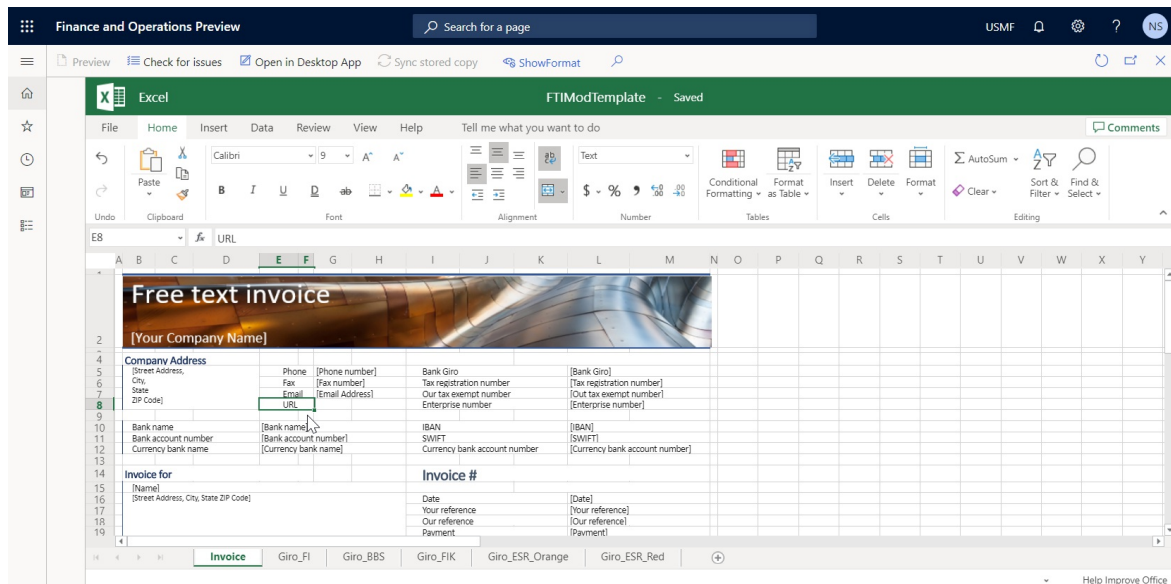


## Add the label for a new field to the template

1. On the BDM template editor page, on the Excel ribbon, on the **View** tab, select the **Headings** and **Gridlines** check boxes for the editable Excel template.

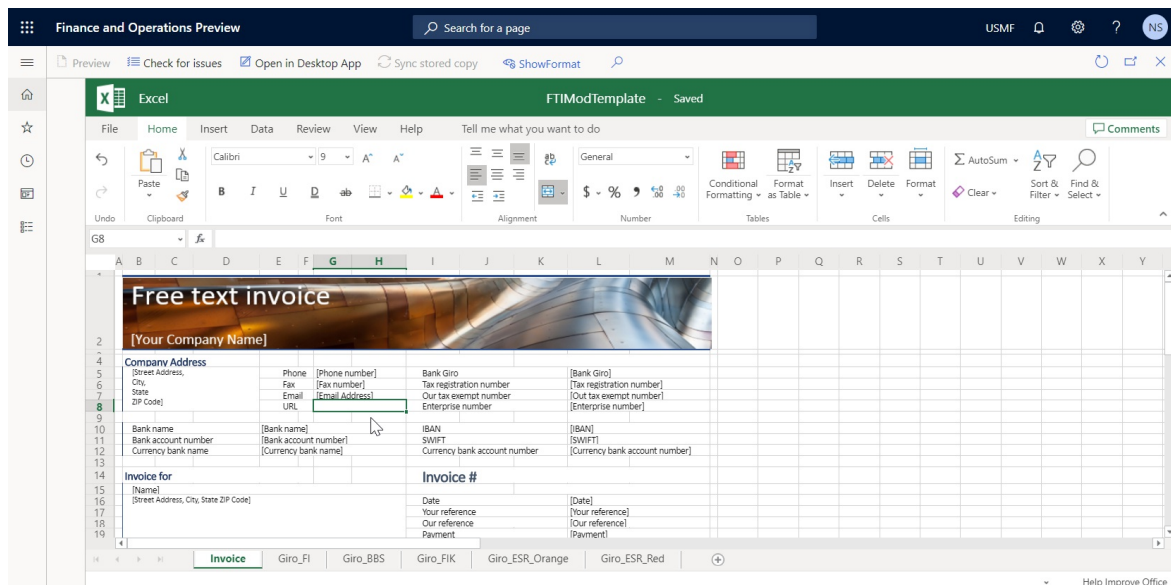


2. Select cells **E8:F8**.
3. On the Excel ribbon, on the **Home** tab, select **Merge & Center** to merge the selected cells into a new merged **E8:F8** cell.
4. In the merged cell **E8:F8**, enter **URL**.
5. Select merged cell **E7:F7**, select **Format painter**, and then select merged cell **E8:F8** to format it in the same way as merged cell **E7:F7**.



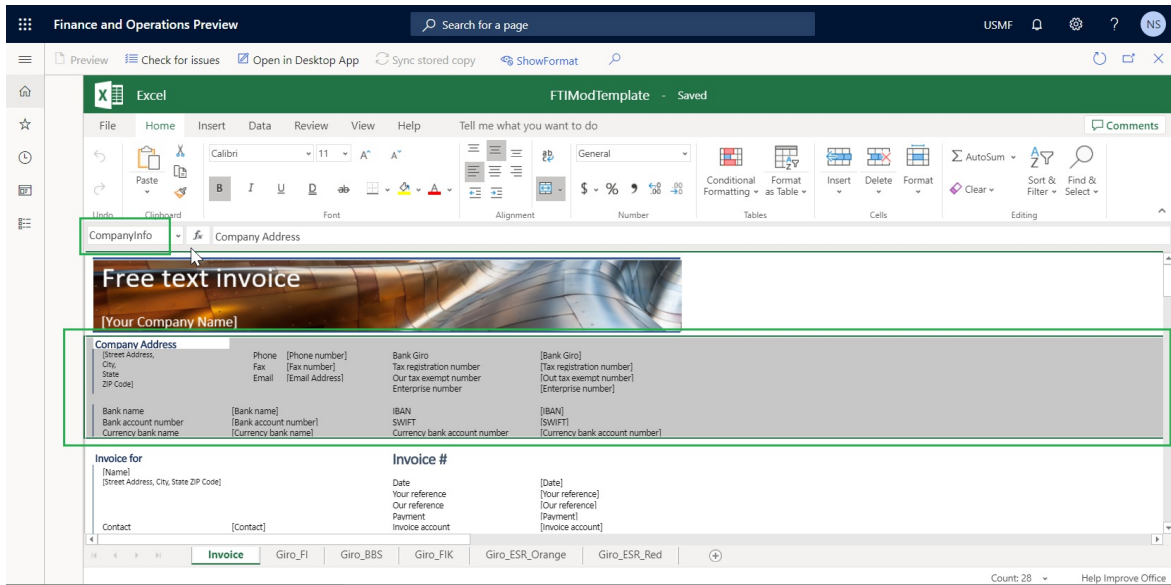
### Format the template to reserve space for a new field

1. On the BDM template editor page, select merged cell G8:H8.
2. On the Excel ribbon, on the Home tab, select Merge & Center to merge the selected cells into a new merged G8:H8 cell.
3. Select merged cell G7:H7, select Format painter, and then select merged cell G8:H8 to format it in the same way as merged cell G7:H7.



4. In the Name box field, select CompanyInfo.

The **CompanyInfo** range of the current Excel template holds all the fields that are used to fill the header of a generated report with the details of the current company as a seller party.

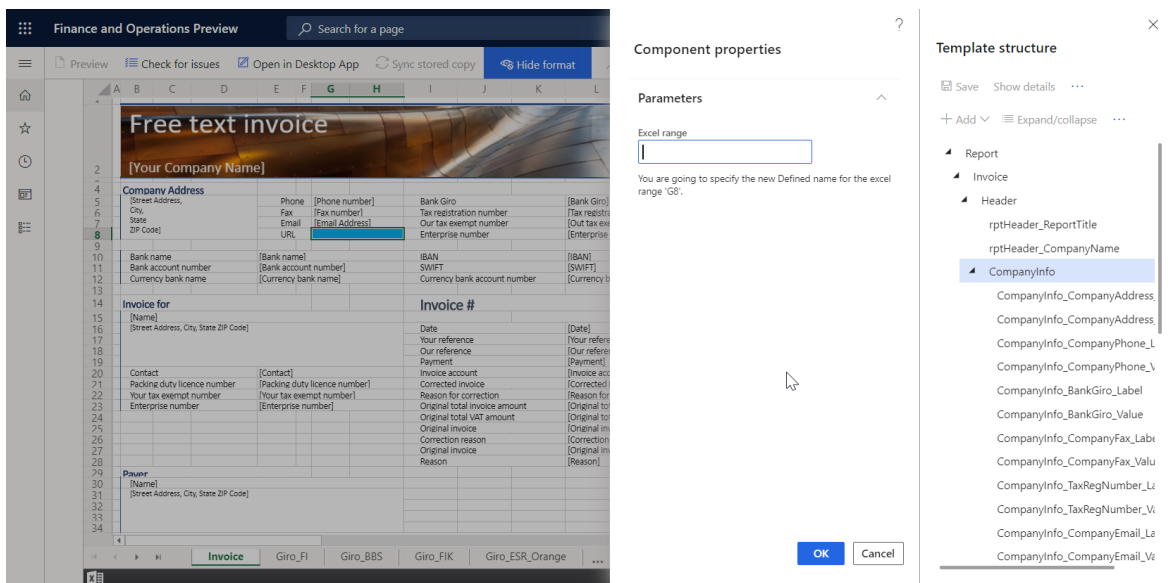


### Add a new field to the template

1. On the BDM template editor page, on the Action Pane, select **Show format**.
2. In the **Template structure** pane, select **Add**.

#### NOTE

You must adjust the section of the template that you want to use as a new field. You already made this adjustment by formatting merged cell G8:H8.



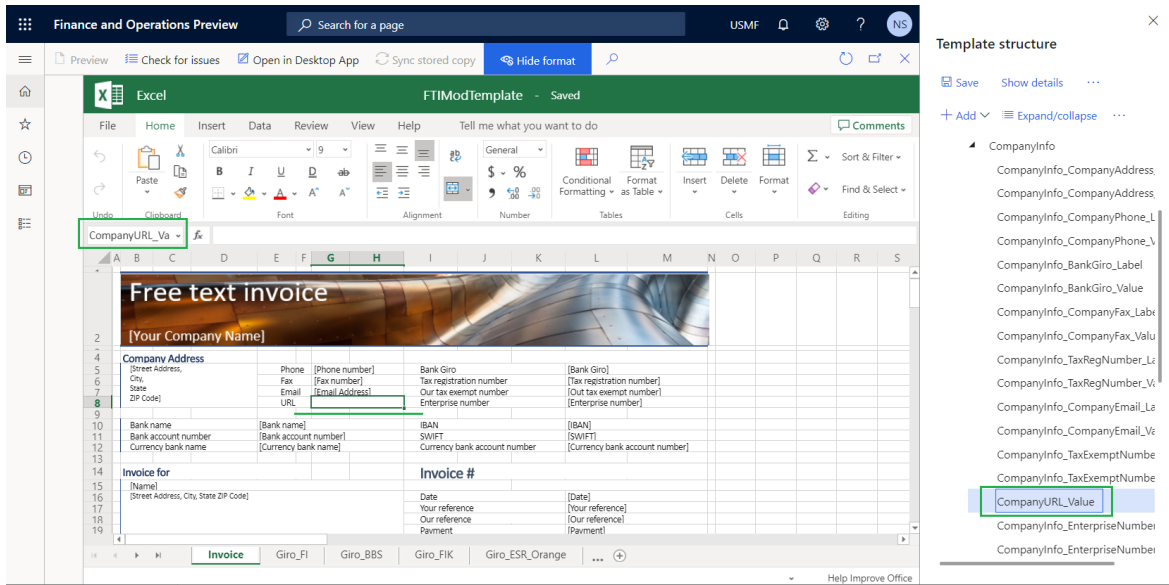
3. Select **Excel\Cell** to add a new field as a cell in the template.

You can select **Excel\Range** if you want to add a new range to the template. The range that is entered can contain multiple cells. You can add these cells later.

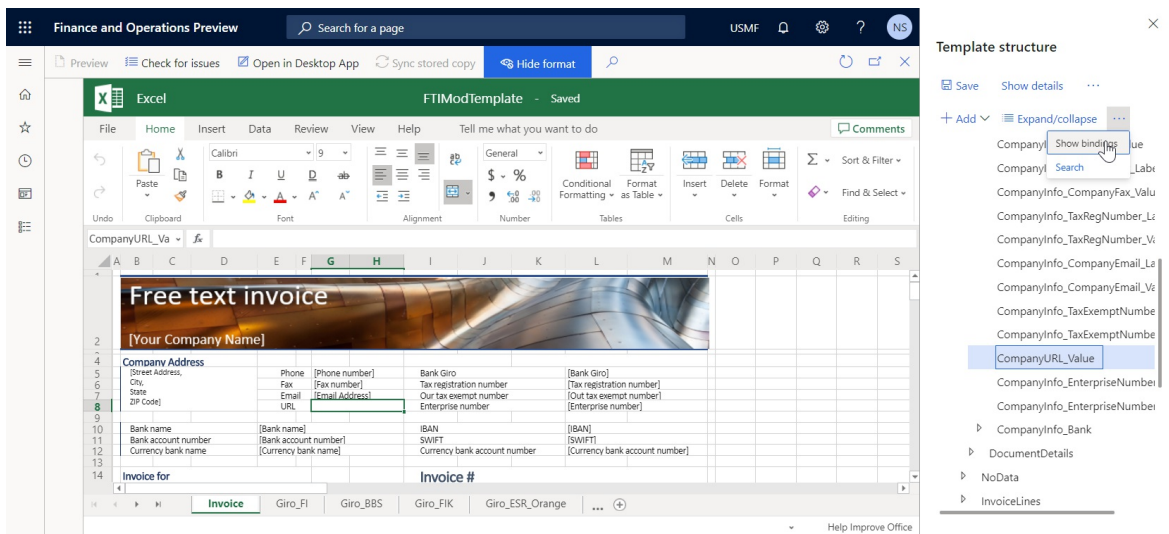
Notice that the **CompanyInfo** template component, is automatically selected in the **Template structure** pane, because it's the most suitable parent component in the current template structure for the field that you're adding.

4. In the **Excel range** field, enter **CompanyURL\_Value**.
5. Select **OK**.





6. In the Template structure pane, select the ellipsis button (...), and then select Show bindings.

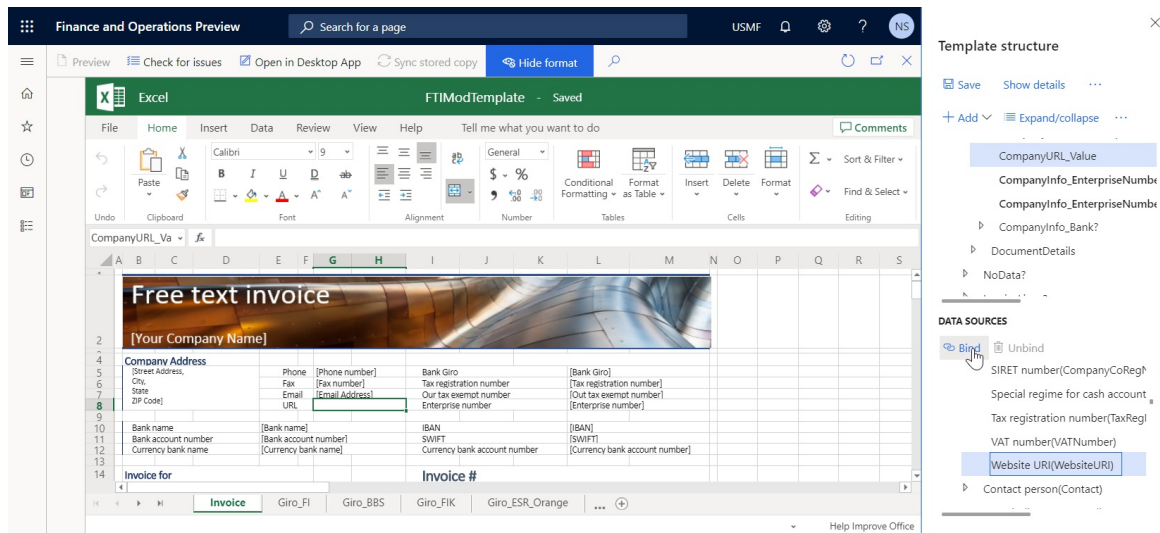


The Template structure pane now shows the data sources that are available in the underlying ER format.

7. Select CompanyInfo\_Value as the field that you plan to bind to a data source of the underlying ER format.

8. In the Data sources section of the Template structure pane, expand Model > InvoiceBase > CompanyInfo.

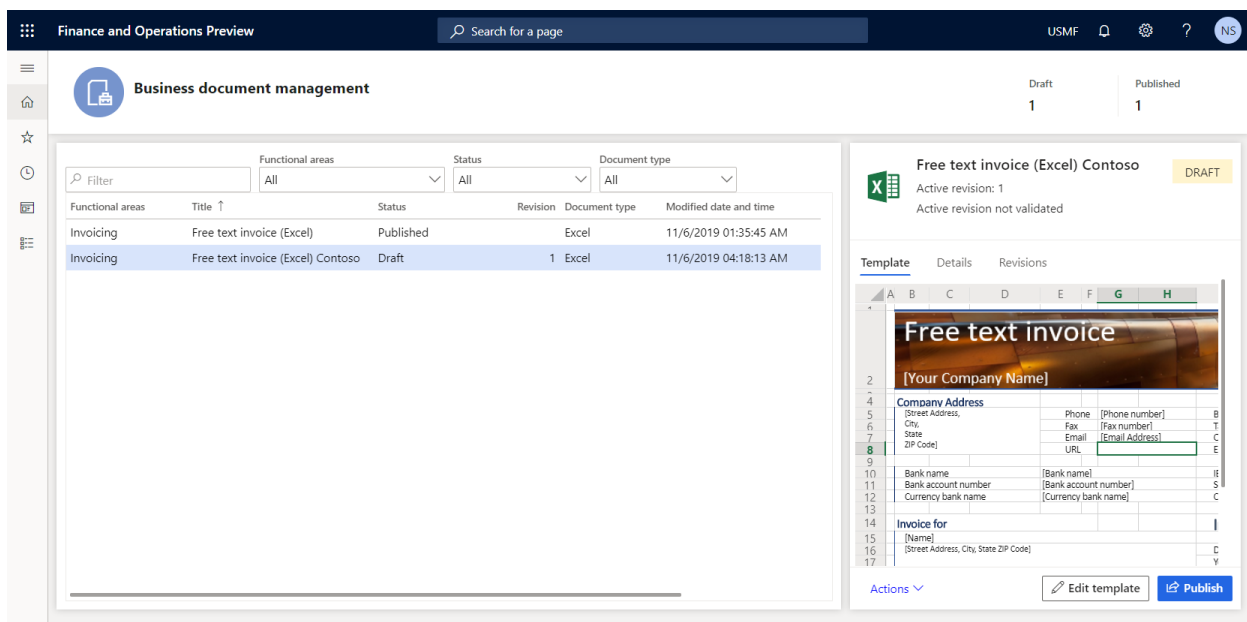
9. Under CompanyInfo, select the WebsiteURI item.



10. Select **Bind**.

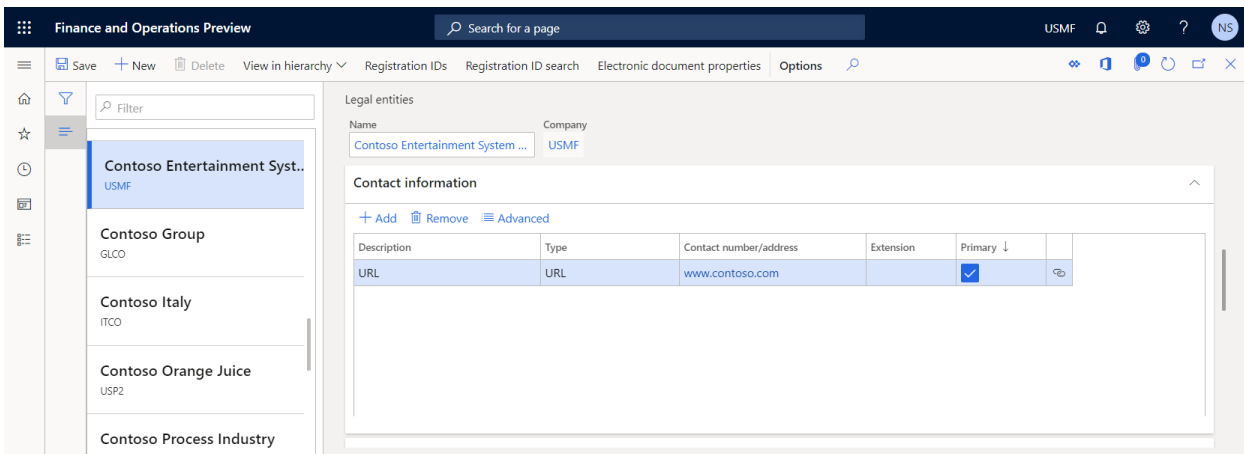
11. In the **Template structure** pane, select **Save**, and then close the BDM template editor page.

In the **Business document management** workspace, the **Template** tab in the right pane shows the updated template. In the grid, notice that the **Status** field for the edited template has been changed to **Draft**, and the **Revision** field is no longer blank. These changes indicate that the process of editing this template has been started.



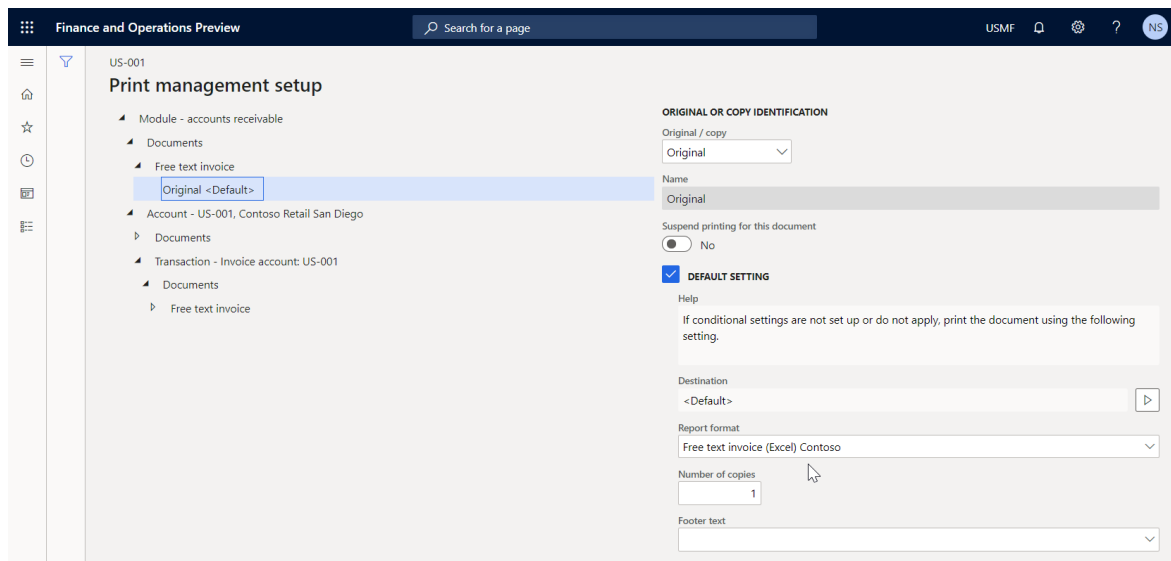
## Review company settings

1. Go to **Organization administration > Organizations > Legal entities**.
2. On the **Contact information** FastTab, verify that the company URL is entered.

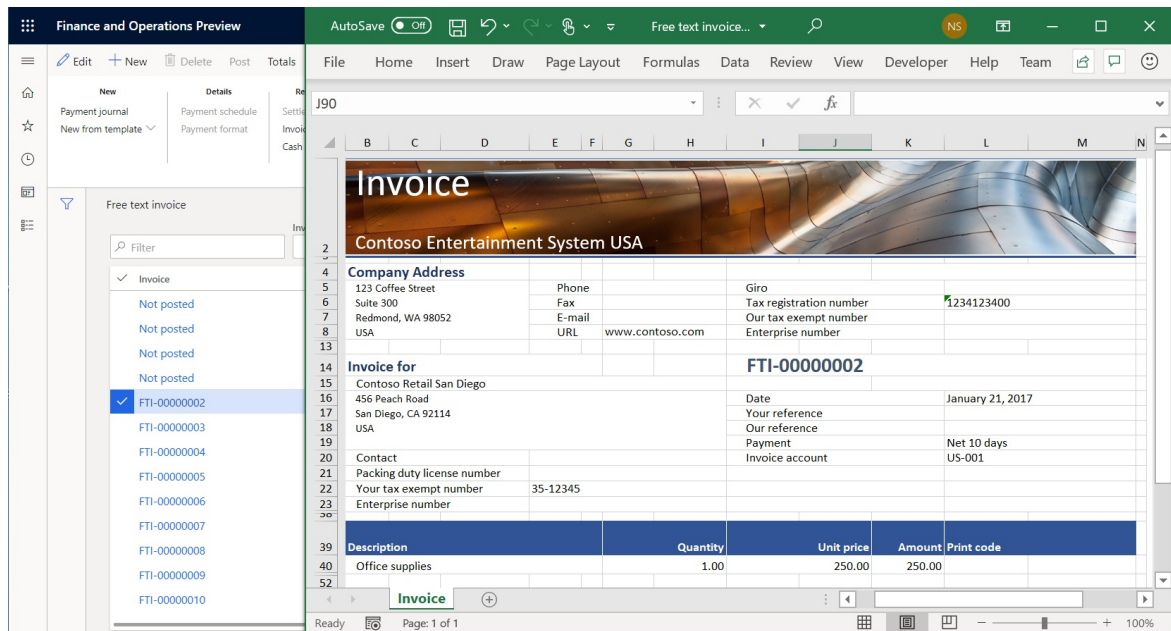


## Generate business documents to test the updated template

1. In the application, change the company to **USMF**, and go to **Accounts receivable > Invoices > All free text invoices**.
2. Select invoice **FTI-00000002**, and then select **Print management**.
3. In the left pane, expand **Module - accounts receivable > Documents > Free text invoice**.
4. Under **Free text invoice**, select the **Original** document level to specify the scope of invoices for processing.
5. In the right pane, in the **Report format** field, select the **Free text invoice (Excel) Contoso** template for the specified document level.



6. Press **Esc** to close the current page.
7. Select **Print > Selected**.
8. Download the generated document, and open it in Excel.



The modified template is used to generate the free text invoice report for the selected item. To analyze how this report is affected by changes that you make to the template, run the report in one application session immediately after you change the template in another application session.

## Related links

[Electronic reporting \(ER\) overview](#)

[Business document management overview](#)

[Design a configuration for generating reports in OPENXML format](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Extend the list of Electronic reporting (ER) functions

2/18/2021 • 2 minutes to read • [Edit Online](#)

Various types of functions are supported in Electronic reporting expressions for data transformation – text, date and time, mathematical logical, information, data type conversion, and other (business domain-specific functions). In addition to built-in functions, Electronic reporting lets you extend the list of available functions. This article includes an overview of key tasks that you must complete to introduce a new function.

All Electronic reporting functions in application code are represented as classes that extend the **ERExpression** class. Two types of functions are recognized:

- **Fixed number of arguments** – These functions are represented by classes that include methods that have the prefix **parm** (see **parmInput**, **parmStartNum** in the sample code the follows). The order of arguments is set by the **SysOperationDisplayOrderAttribute** attribute.
- **Variable number of arguments** – These functions (see the **ERExpressionGenericCase** class) are represented by classes that implement the **ERObjectContainer** interface. An additional **Add** method is used to declare the types that a function accepts.

Here are the recommended steps for introducing a new function for Electronic reporting expressions:

- Select a base class for your function, based on the return value type (see **ERExpressionString** in the sample code that follows).
  - Create a new class that extends the selected class (see **ERExpressionStringMid** in the sample code the follows).
    - Provide required attributes:
      - **SysOperationLabelAttribute** – This attribute defines the function’s name.
      - **SysOperationHelpTextAttribute** – This attribute defines the function’s Help text.
      - **ERComponentGroupAttribute** – This attribute defines the group that the function belongs to. (For more information, see [Formula designer in Electronic reporting \(ER\)](#).)
    - Provide arguments:
      - For a fixed number of arguments function, provide methods that have the prefix **parm**, and use the **SysOperationDisplayOrderAttribute** attribute to set the order of the arguments.
      - For a variable number of argument function, implement the **ERObjectContainer** interface.
  - Provide an evaluation method.

Here is an example.

```

/// <summary>
/// Returns the characters from the middle of a text string, given a starting position and length.
/// </summary>
[
    SysOperationLabelAttribute ('MID'),
    SysOperationHelpTextAttribute ("@ElectronicReporting:ExpressionStringMidHelpText"),
    ERComponentGroupAttribute ("@ElectronicReporting:String")
]
class ERExpressionStringMid extends ERExpressionString
{
    ERExpressionString input;
    ERExpressionInt startNum;
    ERExpressionInt numChars;
    public str evaluateString(ERIDataContext _dataContext)
    {
        return subStr(
            this.parmInput().evaluateString(_dataContext),
            this.parmStartNum().evaluateInt(_dataContext),
            this.parmNumChars().evaluateInt(_dataContext));
    }
    [DataMemberAttribute, SysOperationLabelAttribute ("@ElectronicReporting:Input"),
    SysOperationDisplayOrderAttribute ("1")]
    public ERExpressionString parmInput(ERExpressionString _input = input)
    {
        input = _input;
        return input;
    }
    [DataMemberAttribute, SysOperationLabelAttribute ("@ElectronicReporting:NumChars"),
    SysOperationDisplayOrderAttribute ("3")]
    public ERExpressionInt parmNumChars(ERExpressionInt _numChars = numChars)
    {
        numChars = _numChars;
        return numChars;
    }
    [DataMemberAttribute, SysOperationLabelAttribute ("@ElectronicReporting:StartNum"),
    SysOperationDisplayOrderAttribute ("2")]
    public ERExpressionInt parmStartNum(ERExpressionInt _startNum = startNum)
    {
        startNum = _startNum;
        return startNum;
    }
    public str toString()
    {
        return ERExpressionStringPresenter::namedFunctionToStr(this);
    }
}

```

## Suggested guidance

The following guidance is intended to help you design your custom Electronic reporting functions:

- Reuse the names of Microsoft Excel functions whenever you can, so that Electronic reporting formulas remain Excel-like. In this way, you will keep Electronic reporting formulas intelligible for end users.
- Electronic reporting doesn't support list types for primitive data types. Therefore, we have decided to use a data container list that has a single **Value** item in it.
- Release a new function's list extension as a new application hotfix. Electronic reporting designers will refer to the hotfix number in Electronic reporting configurations that use that new custom function. Whenever a configuration of this type is imported into a new instance, Electronic reporting will evaluate whether the required hotfix has been installed, to maintain compliance between the Electronic reporting configuration and the version that configuration is imported into.

# Additional resources

[Electronic reporting \(ER\) overview](#)

[Formula designer in Electronic reporting \(ER\)](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# ER framework API changes for Application update 7.3

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how the API of the Electronic reporting (ER) framework has been changed in the Dynamics 365 for Finance and Operations, Enterprise edition Application update 7.3.

There are two types of changes to the ER APIs:

- Several X++ classes were moved from X++ to an external assembly.
- The rest of X++ classes were marked as internal.

## How to access classes that were moved from X++ to an external assembly

To access external classes, you need to add the **using** directive to the beginning of your file.

```
using Microsoft.Dynamics365.LocalizationFramework;
```

You can then access an external class without any additional changes, for example.

```
var destination = new ERFileDestinationMemory();
```

You can also create an alias for your namespace.

```
using LF = Microsoft.Dynamics365.LocalizationFramework;
```

You can then refer to an external class by using the namespace alias that you created.

```
var destination = new LF.ERFileDestinationMemory();
```

## How to access internal X++ objects by using ERObjectsFactory

In Application update 7.3 and later updates, the calling code must access the ER objects by using the methods of the **ERObjectsFactory** class. Several examples of these changes are shown.

### Code to display a format mapping lookup

Before Application update 7.3

```
// pattern
ERFormatMappingTableLookup::lookupFormatMapping(<form control>, <model name>[, <data container name>]);
// sample code
ERFormatMappingTableLookup::lookupFormatMapping(_referenceGroupControl, bankLCMiscChargeReportERModelName);
```

Application update 7.3 and later



```
// pattern
ERObjectsFactory::createFormatMappingTableLookupForControlAndModel(<form control>, <model name>[, <data
container name>]).performFormLookup();
// sample code
ERObjectsFactory::createFormatMappingTableLookupForControlAndModel(_referenceGroupControl,
bankLCMiscChargeReportERModelName).performFormLookup();
```

## Code to run a format mapping for data export

Before Application update 7.3

```
// pattern
ERFormatMappingRun::constructByFormatMappingId(<format mapping id>, <file name>, <show prompt
dialog>).run();
// sample code
ERFormatMappingRun::constructByFormatMappingId(erBinding, '', true).run();
```

Application update 7.3 and later

```
// pattern
ERObjectsFactory::createFormatMappingRunByFormatMappingId(<format mapping id>, <file name>, <show prompt
dialog>).run();
// sample code
ERObjectsFactory::createFormatMappingRunByFormatMappingId(erBinding, '', true).run();
```

## Code to run a format mapping for data import

Before Application update 7.3

```
// pattern
ERModelMappingDestinationRun::constructByImportFormatMappingId(<mapping id>, <integration point>).run();
// sample code
ERModelMappingDestinationRun::constructByImportFormatMappingId(custPaymModeTable.ERModelMappingTable,
CustVendOutPaymConstants::IntegrationPoint).run();
```

Application update 7.3 and later

```
// pattern
ERObjectsFactory::createMappingDestinationRunByImportFormatMappingId(<mapping id>, <integration
point>).run();
// sample code
ERObjectsFactory::createMappingDestinationRunByImportFormatMappingId(custPaymModeTable.ERModelMappingTable,
CustVendOutPaymConstants::IntegrationPoint).run();
```

## Code to create a browser file destination

Before Application update 7.3

```
// sample code
new ERFileDestinationBrowser();
```

Application update 7.3 and later

```
// sample code
ERObjectsFactory::createFileDestinationBrowser();
```

## Code to create an attachment file destination

## Before Application update 7.3

```
// pattern
ERFileDestinationAttachment::construct(<record>, ERDocuManagement::instance().otherDocuType());
// sample code
ERFileDestinationAttachment::construct(_cashRegisterFiscalTrans_W,
ERDocuManagement::instance().otherDocuType());
```

## Application update 7.3 and later

```
// pattern
ERObjectsFactory::createFileDestinationAttachmentWithOtherDocuType(<record>);
// sample code
ERObjectsFactory::createFileDestinationAttachmentWithOtherDocuType(_cashRegisterFiscalTrans_W);
```

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Specify a custom storage location for generated documents

2/18/2021 • 4 minutes to read • [Edit Online](#)

The application programming interface (API) of the Electronic reporting (ER) framework lets you extend the list of storage locations for documents that ER formats generate. This topic includes an overview of the main tasks that you must complete to add a custom storage location.

## Prerequisites

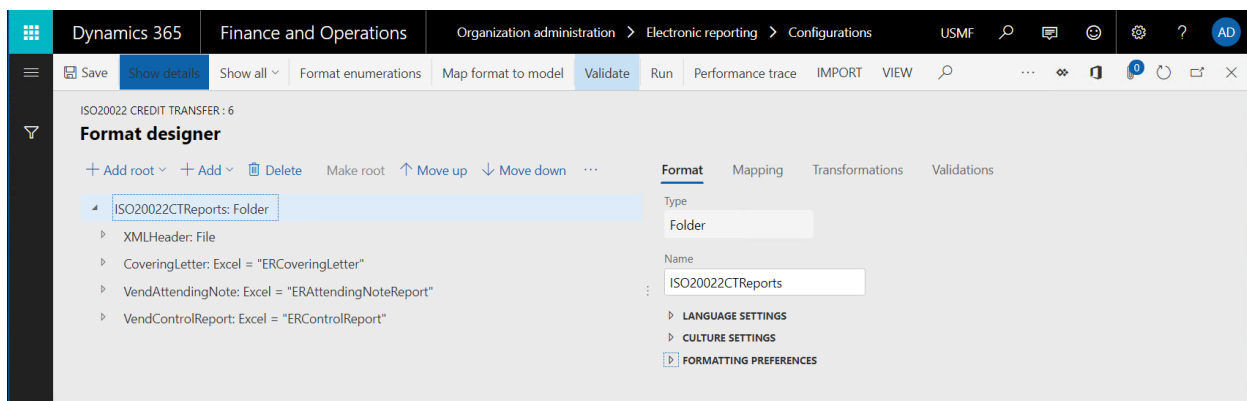
You must deploy a topology that supports continuous build. (For more information, see [Deploy topologies that support continuous build and test automation](#).) You must have access to this topology for one of the following roles:

- Electronic reporting developer
- Electronic reporting functional consultant
- System administrator

You must also have access to the development environment for this topology.

## Create or import an ER format configuration

In the current topology, [create a new ER format](#) to generate documents that you plan to add a custom storage location for. Alternatively, [import an existing ER format into this topology](#).



### IMPORTANT

The ER format that you create or import must contain at least one of the following format elements:

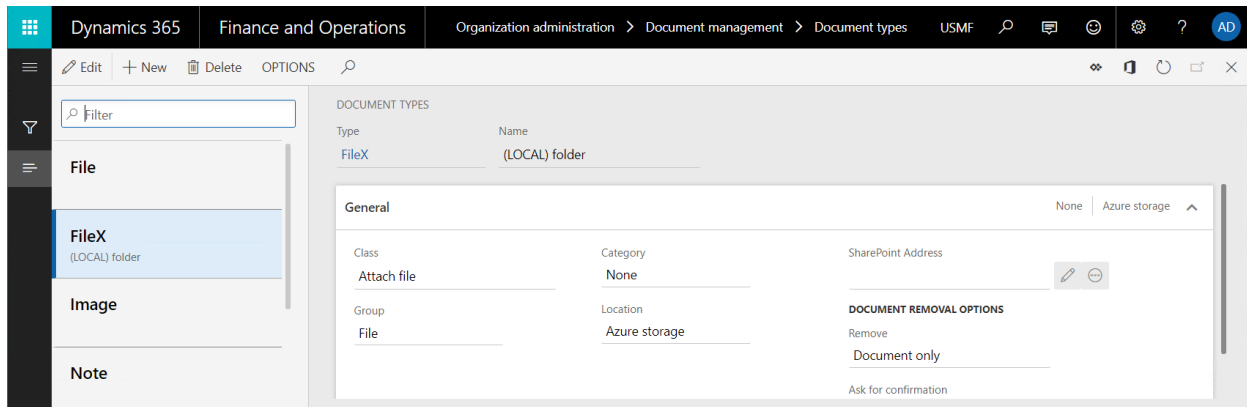
- File
- Folder
- Merger
- Attachment

## Create a new document type

To specify how documents that an ER format generates are routed, you must configure [Electronic reporting \(ER\) destinations](#). In each ER destination that is configured to store generated documents as files, you must specify a

document type of the Document management framework. Different document types can be used to route documents that different ER formats generate.

1. Add a new **document type** for the ER format that you created or imported earlier. In the illustration that follows, the document type is **FileX**.
2. To differentiate this document type from other document types, include a specific keyword in its name. For example, in the illustration that follows, the name is **(LOCAL) folder**.
3. In the **Class** field, specify **Attach file**.
4. In the **Group** field, specify **File**.



#### NOTE

Document types are company-specific. To use an ER format with a configured destination in multiple companies, you must configure a separate document type in each company.

## Review source code

Review the code of the `insertFile()` method of the `ERDocuManagement` class. Notice that the `AttachingFile()` event is raised while the generated file is attached to a record.

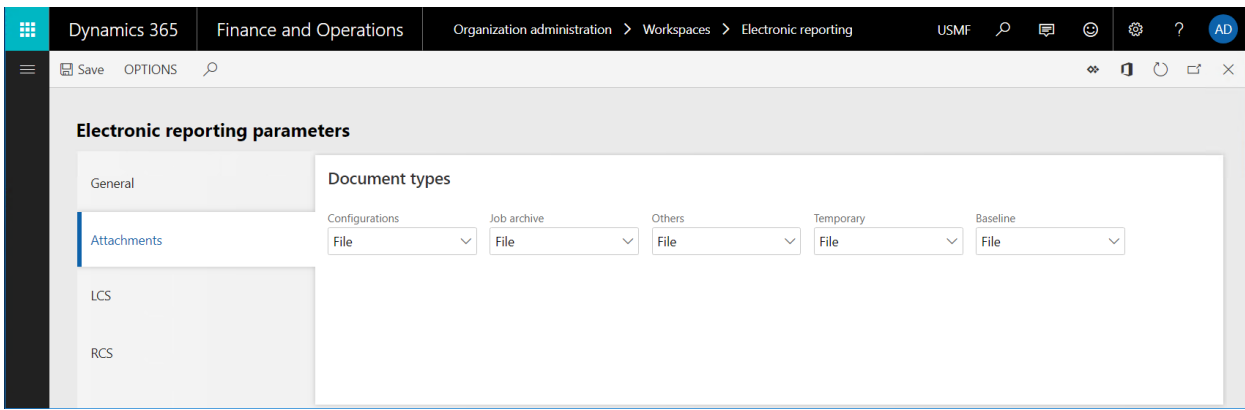
```

/// <summary>
/// Inserts file as attachment in Document Management.
/// </summary>
/// <param name = "_owner">A record as the attachment owner.</param>
/// <param name = "_stream">The file stream.</param>
/// <param name = "_filePath">The file path with name.</param>
/// <param name = "_attachmentName">The name of file attachment.</param>
/// <returns>The reference to inserted file.</returns>
[Hookable(false)]
public DocuRef insertFile(
    Common _owner,
    System.IO.Stream _stream,
    str _filePath,
    str _attachmentName,
    DocuTypeId _docuTypeId)
{
    DocuRef docuRef;
    if (_stream)
    {
        DocuType::createDefaults();
        if (!this.isDocuTypeValid(_docuTypeId))
        {
            throw error(strFmt("@ElectronicReporting:DocuTypeIsValid", _docuTypeId));
        }
        var args = ERDocuManagementAttachingFileEventArgs::construct(_owner, _stream, _filePath,
            _attachmentName, _docuTypeId);
        ERDocuManagementEvents::onAttachingFile(args);
        if (args.isHandled())
        {
            docuRef = args.getDocuRef();
        }
        else
        {
            docuRef = this.attachFile(_owner, _stream, _filePath, _attachmentName, _docuTypeId);
        }
    }
    return docuRef;
}

```

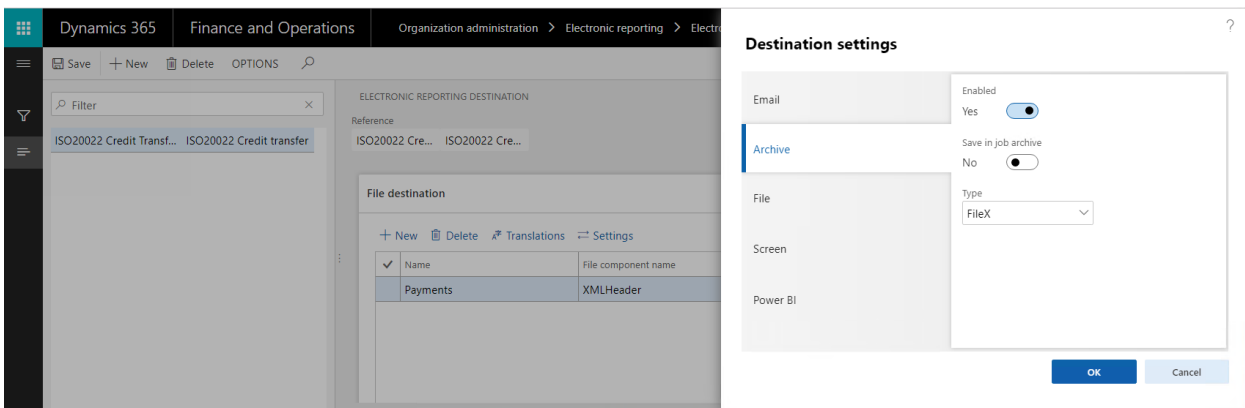
The **AttachingFile()** event is raised when the following ER destinations are processed:

- **Archive** – When this destination is used, a new record for the ER format that is run is created in the ERFormatMappingRunJobTable table. The **Archived** field in this record is set to **False**. If the ER format is successfully run, the generated document is attached to this record, and the **AttachingFile()** event is raised. The document type that is selected in this ER destination determines the storage location for the attached file (Microsoft Azure Storage or a Microsoft SharePoint folder).
- **Job archive** – When this destination is used, a new record for the ER form that is run is created in the ERFormatMappingRunJobTable table. The **Archived** field in this record is set to **True**. If the ER format is successfully run, the generated document is attached to this record, and the **AttachingFile()** event is raised. The document type that is configured in the ER parameters determines the storage location for the attached file (Azure Storage or a SharePoint folder).



## Configure an ER destination

1. Configure the archived destination for one of the previously mentioned elements (file, folder, merger, or attachment) of the ER format that you created or imported. For guidance, see [ER Configure destinations](#).
2. Use the document type that you added earlier for the configured destination. (For the example in this topic, the document type is FileX.)



## Modify source code

1. Add a new class to your Microsoft Visual Studio project, and write code to subscribe to the **AttachingFile()** event that was mentioned earlier. (For more information about the extensibility pattern that is used, see [Respond by using EventHandlerResult](#).) For example, in the new class, write code that performs the following actions:
  - a. Store generated files in a folder of the local file system of the server that runs the Application Object Server (AOS) service.
  - b. Store these generated files only when the new document type (for example, the **FileX** type that has the "(LOCAL)" keyword in its name) is used while a file is attached to the record in the ER execution job log.

```

class ERDocuSubscriptionSample
{
    void new()
    {
    }
    [SubscribesTo(classStr(ERDocuManagementEvents),
    staticDelegateStr(ERDocuManagementEvents,
    attachingFile))]
    public static void ERDocuManagementEvents_attachingFile(ERDocuManagementAttachingFileEventArgs
    _args)
    {
        if (!_args.isHandled())
        {
            DocuType docuType = DocuType::find(_args.getDocuTypeId());
            if (strContains(docuType.Name, '(LOCAL)'))
            {
                _args.markAsHandled();
                var stream = _args.getStream();
                if (stream.CanSeek)
                {
                    stream.Seek(0, System.IO.SeekOrigin::Begin);
                }
                using (var localStream = System.IO.File::OpenWrite(@"c:\0\" +
                _args.getAttachmentName()))
                {
                    stream.CopyTo(localStream);
                }
            }
        }
    }
}

```

2. Rebuild your project.

## Run the ER format that you created or imported

1. Execute the ER format that you created or imported.
2. Go to **Organization administration > Electronic reporting > Electronic reporting jobs**. Find the record that was created for this execution job, and that has the generated file attached to it.
3. Explore the local C:\0 folder to find same generated file.

## Additional resources

- [Electronic reporting \(ER\) destinations](#)
- [Extensibility home page](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Mobile app home page

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes the **Finance and Operations (Dynamics 365)** mobile app and provides links to resources that can help you implement it in your organization.

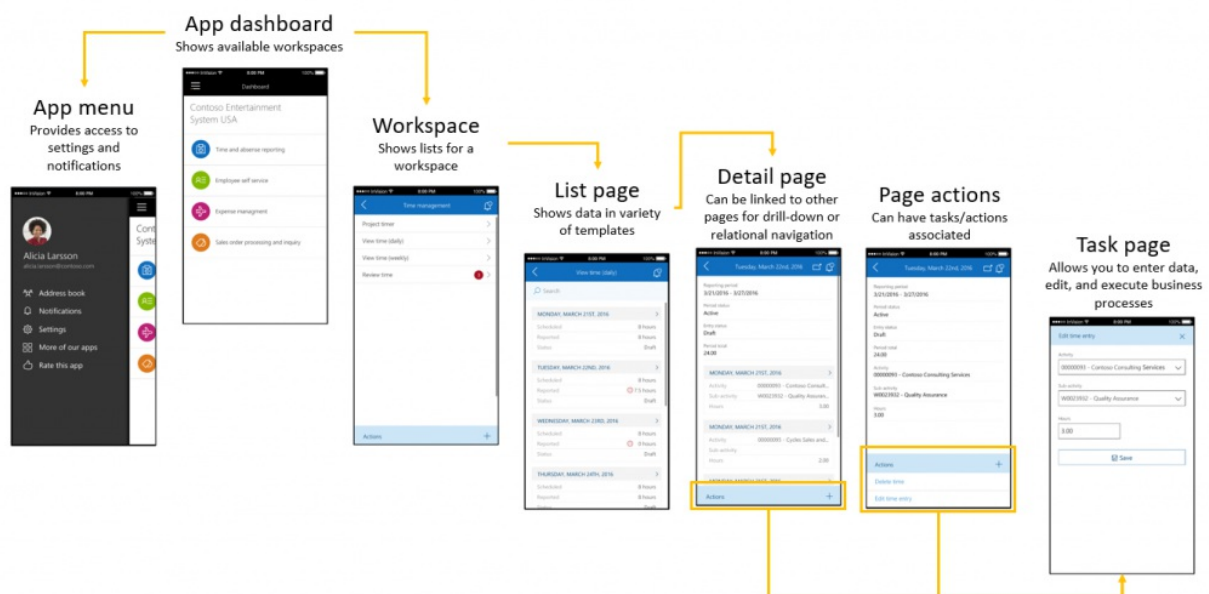
## Overview

The mobile app enables your organization to make its business processes available on mobile devices. After your IT admin enables the mobile workspaces for your organization, users can sign in to the app and immediately begin to run business processes from their mobile devices. The mobile app includes the following features that can help increase productivity:

- Users can view, edit, and act on business data, even if they have intermittent network connectivity or their mobile devices are completely offline. When a device reestablishes a network connection, offline data operations are automatically synchronized.
- IT admins or developers can build and publish mobile workspaces that have been tailored to their organization. The app uses your existing code assets. Therefore, you don't have to re-implement your validation procedures, business logic, or security configuration.
- IT admins or developers can easily design mobile workspaces by using the point-and-click workspace designer that is included with the web client.
- IT admins or developers can optionally optimize the offline capabilities of workspaces by using the Business logic extensibility framework. Because data continues to be processed while a device is offline, your mobile scenarios remain rich and fluid, even if devices don't have constant network connectivity.

## Elements of the mobile app

Navigation in the mobile app consists of four basic concepts: the dashboard, workspaces, pages, and actions.



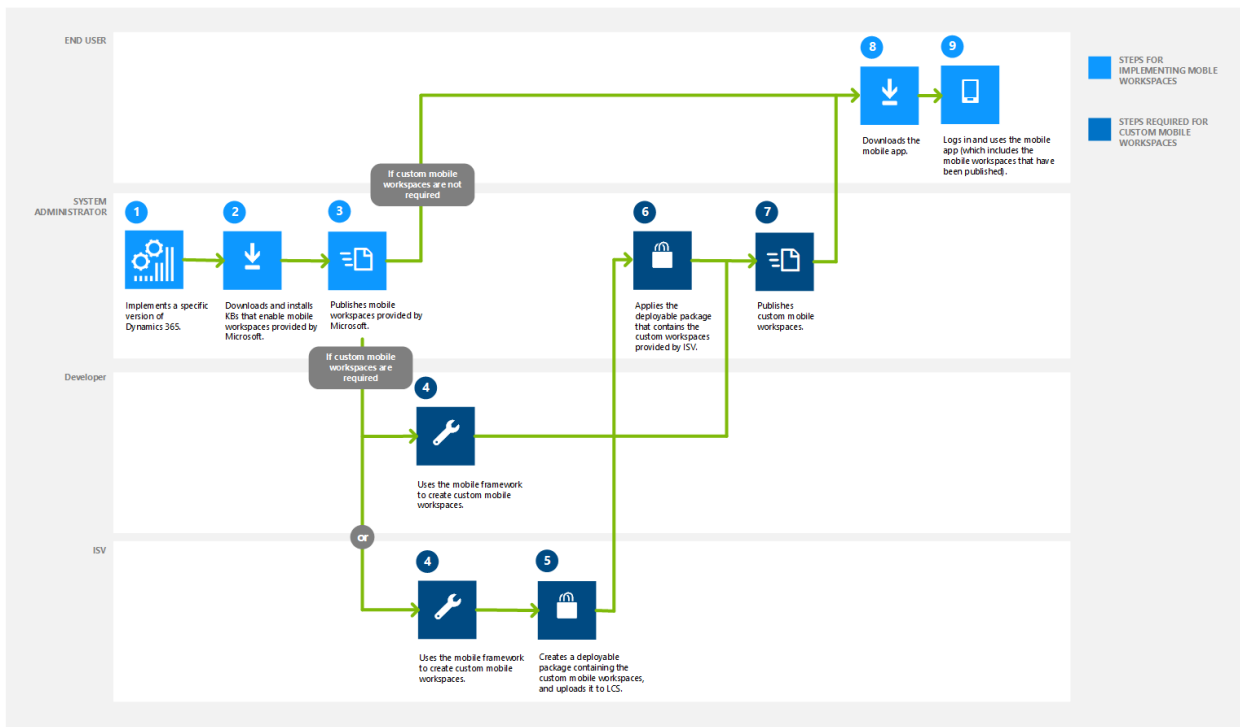
1. When you start the app, you go to the **dashboard**.
2. On the dashboard, you can see a list of **workspaces** that have been published.



3. In each workspace, you can see a list of **pages** that are available for that workspace.
4. After you're on a page, you can perform several actions. Here are some examples:
  - View detailed data.
  - Navigate to other pages for related data, such as entity details or lines.
  - See a list of actions that are available for that page. Actions let you create or edit existing data.

## Implementation process

The following illustration shows the process for implementing both mobile workspaces that are provided by Microsoft and custom mobile workspaces.



The following table includes links to resources that can help you implement both mobile workspaces that are provided by Microsoft and custom mobile workspaces. The numbers in the first column correspond to the numbered steps in the previous illustration.

STEP	ROLE	ACTION	RESOURCES TO HELP YOU COMPLETE THE ACTION
1	System administrator	Implement the Finance and Operations app in your organization.	<ul style="list-style-type: none"> <li>• If you haven't yet deployed a version of Microsoft Dynamics 365, see <a href="#">Deploy a demo environment</a>.</li> <li>• To see a list of mobile workspaces that can be used, see <a href="#">Mobile workspaces recently released</a>.</li> </ul>

STEP	ROLE	ACTION	RESOURCES TO HELP YOU COMPLETE THE ACTION
2	System administrator	<b>If you're using Microsoft Dynamics 365 for Operations version 1611:</b> Download and install KBs that enable the mobile workspaces that are provided by Microsoft.	See the following topics for more information: <ul style="list-style-type: none"> <li>• <a href="#">Cost controlling mobile workspaces</a></li> <li>• <a href="#">Inventory on-hand mobile workspace</a></li> <li>• <a href="#">Sales orders mobile workspaces</a></li> <li>• <a href="#">Vendor collaboration mobile workspace</a></li> <li>• <a href="#">Project time entry mobile workspace</a></li> <li>• <a href="#">Expense management mobile workspace</a></li> </ul>
3	System administrator	Publish the mobile workspaces that are provided by Microsoft.	<a href="#">Publish a mobile workspace</a>
4	Developer or independent software vendor (ISV)	Use the mobile platform to create custom mobile workspaces.	<a href="#">Mobile platform</a>
5	ISV	Create a deployable package that contains custom mobile workspaces, and upload the package to Microsoft Dynamics Lifecycle Services (LCS).	<a href="#">Create a deployable package</a>
6	System administrator	Apply the deployable package that contains the custom workspaces that are provided by the independent software vendor (ISV).	<a href="#">Apply a deployable package</a>
7	System administrator	Publish the custom mobile workspaces that are provided by the ISV.	<a href="#">Publish a mobile workspace</a>
8	User	Download and install the mobile app.	<a href="#">Finance and Operations app for Android</a> <a href="#">Finance and Operations app for iOS</a> (Windows Phone unsupported)
9	User	Sign in, and use the mobile app. The app includes the mobile workspaces that have been published by the system administrator.	To see a list of mobile workspaces that are provided by Microsoft, see <a href="#">Mobile workspaces recently released</a> .

# Troubleshooting

## Mobile platform resources

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Available mobile workspaces

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic lists the mobile workspaces that are available for use with the Dynamics 365 for Unified Operations mobile app.

MOBILE WORKSPACE	DESCRIPTION	AVAILABILITY
<a href="#">Company directory</a>	Allows you to view and contact other employees in your organization.	June 2017
<a href="#">My team</a>	You can view your direct reports and extended staff, as well as send praise for individuals in your reporting chain.	June 2017
<a href="#">Invoice approval</a>	Provides a listing of invoices that have been assigned to you through the vendor invoice header workflow process.	June 2017
<a href="#">Expense management</a>	You can capture and upload a receipt, so that you can attach it to an expense report later. The mobile workspace also lets you quickly create an expense line by using an attached receipt.	April 2017
<a href="#">Purchase order approval</a>	View and respond to purchase orders with actions such as approve or reject.	April 2017
<a href="#">Project time entry</a>	You can enter and save time against a project by using your mobile device.	March 2017
<a href="#">Cost controlling</a>	Cost center managers can see the performance of the cost center.	January 2017
<a href="#">Inventory on-hand</a>	Gain insights into reserved and available inventory.	January 2017
<a href="#">Sales orders</a>	You can stay up to date on your sales orders.	January 2017
<a href="#">Vendor collaboration</a>	Vendors can stay up to date on the purchase orders that have been sent to them for approval. They can also view information about new and updated purchase orders and contacts.	January 2017

MOBILE WORKSPACE	DESCRIPTION	AVAILABILITY
<a href="#">Asset management</a>	This workspace lets users view and create maintenance requests and work orders. Users can also view the assigned work order jobs in a calendar or list view. Assets and functional locations can also be viewed and searched for.	October 2019

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Use the Asset management mobile workspace

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic provides information about the **Asset management** mobile workspace. This workspace lets users view and create maintenance requests and work orders. Users can also view the assigned work order jobs in a calendar or list view. Assets and functional locations can also be viewed and searched for.

## Overview

Asset Management is an advanced module for managing assets and work order jobs in Dynamics 365 Supply Chain Management. The **Asset management** mobile workspace lets users quickly view assigned work order jobs on the mobile device of their choice. Users can also create and manage maintenance requests, update lifecycle state, and view asset and functional location details by using their mobile device.

Specifically, the **Asset management** mobile workspace lets users perform these tasks:

- Create, view, and edit maintenance requests, take a photo or attach an existing image to the maintenance request, change the maintenance request lifecycle state.
- Create, view, and edit work orders, take a photo or attach an existing image to the work order, change the work order lifecycle state, view work order jobs.
- View assigned work order jobs in a calendar view.
- Create, view, and edit work order job, update asset counters, view maintenance checklist, view and edit work order job notes, view the tools required for the work order job.
- View or search for a specific asset or functional location.

## Prerequisites

Before you can use the **Asset management** mobile workspace, your admin must set up the required user and worker accounts, and publish the workspace. For more information, see [Set up the Asset management mobile workspace](#).

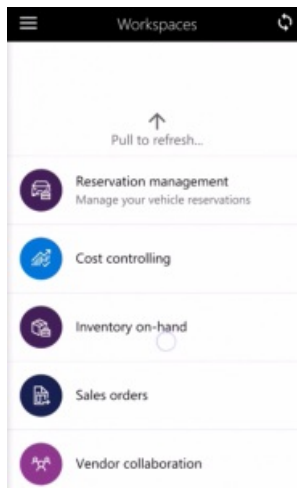
## Download and install the mobile app

Download and install the Dynamics 365 for Unified Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

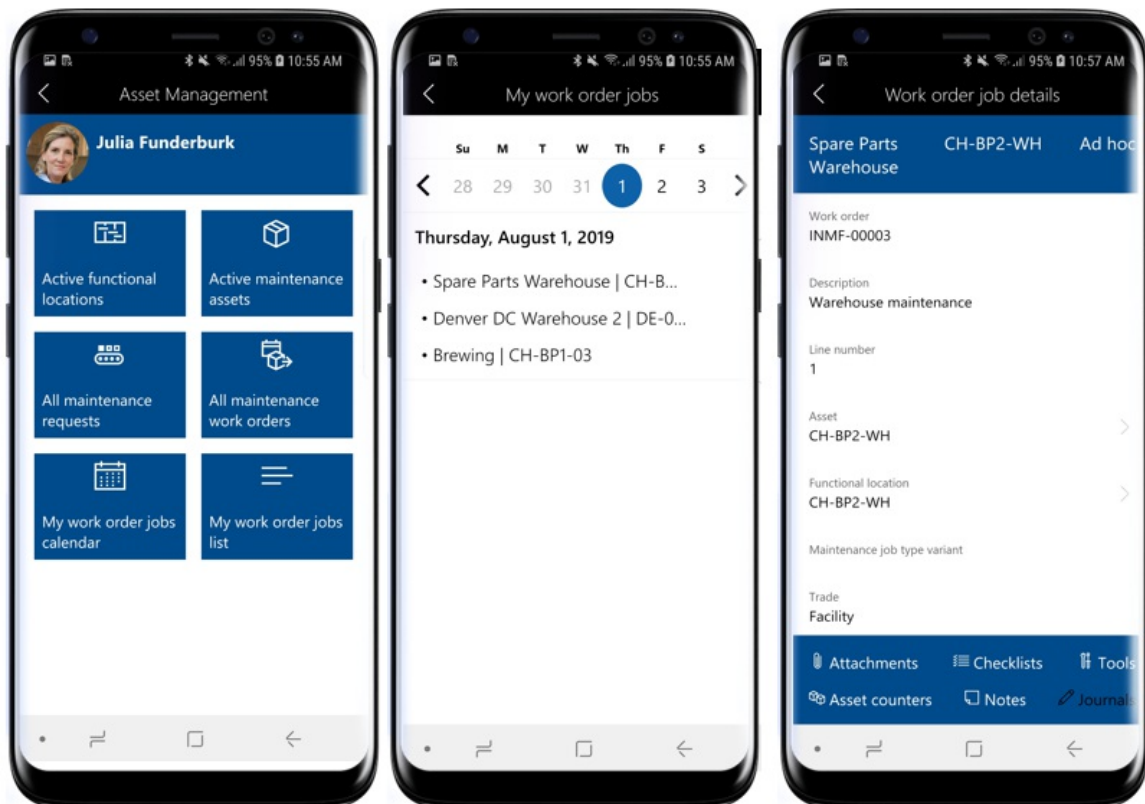
## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you'll have to refresh the list of mobile workspaces.



## View assigned work order jobs in calendar view

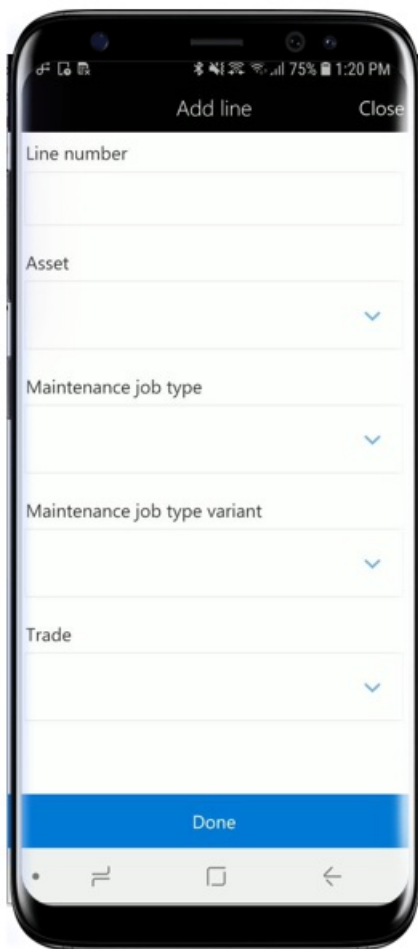
1. On your mobile device, open the **Asset management** workspace.
2. Select **My work order jobs calendar**.
3. Select the date you want to view work order jobs for. In the list, you'll see the asset ID and functional location ID for each work order job.
4. Select a work order job in the list to see job details: Asset and functional location details as well as other navigation links to view **Attachments**, **Checklists**, **Tools**, **Asset counters**, **Notes**, **Journals**.



## Create a work order job

1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance work orders**.
3. Select the work order you want to create a new work order job for.

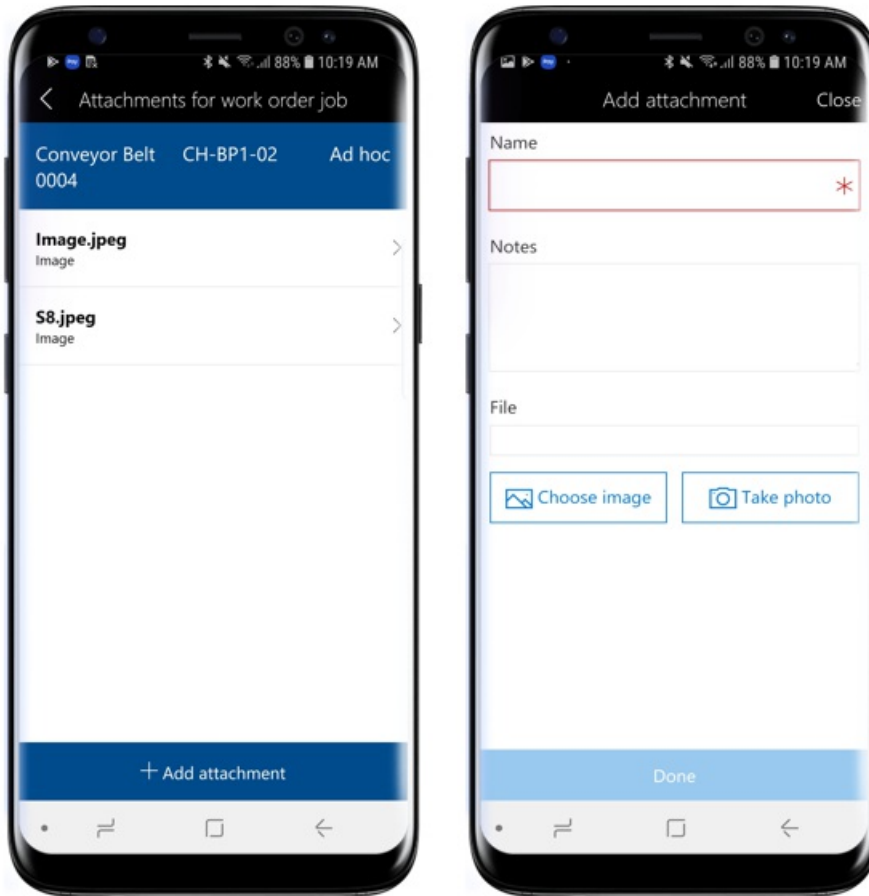
4. Select the **Add line** button.
5. Select the **Asset** you want to create a work order job for.
6. Select **Maintenance job type**, **Maintenance job type variant** and **Trade**.
7. Select **Done**.



## Add attachment to a work order job

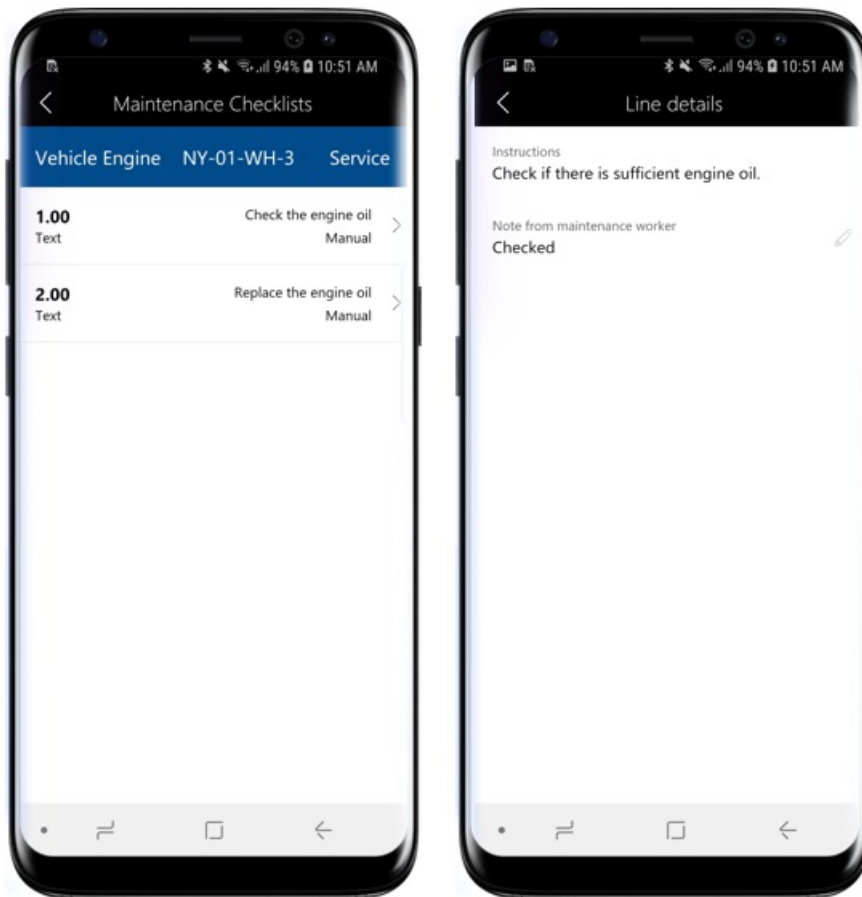
1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance work orders**.
3. Select the work order > work order job you want to add an attachment to.
  - Alternatively, you can also select **My work order jobs calendar** or **My work order jobs list** on the home page to navigate to the **Work order job details** page.
4. Select **Attachments** on the **Work order job details** page.
5. You'll see existing attachments on the work order job. Select **Add attachment**.
6. Enter **Name** and **Notes** for the attachment.
7. Select **Choose image** to select a photo from the mobile gallery, or **Take photo** to take a photo.
8. Select **Done**.





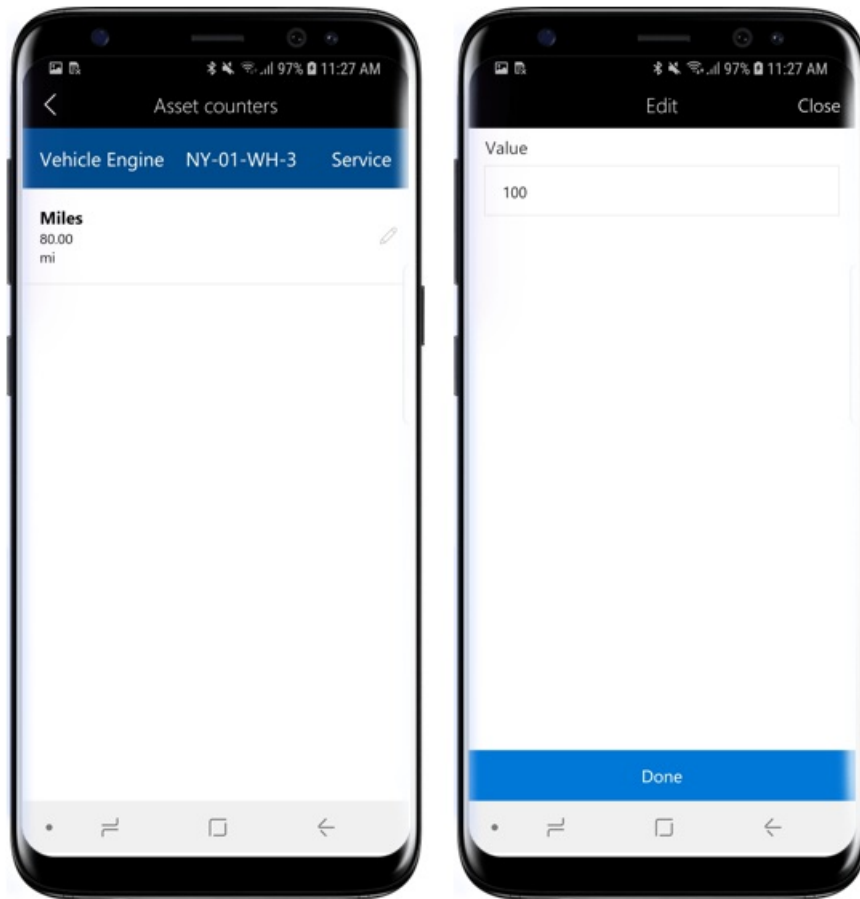
## View maintenance checklist on a work order job

1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance work orders**.
3. Select the work order > work order job you want to view checklists for.
  - Alternatively, you can also select **My work order jobs calendar** or **My work order jobs list** on the home page to navigate to the **work order job details** page.
4. Select **Checklists** on the **Work order job details** page.
5. You'll see a list of checklist lines related to the work order job. Select a checklist line to view **Instructions** and add **Notes**.
6. Select the back button (<) to return to the previous page.



## View and update asset counters on a work order job

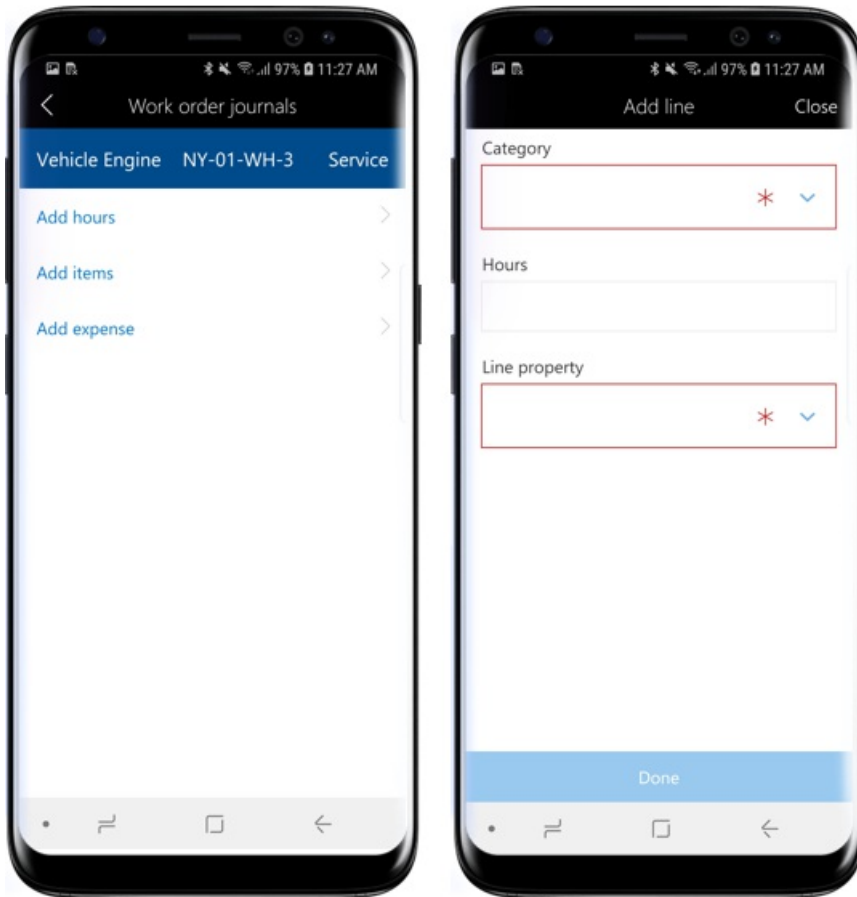
1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance work orders**.
3. Select the work order > work order job you want to view asset counters for.
  - Alternatively, you can also select **My work order jobs calendar** or **My work order jobs list** on the home page to navigate to the **work order job details** page.
4. Select **Asset counters** on the **Work order job details** page.
5. You see a list of asset counters related to the work order job. Select the pencil icon on an asset counter line to update the counter value.
6. Enter a new counter value, and select **Done**.



## Register consumption on a work order job

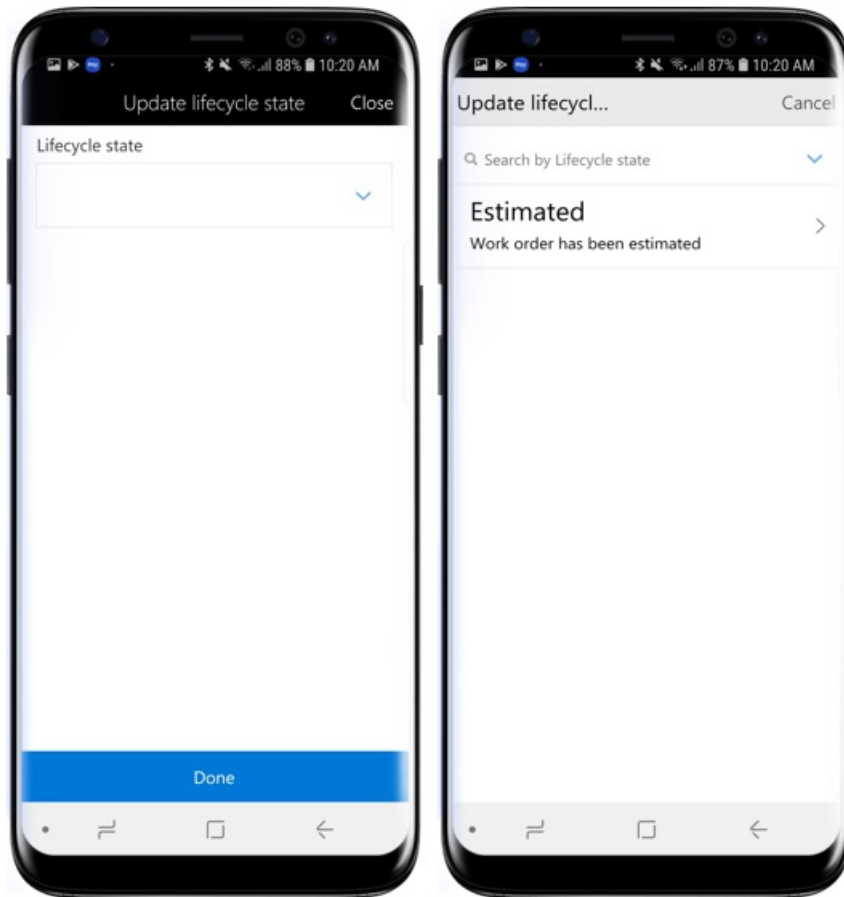
1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance work orders**.
3. Select the work order > work order job you want to add consumption registrations for.
  - Alternatively, you can also select **My work order jobs calendar** or **My work order jobs list** on the home page to navigate to the **work order job details** page.
4. Select **Journals** on the **Work order job details** page.
5. Select **Add hours** to create work hour registrations.
  - a. Select the **Category** from the lookup.
  - b. In the **Hours** field, enter the number of work hours spent on the work order job.
  - c. Select the appropriate **Line property**.
  - d. Select **Done**.
6. Select **Add items** to create item registrations.
  - a. Select the **Item number** from the lookup.
  - b. Select the **Site** from the lookup.
  - c. Enter the **Quantity** of items consumed.
  - d. Select **Done**.
7. Select **Add expense** to create expense registrations.
  - a. Select the **Category** from the lookup.
  - b. Enter the quantity for the expense registration.
  - c. Select the **Sales currency** from the lookup.
  - d. Enter the **Cost price** for the expense registration.

e. Select **Done**.



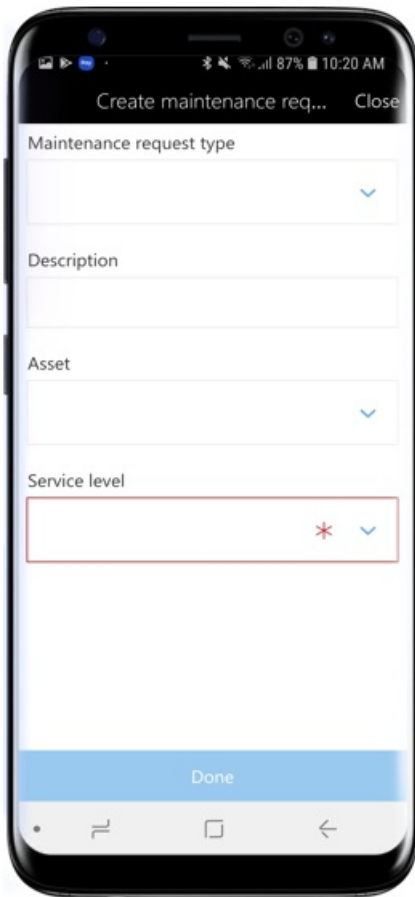
## Update lifecycle state on a work order

1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance work orders**.
3. Select the work order you want to update lifecycle state for.
4. Select the **Update state** button at the bottom of the screen.
5. Select a new lifecycle state from the list.
6. Select **Done**.



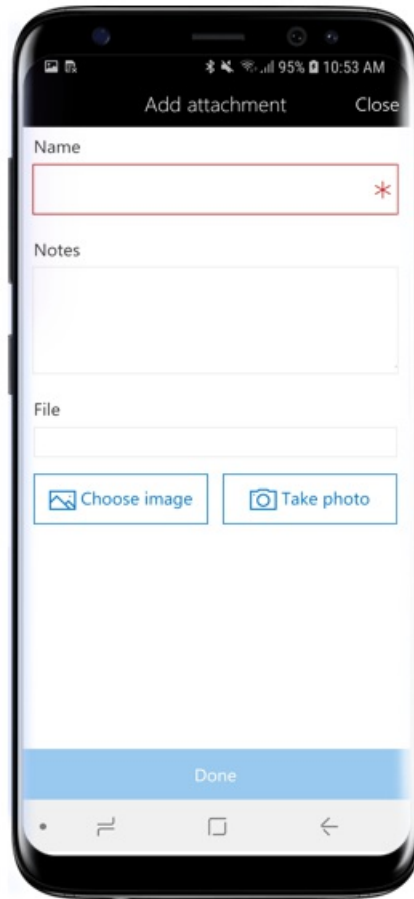
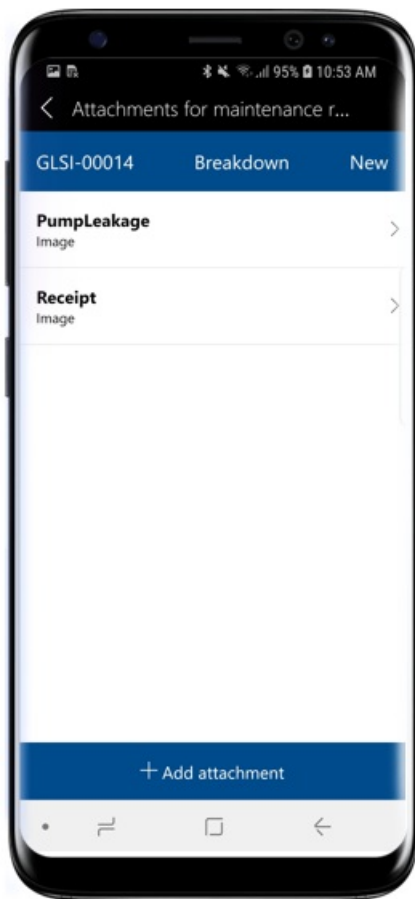
## Create a maintenance request

1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance requests**.
3. Select **Actions** at the bottom of the screen, and select **Create maintenance request**.
4. If number sequence is enabled for maintenance requests in **Asset management**, the **Maintenance request** field is hidden because it is automatically filled out. If the **Maintenance request** field is visible, enter a maintenance request ID.
5. Select a **Maintenance request type**.
6. Enter a **Description** for the maintenance request.
7. Select the **Asset** you want to create the request for.
8. Select the **Service level** for the maintenance request.
9. Select **Done**.



## Add attachment to a maintenance request

1. On your mobile device, open the **Asset management** workspace.
2. Select **All maintenance requests**.
3. Select the maintenance request you want to add an attachment to.
4. Select **Attachments** at the bottom of the screen.
5. Select **Add attachments**.
6. Enter **Name** and **Notes** for the attachment.
7. Select **Choose image** to select a photo from the mobile gallery or **Take photo** to take a photo.
8. Select **Done**.



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Company directory mobile workspace

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides information about the **Company directory** mobile workspace. This workspace lets users view and contact other employees in their organization.

This mobile workspace can be used with the Finance and Operations mobile app.

## Overview

The **Company directory** mobile workspace lets users perform these tasks:

- View a list of employees in the organization.
- Search for employees in the organization.
- View contact information for employees.
- Contact employees from the profile information.

## Prerequisites

Before you can use this mobile workspace, the following prerequisites must be met.

PREREQUISITE	ROLE	DESCRIPTION
One of the following products must be deployed in your organization: <ul style="list-style-type: none"><li>• A Finance and Operations app</li><li>• Microsoft Dynamics 365 Human Resources</li></ul>	System administrator	If you don't already have a Finance and Operations app deployed in your organization, see <a href="#">Deploy a demo environment</a> . If you don't already have Human Resources deployed in your organization, the system administrator can access a trial version from the <a href="#">Human Resources webpage</a> .
The <b>Company directory</b> mobile workspace must be published.	System administrator	See <a href="#">Publish a mobile workspace</a> .

## Download and install the mobile app

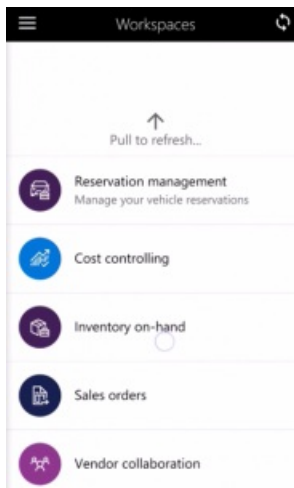
Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Microsoft Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.





## View the company directory by using the mobile workspace

1. In the mobile app, select the **Company directory** workspace. A list of employees is shown.
2. Select an employee. The **Employee profile** page appears. The information on this page includes the employee's first name, last name, title, and department.

## Search the company directory by using the mobile workspace

1. In the mobile app, select the **Company directory** workspace.
2. In the **Search** field, enter an employee's first name, last name, title, or department to start the search.
3. Select an employee. The **Employee profile** page appears. The information on this page includes the employee's first name, last name, title, and department.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cost controlling mobile workspace

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides information about the **Cost controlling** mobile workspace. This workspace lets cost center managers view information about cost center performance anytime and anywhere.

This mobile workspace is intended to be used with the Finance and Operations mobile app.

## Overview

The **Cost controlling** mobile workspace provides an instant view of the current performance of cost centers by comparing actual costs against the budgeted costs. You can drill down to the status of individual cost elements.

For example, an employee receives an invitation to an international conference, but the organization must cover all the travel expenses. The employee asks their manager whether they can attend the conference. The manager opens the **Cost controlling** mobile workspace on their mobile device to see whether there is budget for the employee to attend the conference.

### Data security

The data in the **Cost controlling** mobile workspace is secured through user credentials. Cost center managers are allowed to see data only for their own cost center. The access-level security is managed in the **Cost accounting** module.

Cost accountants define the configuration of the **Cost controlling** mobile workspace in the **Cost accounting** module. After the workspace is published to the mobile app, it's available in the app. Therefore, all cost center managers in the organization can view data in the same format.

### Actions, views, and links

The **Cost controlling** mobile workspace provides the following actions, views, and links:

- **Actions:**
  - Use **Select configuration** to select a layout.
  - Use **Select cost object** to select the cost centers to filter data on.

#### NOTE

The cost centers that appear in the list depend on the access that is granted in the **Cost accounting** module.

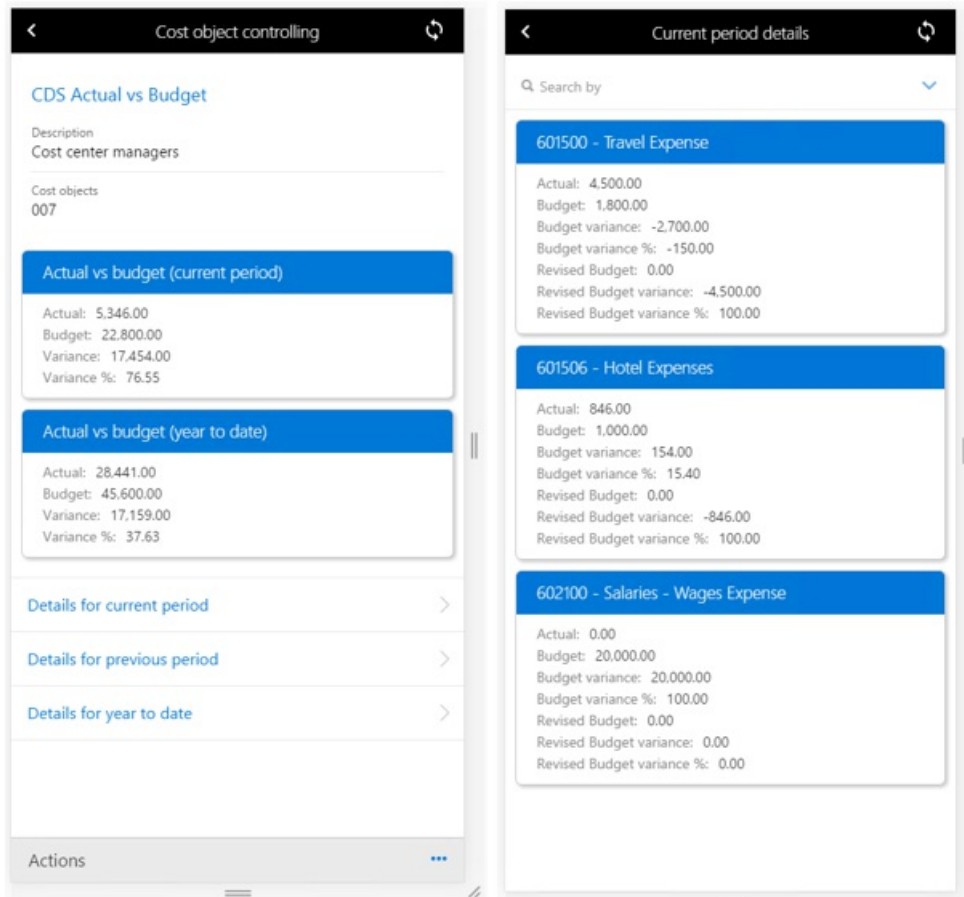
- **Views:** Based on the actions that are selected and the configuration in the **Cost accounting** module, you can view the following information on the cards:
  - Actual vs budget (current period)
  - Actual vs revised budget (current period)
  - Actual vs budget (previous period)
  - Actual vs revised budget (previous period)
  - Actual vs budget (year to date)
  - Actual vs revised budget (year to date)

The following amounts are shown on every card: Actual, Budget, Variance, and Variance %.

- **Links:**

- Details for current period
- Details for previous period
- Details for year to date

When you select a link, a card is shown for each cost element. The following amounts are shown on every card: Actual, Budget, Budget variance, Budget variance %, Revised budget, Revised budget variance, and Revised budget variance %.



## Prerequisites

The prerequisites differ, based on the version of Microsoft Dynamics 365 that has been deployed for your organization.

### Prerequisites if you use Microsoft Dynamics 365 Finance

If Finance has been deployed for your organization, the system administrator must publish the **Cost controlling** mobile workspace. For instructions, see [Publish a mobile workspace](#).

### Prerequisites if you use version 1611 with Platform update 3 or later

If version 1611 with Platform update 3 or later has been deployed for your organization, the system administrator must complete the following prerequisites.

PREREQUISITE	ROLE	DESCRIPTION
--------------	------	-------------

PREREQUISITE	ROLE	DESCRIPTION
Implement KB 4013633.	System administrator	KB 4013633 is an X++ update or metadata hotfix that contains the <b>Cost controlling</b> mobile workspace. To implement KB 4013633, your system administrator must follow these steps. <ol style="list-style-type: none"> <li>1. <a href="#">Download the metadata hotfix from Microsoft Dynamics Lifecycle Services (LCS)</a>.</li> <li>2. <a href="#">Install the metadata hotfix</a>.</li> <li>3. <a href="#">Create a deployable package</a> that contains the <b>SCMMobile</b> model, and then upload the deployable package to LCS.</li> <li>4. <a href="#">Apply the deployable package</a>.</li> </ol>
Publish the <b>Cost controlling</b> mobile workspace.	System administrator	See <a href="#">Publish a mobile workspace</a> .

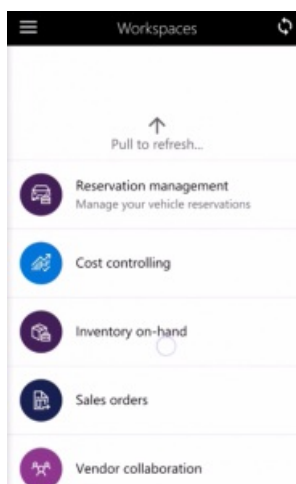
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



## View the performance of your cost center by using the Cost controlling mobile workspace

1. On your mobile device, select the **Cost controlling** workspace.

2. Select **Cost object controlling**.
3. Select **Actions**.
4. Select **Select configuration** to select a cost controlling layout.
5. Select **Done**.
6. Select **Actions**.
7. Select **Select cost object** to select the cost centers that you've been granted access to.
8. Select **Done**.
9. View the overall performance of your cost center.
10. Select the **Details for current period** link.
11. View the performance of individual cost elements.
12. You can also search for specific cost elements.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Inventory on-hand mobile workspace

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides information about the **Inventory on-hand** mobile workspace. This workspace helps you gain insights into reserved and available inventory anytime and anywhere.

This mobile workspace is intended to be used with the Finance and Operations mobile app.

## Overview

Typically, companies have multiple shipments and multiple receipts of inventory every day. These movements constantly change the on-hand inventory status. The **Inventory on-hand** mobile workspace lets you see the cross-company on-hand inventory status, so that you can gain the latest insights into inventory data on the mobile device of your choice. Regardless of whether you work in the warehouse, purchasing, sales, manufacturing, or management, or have other roles, you can access on-hand inventory data anytime and anywhere.

The mobile workspace provides an instant view of the on-hand status across facilities. It lets you view on-hand inventory across facilities, current material reservations, and unreserved on-hand inventory. You can also enter item numbers to query on-hand inventory, and can do a filtered search for on-hand products or variants.

Specifically, the mobile workspace provides these features:

- You can search by product number or product name to find products to view the on-hand inventory status for.
- For the selected products, you can view the following information:
  - On-hand inventory per site
  - On-hand inventory per warehouse
  - On-hand inventory per location
  - On-hand inventory per batch (for batch-controlled products)
  - On-hand inventory per inventory status
- Product on-hand inventory is shown in the following ways:
  - By physical inventory (This view represents the total amount.)
  - By physical reserved (This view represents the reserved amount.)
  - By available physical (This view represents available amount that has no reservations.)

## Prerequisites

The prerequisites differ, based on the version of Supply Chain Management that has been deployed for your organization.

### **Prerequisites if you use Supply Chain Management**

If Supply Chain Management has been deployed for your organization, the system administrator must publish the **Inventory on-hand** mobile workspace. For instructions, see [Publish a mobile workspace](#).

### **Prerequisites if you use Platform update 3 or later**

If Platform update 3 or later has been deployed for your organization, the system administrator must complete the following prerequisites.

PREREQUISITE	ROLE	DESCRIPTION
Implement KB 4013633.	System administrator	KB 4013633 is an X++ update or metadata hotfix that contains the <b>Inventory on-hand</b> mobile workspace. To implement KB 4013633, your system administrator must follow these steps. <ol style="list-style-type: none"> <li>1. <a href="#">Download the metadata hotfix from Microsoft Dynamics Lifecycle Services (LCS)</a>.</li> <li>2. <a href="#">Install the metadata hotfix</a>.</li> <li>3. <a href="#">Create a deployable package</a> that contains the <b>SCMMobile</b> model, and then upload the deployable package to LCS.</li> <li>4. <a href="#">Apply the deployable package</a>.</li> </ol>
Publish the <b>Inventory on-hand</b> mobile workspace.	System administrator	See <a href="#">Publish a mobile workspace</a> .

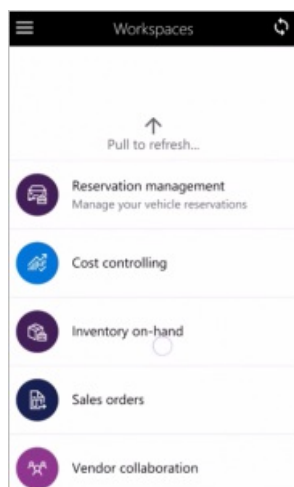
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



View the on-hand inventory for a product by using the Inventory on-hand mobile workspace

1. On your mobile device, select the **Inventory on-hand** workspace.
2. Select **Check on-hand for an item**. You see a list of the products that are loaded into your app for offline use. By default, 50 items are loaded, but a developer can change this number. For more information, developers should see [Mobile platform](#).
3. If your item isn't in the list, select **Search more**. Search by product number, or switch to a search by product name.
4. Select a product. If the item has an image, the image is shown.
5. Select one of the following options to view the status of on-hand inventory:
  - View on-hand per site
  - View on-hand per warehouse
  - View on-hand per location
  - View on-hand per batch (for batch-controlled products)
  - View on-hand per inventory status

Product on-hand inventory is shown in the following ways:

- By physical inventory (This view represents the total amount.)
- By physical reserved (This view represents the reserved amount.)
- By available physical (This view represents the available amount that has no reservations.)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Invoice approvals mobile workspace

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides information about the **Invoice approvals** mobile workspace. This workspace provides a list of invoices that have been assigned to you through the vendor invoice header workflow process.

This mobile workspace is intended to be used with the Finance and Operations mobile app.

## Overview

The **Invoice approvals** mobile workspace lets Accounts payable clerks and managers view invoices that have been assigned to them as part of the vendor invoice header workflow process. You can view the invoice information, and even the line and distribution details, to help you make informed approval decisions. From the workspace, you can take action to move the invoice through the workflow process.

## Prerequisites

Before you can use this mobile workspace, the following prerequisites must be met.

PREREQUISITE	ROLE	DESCRIPTION
Microsoft Dynamics 365 Finance must be deployed in your organization.	System administrator	See <a href="#">Deploy a demo environment</a> .
The <b>Invoice approvals</b> mobile workspace must be published.	System administrator	See <a href="#">Publish a mobile workspace</a> .

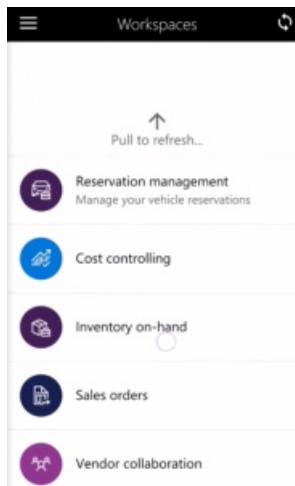
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Microsoft Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



## Approve invoices by using the Invoice approvals mobile workspace

1. On your mobile device, select the **Invoice approvals** workspace.
2. Select the invoice that has been assigned to you by the vendor invoice header workflow process.
3. On the **Invoice details** page, review the invoice header information, such as the vendor and date information.
4. Select a line on the invoice to view more detailed information about it in the **Invoice line details** view.
5. In the **Invoice line details** view, select **Distributions** to show the line distributions. Here, you can view the accounting for the invoice line. The information that is shown includes the financial dimensions and the main account.
6. On the **Invoice details** page, select **Distributions** to show all distributions. Here, you can view the accounting for the whole invoice. The information that is shown includes the financial dimensions and the main accounts.
7. Select **Attachments** to view any notes or files that are attached to the invoice.
8. On the **Invoice details** page, select the appropriate workflow action to complete your review process.
9. Select **Done**.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# My team mobile workspace

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides information about the **My team** mobile workspace. This workspace lets managers view their direct reports and extended staff. They can also send praise to individuals in their reporting chain.

This mobile workspace is intended to be used with the Finance and Operations mobile app.

## Overview

The **My team** mobile workspace lets managers perform these tasks:

- View a list of the manager's direct reports.
- View a list of the manager's extended reporting team.
- View detailed information for each team member, such as birth date, seniority date, years of service, and compensation and performance information.
- Send praise to any individual in the manager's extended reporting team.

## Prerequisites

Before you can use this mobile workspace, the following prerequisites must be met.

PREREQUISITE	ROLE	DESCRIPTION
One of the following products must be deployed in your organization: <ul style="list-style-type: none"><li>• A Finance and Operations app</li><li>• Microsoft Dynamics 365 Human Resources</li></ul>	System administrator	If you don't already have a Finance and Operations app deployed in your organization, see <a href="#">Deploy a demo environment</a> . If you don't already have Human Resources deployed in your organization, the system administrator can access a trial version from the <a href="#">Human Resources webpage</a> .
The <b>My team</b> mobile workspace must be published.	System administrator	See <a href="#">Publish a mobile workspace</a> .

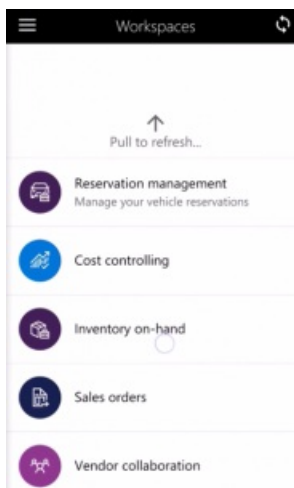
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Microsoft Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



## View team members by using the My team mobile workspace

1. In the mobile app, select the **My team** workspace. A list of team members is shown. The list also shows each team member's title, and any direct reports that the member has.
2. Select a team member. The **Team member summary** page appears. The information on this page includes the team member's birth date, seniority date, years of service, years in the current position, and compensation information.

## View extended team members by using the My team mobile workspace

1. In the mobile app, select the **My team** workspace. A list of team members is shown. The list also shows each team member's title, and any direct reports that the member has.
2. Select the **Direct reports** link. A list of your extended team is shown.
3. Select a team member. The **Team member summary** page appears. The information on this page includes the team member's birth date, seniority date, years of service, years in the current position, and compensation information.

## Send praise about team members by using the My team mobile workspace

1. In the mobile app, select the **My team** workspace. A list of team members is shown. The list also shows each team member's title, and any direct reports that the member has.
2. Select a team member. The **Team member summary** page appears.
3. Select **Send praise**.
4. Enter the text of the praise that you want to send.
5. Select **Done**.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Purchase order approval mobile workspace

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides information about the **Purchase order approval** mobile workspace. This workspace lets you view purchase orders and respond to them through actions. For example, you can approve or reject a purchase order.

## Overview

Purchase orders that requires approval go through an approval workflow. The workflow can include various steps that require that one or more people take action. For example, a person might have to complete a task or approve the purchase order.

The **Purchase order approval** mobile workspace lets you easily view and respond to purchase orders from your mobile device. This workspace also lets you take the same workflow actions that you can take from the web client.

## Prerequisites

The prerequisites vary, depending on the version of Supply Chain Management that has been deployed for your organization.

### Prerequisites if you use Supply Chain Management

If Supply Chain Management has been deployed for your organization, the system administrator must publish the **Purchase order approval** mobile workspace. For instructions, see [Publish a mobile workspace](#).

### Prerequisites if you use Microsoft Dynamics 365 for Operations version 1611 with Platform update 3 or later

If Microsoft Dynamics 365 for Operations version 1611 with Platform update 3 or later has been deployed for your organization, the system administrator must complete the following prerequisites.

PREREQUISITE	ROLE	DESCRIPTION
Implement KB 4017918.	System administrator	KB 4017918 is an X++ update or metadata hotfix that contains the <b>Purchase order approval</b> mobile workspace. To implement KB 4017918, your system administrator must follow these steps. <ol style="list-style-type: none"><li>1. <a href="#">Download the metadata hotfix from Microsoft Dynamics Lifecycle Services (LCS)</a>.</li><li>2. <a href="#">Install the metadata hotfix</a>.</li><li>3. <a href="#">Create a deployable package</a> that contains the <b>SCMMobile</b> model, and then upload the deployable package to LCS.</li><li>4. <a href="#">Apply the deployable package</a>.</li></ol>
Publish the <b>Purchase order approval</b> mobile workspace.	System administrator	See <a href="#">Publish a mobile workspace</a> .

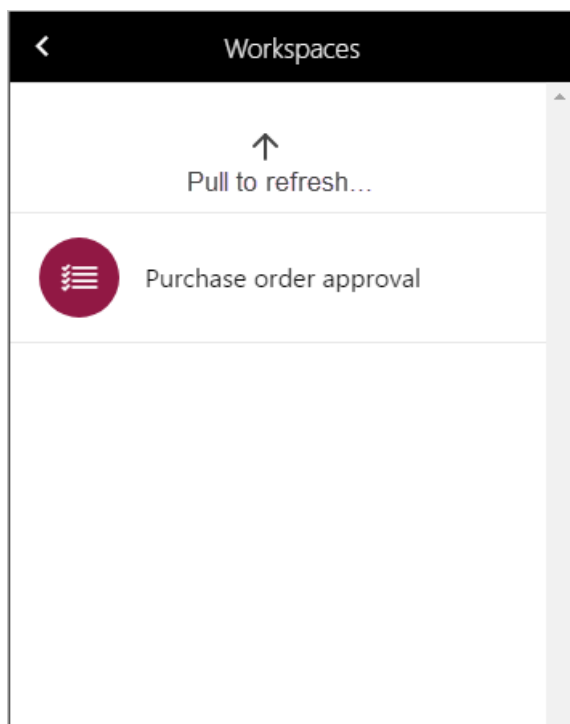
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Microsoft Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



## View orders that are assigned to you

1. On your mobile device, select the **Purchase order approval** workspace.
2. Select **Orders assigned to me** to view all the purchase orders for which you've been asked to take action in the purchase order approval workflow.
3. Select an order. On the **Order details** page, you will see the order header information and lines. You can also find guidelines from the workflow task.
4. Select **Accounting distributions** to open the **Header accounting distributions** page.
5. Return to the **Order details** page, and select a line. From the order line details, you can also explore the line-specific accounting distributions.

## Complete an action on the purchase order

After you've viewed the purchase order that is assigned to you and read the workflow instructions, you should be ready to take action.

1. On your mobile device, select the **Purchase order approval** workspace.
2. Select **Orders assigned to me** to view all the purchase orders for which you've been asked to take

action in the purchase order approval workflow.

3. Select an order, and view the details page.
4. Select **Actions** to show the available actions. The actions that are available depend on the task that has been assigned to you.

TASK ACTION	APPROVAL ACTION
Complete	Approve
Return	Reject
Request change	Request change
Delegate	Delegate

5. Select the appropriate action.
6. On the **Complete task** page, enter a comment. Note that if you select the **Delegate** action, you must select a user to delegate the task to.
7. Select **Done**. After you refresh your workspace, the purchase order will no longer be in your list.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Sales orders mobile workspace

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides information about the **Sales orders** mobile workspace. This workspace helps you stay up to date about your sales orders anywhere and anytime.

This mobile workspace is intended to be used with the Finance and Operations mobile app.

## Overview

The **Sales orders** mobile workspace lets you view detailed information about each sales order. This information includes the status of the order, contact information for the customer, and contact information for the order taker. The **Sales orders** mobile workspace provides an instant view of sales orders. You can view all sales orders, view sales orders by customer, or view information about a specific sales order.

The mobile workspace provides two views to help you analyze sale orders in depth.

### **View all sales orders**

This view lists all sales orders.

- Use one of the following filters to select the sales orders to view:
  - Search by sales order
  - Search by customer account
  - Search by customer name
  - Search by status
  - Search by release status
  - Search by created date and time
- After you select sales orders, you can view the details of specific orders. Specifically, you can view the following information:
  - Customer name and address information
  - Various dates for the sales order, such as the requested ship date and the confirmed ship date
  - Contact information for the order taker
  - Customer contact information
  - Order lines
  - Shipments that show how and when a sales order was shipped

### **View orders for a customer**

This view lists sales orders by customer.

- Use one of the following filters to view orders for a customer:
  - Search by name
  - Search by account
- After you select a customer, you can view the following information:
  - Customer name and group
  - Customer contact information
  - Customer sales orders and details about those sales orders:



- Customer name and address information
- Various sales order dates
- Contact information for the order taker
- Customer contact information
- Order lines
- Shipments that show how and when a sales order was shipped

## Prerequisites

The prerequisites differ, based on the version of Microsoft Dynamics 365 that has been deployed for your organization.

### Prerequisites if you use Supply Chain Management

If Supply Chain Management has been deployed for your organization, the system administrator must publish the **Sales orders** mobile workspace. For instructions, see [Publish a mobile workspace](#).

### Prerequisites if you use Dynamics 365 for Operations version 1611 with platform update 3 or later

If Dynamics 365 for Operations version 1611 with platform update 3 or later has been deployed for your organization, the system administrator must complete the following prerequisites.

PREREQUISITE	ROLE	DESCRIPTION
Implement KB 4013633.	System administrator	KB 4013633 is an X++ update or metadata hotfix that contains the <b>Sales orders</b> mobile workspace. To implement KB 4013633, your system administrator must follow these steps. <ol style="list-style-type: none"> <li>1. <a href="#">Download the metadata hotfix from Microsoft Dynamics Lifecycle Services (LCS)</a>.</li> <li>2. <a href="#">Install the metadata hotfix</a>.</li> <li>3. <a href="#">Create a deployable package</a> that contains the <b>SCM Mobile</b> model, and then upload the deployable package to LCS.</li> <li>4. <a href="#">Apply the deployable package</a>.</li> </ol>
Publish the <b>Sales orders</b> mobile workspace.	System administrator	See <a href="#">Publish a mobile workspace</a> .

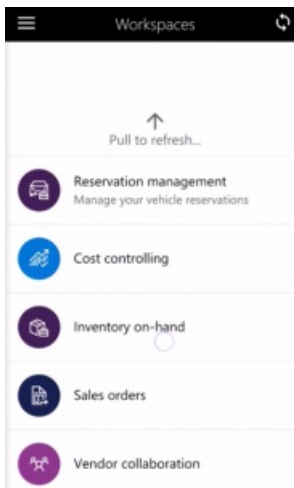
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company is shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



## View information about sales orders for a customer by using the Sales order mobile workspace

1. On your mobile device, select the **Sales orders** workspace.
2. Select **View orders for a customer**.
3. Use account or customer name information to find the customer.
4. Select the customer.
5. Select **Contact information** or **Sales orders**. If you select **Sales orders**, a list of sales orders for the customer is shown.
6. Select **Sales order**. You can now view information about sales order lines, information about shipments, customer contact information, and contact information for the order taker.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Vendor collaboration mobile workspace

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic provides information about the **Vendor collaboration** mobile workspace. This workspace helps your vendors stay up to date about the purchase orders that have been sent to them for approval. They can also view information about new and updated purchase orders and contacts.

This mobile workspace is intended to be used with the Finance and Operations mobile app.

## Overview

The **Vendor collaboration** mobile workspace keeps vendors informed about new purchase orders, so that they can view purchase orders and then respond to them in the web client.

### NOTE

The mobile workspace should be used as a supplement to the vendor collaboration web interface, not a replacement for it.

Your vendors can use the **Vendor collaboration** mobile workspace to view new purchase orders that are sent to them for approval. It shows purchase order information, such as products, quantities, and requested delivery dates. Price information is also available, depending on the configuration of each vendor.

A user who signs in as a vendor will see which purchase orders have been responded to, and which purchase orders are still awaiting customer action. For example, a purchase order might be awaiting customer action because the vendor suggested another delivery date, but the customer hasn't yet agreed to that date. The vendor will also see a list of purchase orders that have been confirmed but haven't yet been delivered.

To respond to a purchase order, the vendor must use the vendor collaboration web interface that is available in the web client. There, the vendor can also get more information about the order, such as document attachments, the delivery address per line, and charges that are associated with the vendor.

Vendors that have a special security role can see which contact persons are registered for a vendor account. The same security role lets a vendor view the status of any user request that has been submitted.

The vendor collaboration web interface in the web client must be used to create new contacts and submit new user requests.

The **Vendor collaboration** mobile workspace lets a vendor perform these tasks:

- View new purchase orders that are sent to the vendor.
- View purchase orders that the vendor has responded to, and that are awaiting customer action.
- View purchase orders that have been confirmed but haven't yet been fully received.
- View contact person information that is registered for the vendor account. (This task requires an additional security role.)
- View information about a user request that was submitted by the vendor, and follow the status of the request. (This task requires an additional security role.)

## Prerequisites

The prerequisites vary, depending on the version of Microsoft Dynamics 365 that has been deployed for your

organization.

### Prerequisites if you use Supply Chain Management

If Supply Chain Management has been deployed for your organization, the system administrator must publish the **Vendor collaboration** mobile workspace. For instructions, see [Publish a mobile workspace](#).

### Prerequisites if you use Microsoft Dynamics 365 for Operations version 1611 with Platform update 3 or later

If Microsoft Dynamics 365 for Operations version 1611 with Platform update 3 or later has been deployed for your organization, the system administrator must complete the following prerequisites.

PREREQUISITE	ROLE	DESCRIPTION
KB 3216943 must be implemented if you're using Platform update 3.	System administrator	KB 3216943 is a binary update that is required if you're using Platform update 3. To implement this KB, the system administrator must follow these steps. <ol style="list-style-type: none"><li>1. Download KB 3216943 from Microsoft Dynamics Lifecycle Services (LCS).</li><li>2. Install the binary update, which is delivered as a deployable package. For information about how to apply a deployable package, see <a href="#">Apply a deployable package</a>.</li></ol>
KB 4013633 must be implemented.	System administrator	KB 4013633 is an X++ update or metadata hotfix that contains the <b>Inventory on-hand</b> mobile workspace. To implement KB 4013633, your system administrator must follow these steps. <ol style="list-style-type: none"><li>1. <a href="#">Download the metadata hotfix from LCS</a>.</li><li>2. <a href="#">Install the metadata hotfix</a>.</li><li>3. <a href="#">Create a deployable package</a> that contains the <b>SCMMobile</b> model, and then upload the deployable package to LCS.</li><li>4. <a href="#">Apply the deployable package</a>.</li></ol>
The <b>Vendor collaboration</b> mobile workspace must be published.	System administrator	See <a href="#">Publish a mobile workspace</a> .
The vendor user must have access to the vendor collaboration web interface in the web client and must set up a vendor collaboration user.	Purchasing professionals and the system administrator	Follow the steps in the following topics to set up and work with the vendor collaboration web interface. <ul style="list-style-type: none"><li>• <a href="#">Use vendor collaboration to work with external vendors</a></li><li>• <a href="#">Manage vendor collaboration users</a></li><li>• <a href="#">Set up and maintain vendor collaboration</a></li><li>• <a href="#">Use vendor collaboration to work with customers in Supply Chain Managements</a></li></ul>

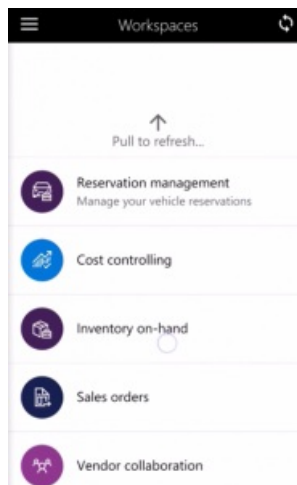
## Download and install the mobile app

Download and install the Finance and Operations mobile app:

- [For Android phones](#)
- [For iPhones](#)

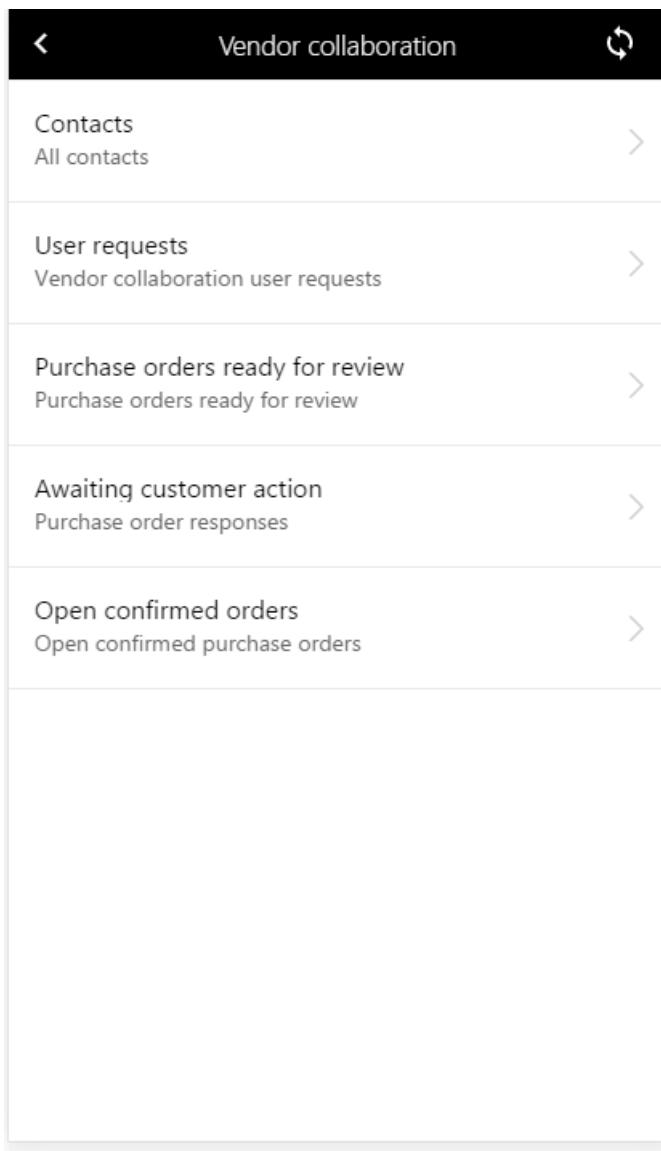
## Sign in to the mobile app

1. Start the app on your mobile device.
2. Enter your Microsoft Dynamics 365 URL.
3. The first time that you sign in, you're prompted for your user name and password. Enter your credentials.
4. After you sign in, the available workspaces for your company are shown. Note that if your system administrator publishes a new workspace later, you will have to refresh the list of mobile workspaces.



## Use the Vendor collaboration mobile workspace

When you select the **Vendor collaboration** workspace, you'll see the following options.



The **Vendor collaboration** workspace includes the following pages.

### **Contacts**

The **Contacts** page lets you see all the contacts that have been set up for the vendor account. It shows the contact person's name, primary email address, and user alias, if the contact person has an alias. This page also shows whether the contact person's user account is active. When you select a contact, you see contact details, such as the legal entities that the person is a contact for. You also see contact information, such as a telephone number or an alternative email address.

### **User requests**

The **User requests** page lets you see all the user requests that you've submitted via the vendor collaboration web interface. You can also follow the status of those requests. When you select a user request, you can see what was requested, add or inactivate a user, change security, and see which security roles were requested for the user.

### **Purchase orders ready for review**

The **Purchase orders ready for review** page lets you see all the purchase orders that the customer has sent, but that haven't yet been responded to. You can view selected information about the order, such as which products were requested and when those products should be delivered. Price information is also available, depending on the configuration of the vendor.

You can also see whether the purchase order has notes or attachments. However, to open notes and attachments, you must use vendor collaboration web interface in the web client. Select **Purchase order line** to see all the lines together with their details. For each line, an indicator will show whether there are notes or

attachments, or whether the delivery address differs from the delivery address that is shown on the header.

To respond to the purchase order, you must use the vendor collaboration web interface in the web client.

### **Awaiting customer action**

The **Awaiting customer action** page lets you find purchase orders that you or another person in your company who has access to vendor collaboration has responded to. The purchase orders are visible in this list only if the customer must take one of the following actions on the purchase order:

- If the purchase order was rejected, the customer must either update or cancel the original order, and then send it again. When the purchase order is sent again, it no longer appears on the **Awaiting customer action** page.
- If the purchase order was accepted with changes, the customer must either update the original order and then send it again for review, or update the order per the requested changes and then confirm it immediately. In both cases, the purchase order no longer appears on the **Awaiting customer action** page.
- If the purchase order was accepted but still appears on the **Awaiting customer action** page, the purchase order wasn't automatically confirmed when it was accepted. It's now waiting for a purchasing agent to change the order status to **Confirmed**. Typically, a purchase order is considered an agreement between the customer and the vendor as soon as the vendor accepts the order. Therefore, the update to **Confirmed** status is usually just a formality.

When you select a purchase order, additional details appear about the response. You can see the line details and response for every line. The line status shows which of the following responses has been given:

- Accepted
- Rejected
- Accepted with changes
- Substituted/Substitute
- Split into schedule/Schedule line

Note that the **Delivering** field is set to either **Yes** or **No** to indicate whether the lines will be delivered. A line might not be delivered because for the following reasons:

- The line was rejected.
- A substitution was made, and the original line isn't expected to be delivered as requested in the received order.
- The line was split into multiple schedule lines, and the original line isn't expected to be delivered as requested in the received order.

Any changes that have been made to the order line response are shown. However, uploaded notes and attachments aren't shown. To view notes and attachments, you must use the vendor collaboration web interface in the web client.

### **Open confirmed orders**

When the purchase order is confirmed by the customer (that is, the status of the purchase order is changed to **Confirmed**), it appears in the open confirmed order. It will remain in the list until it's registered as received by the customer.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Publish mobile workspaces

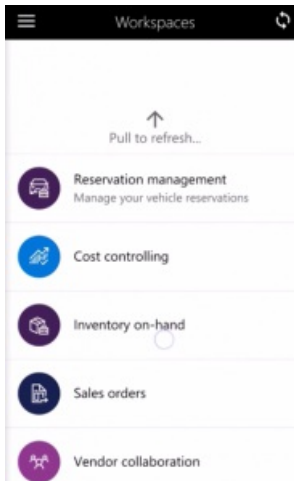
2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the steps that system administrators must follow to publish a mobile workspace. A mobile workspace must be published so that users can access it in the mobile app.

## Publish a mobile workspace

1. In your browser, start your web client.
2. Click **Settings** > **Mobile app**.
3. Select the mobile workspace to publish.
4. Click **Publish**.

After a new workspace is published, users must pull to refresh the list of mobile workspaces.



### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Office integration overview

2/18/2021 • 6 minutes to read • [Edit Online](#)

Applies to these Dynamics 365 apps:

Commerce, Finance, Supply Chain Management

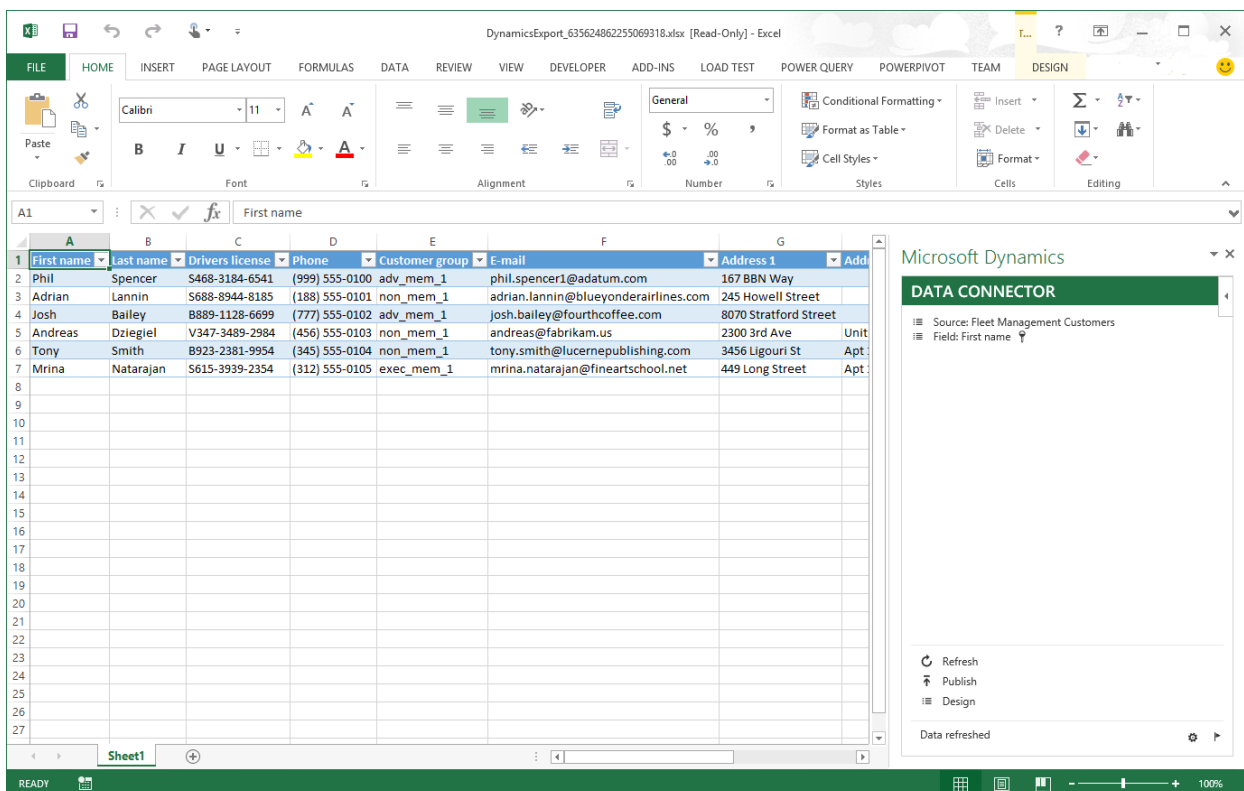
This topic reviews Microsoft Office integration concepts and features. The integration depends on several technologies:

- Working in Microsoft Azure
- Working with Azure Active Directory (Azure AD)
- Running a web client in multiple browsers

The Microsoft Office integration capabilities provide users with a productive environment that helps them get the job done by using Office products.

## Excel Data Connector add-in

Microsoft Excel can change and quickly analyze data. The Excel Data Connector app interacts with Excel workbooks and OData services that are created for publicly exposed data entities. The Excel Data Connector add-in enables Excel to become a seamless part of the user experience. The Excel Data Connector add-in is built by using the Office Web add-ins framework. The add-in runs in a task pane. Office Web Add-ins are web applications that run inside an embedded Internet Explorer browser window.



1	First name	Last name	Drivers license	Phone	Customer group	E-mail	Address 1	Add.
2	Phil	Spencer	S468-3184-6541	(999) 555-0100	adv_mem_1	phil.spencer1@adatum.com	167 BBN Way	
3	Adrian	Lannin	S688-8944-8185	(188) 555-0101	non_mem_1	adrian.lannin@blueyonderairlines.com	245 Howell Street	
4	Josh	Bailey	B889-1128-6699	(777) 555-0102	adv_mem_1	josh.bailey@fourthcoffee.com	8070 Stratford Street	
5	Andreas	Dziewiel	V347-3489-2984	(456) 555-0103	non_mem_1	andreas@fabrikam.us	2300 3rd Ave	Unit
6	Tony	Smith	B923-2381-9954	(345) 555-0104	non_mem_1	tony.smith@lucernepublishing.com	3456 Ligouri St	Apt.
7	Mrina	Natarajan	S615-3939-2354	(312) 555-0105	exec_mem_1	mrina.natarajan@fineartschool.net	449 Long Street	

## Dynamics AX 2012 architecture vs. Finance and Operations architecture

There are several differences between versions. For both, we built lightweight add-ins that run in Excel and use services to connect to the application.

### Dynamics AX 2012

Excel > VSTO (.NET) Add-in > Windows Communication foundation (WCF) > Authentication through Active

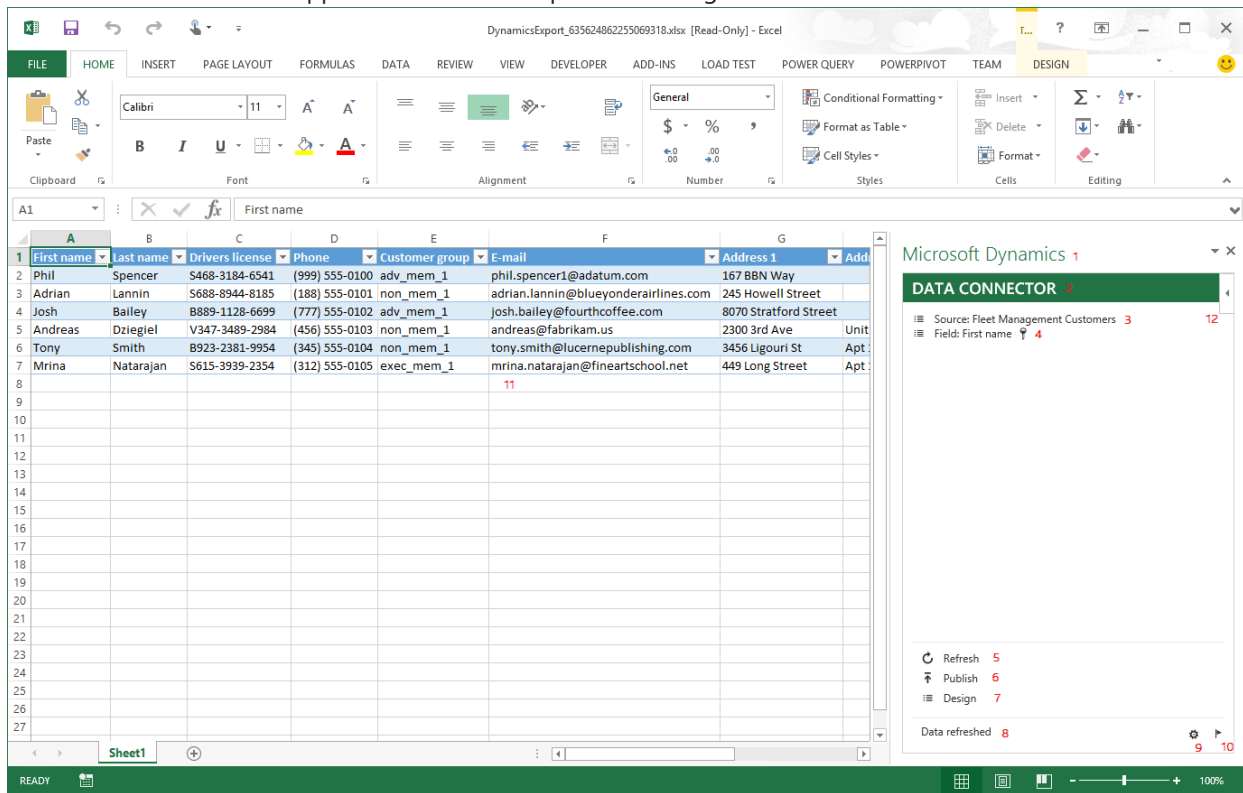
Directory (AD) > AIF SOAP services on the AOS > AX Services and Tables > AX query engine > Database

### Finance and Operations

Excel > Office Web Add-in (JS + HTML) > JavaScript OData API (Olingo) > Authentication through Azure Active Directory (AAD) > AX OData services on the AOS > AX Entities > AX LINQ provider > AX Database

### Office Add-in explained

The Excel Data Connector app is located in a task pane on the right side of a workbook.



The following table describes the parts of the add-in. The numbers correspond to the numbers in the preceding screen shot.

NUMBER	NAME	DESCRIPTION
1	Add-in primary title	The title of the add-in that is provided to the Office Web Add-ins framework.
2	Add-in secondary title	The title of the add-in that is provided by the add-in.
3	Source name	The label of the entity that provides data for the selected data table. You can hover over the label to see the corresponding name.
4	Field name	The label of the field that provides data for the selected data table column. You can hover over the label to see the corresponding name and type.
5	Refresh button	Refresh the data in the workbook.
6	Publish button	Publish the data changes in the workbook.
7	Design button	Open the design-time experience.

NUMBER	NAME	DESCRIPTION
8	Status bar	The status bar provides brief temporary information alerts. Information that is appears in the status bar also appears in the <b>Messages</b> dialog box.
9	Options button	Open the <b>Options</b> dialog box.
10	Messages button	Open the <b>Messages</b> dialog box, which displays the information messages, warnings, and errors that the program provides to the user. A number sometimes appears next to the <b>Messages</b> button to provide a count of the warnings or errors that the user might be interested in.
11	Excel data table containing data	The filter and sort controls in the columns headers can be used on this data. The filters must be removed before data changes are published.
12	Office Web Add-ins menu	The Office Web Add-ins menu button provides several standard links. The most important of the links is used to reload the add-in. When the add-in is reloaded, it updates all the data for the workbook that is contained in tables that are associated with the add-in.

## Authentication

OData sits on the same authentication stack as the server. The add-in uses OAuth to facilitate authentication.

## Lookups and drop-down lists

When you click in a table cell, any lookup, enumeration drop-down list, or date picker that is associated with that cell will be shown inside the add-in, underneath the source and field information. Any value that you select inside the add-in is put into the currently selected table cell.

## Adding and deleting records

To add a record, either start typing in a row directly below a table, or use the Tab key to tab away from the last cell of the last row in the table. To delete a record, select the row by clicking the row label (1, 2, 3, and so on), and delete all the cells in that row. To publish the changes, click **Publish**. The **Messages** dialog box shows how many records were added, edited, and deleted.

## Workbook Designer

You can use the **Workbook Designer** page to design an editable custom export workbook that contains an entity and a set of fields. To open the **Workbook Designer (ExportToExcelWorkbookDesigner)** page, click **Common > Common > Office Integration > Excel workbook designer**. Before you can publish data edits, all the key fields of the entity must be in the Excel table. Key fields have a key symbol next to them. To successfully create or update a record, it must have all the mandatory fields in the Excel table. Mandatory fields have an asterisk (\*) next to them.

#### AVAILABLE FIELDS

🔑	Vehicle rental ID
*	Drivers license
*	First name
*	Last name
*	Start date

To retrieve the resulting workbook, click **Create workbook** in the app bar.

Click **View related form** to see the data that the entity exposes. This button is only enabled for entities that have a **FormRef** property value.

## Document management

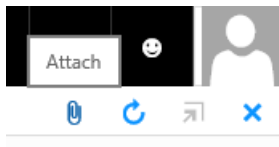
Document management supports saving record attachments in Azure Blob storage and SharePoint Online. Database storage is deprecated. Azure Blob storage is equivalent to storage in the database since documents can only be accessed through the application and it provides the added benefit of providing storage that doesn't negatively affect the performance of the database. Azure blob storage is the default and works immediately. SharePoint storage will work immediately if you have an O365 license since we auto-discover the SharePoint tenant e.g. a user on the TenantA.onmicrosoft.com O365/AAD tenant gets TenantA.sharepoint.com as the SharePoint site. If document management has been turned off by the user, turn it on by clicking **Options > General > Miscellaneous** and setting **Document handling active** to **Yes**.

#### DOCUMENT HANDLING

Document handling active

Yes

On any page that has data, an **Attach** button will be available in the upper-right corner.



The **Attachments** page provides a view of the attachments (documents) that are associated with the record that was selected on the previous page. You can add new attachments to the record by clicking the **New** button (+) in the app bar. For the **File** and **Image** document types, you will be prompted to provide the associated file.

### Document preview

A preview for supported file types is provided on the **Preview** FastTab. Basic document types, such as PNG images and text files, are supported by default. Office document types, such as Microsoft Word, Excel, and PowerPoint files, must use a production Office Web Apps Server, which might not be available in a OneBox configuration.

## Frequently asked questions

### Office Licensing

#### What Microsoft 365 licenses are available?

There are lots of [Microsoft 365 license options](#). You should select the license that makes sense for your organization.

#### After purchasing a Microsoft 365 license, what needs to be done to set up SharePoint storage for attachments?

Open the Document Parameters form and ensure that the SharePoint server has been automatically discovered

and set. Now open or create a Document Type, set the Document Type's location to "SharePoint" and select the folder that the files should be stored in.

## Additional resources

[Office integration tutorial](#)

[Troubleshoot the Office integration](#)

[Application stack and server architecture](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Office integration tutorial

2/18/2021 • 27 minutes to read • [Edit Online](#)

## Applies to these Dynamics 365 apps:

Commerce, Finance, Human Resources, Supply Chain Management

In this tutorial, you will use and build Office integration experiences that involve Excel, Word, Document Management, and email.

## Overview

In this tutorial, you will use and build Microsoft Office integration experiences that involve Microsoft Excel, Microsoft Word, the Document Management subsystem, and email. You will see how Excel and Word use data entities as an entry point into the system, how Excel can become a core part of the user experience, and how Excel and Word can be used for ad-hoc lightweight reporting. You will also see how files can be stored and shared by using the Document Management and email capabilities.

## Prerequisites

For this tutorial, you must access the environment by using Remote Desktop, and you must be provisioned as an administrator on the instance. For more information, see [Deploy and access development environments](#). If you're running Internet Explorer on the virtual machine (VM), you must enable font and file downloads at **Internet Options > Security > Custom Level**. Microsoft Visual Studio 2015 runs on the VM, and it must run as an administrator so that metadata and compilation files can be overwritten. To make sure that Visual Studio runs as an administrator, search for the program, and pin it to the taskbar. Then right-click the shortcut on the taskbar, right-click **Visual Studio 2015**, click **Properties > Advanced**, and select the **Run as administrator** check box. Visual Studio will now run as an administrator via a single left click of the taskbar shortcut.

## Key concepts

- **Entities and OData** – You will use the Microsoft Dynamics Excel Data Connector App (Excel App) to create, read, update, and delete. The connector uses OData services that are created for any entity that is left in the default state of "public" (**DataEntity.Public=Yes**).
- **Apps for Office** – The Excel App is built by using the Apps for Office framework (which is also known as the Office Web API). The Excel App is web-based, and therefore shares technology with the client and will run inside both on-premises Excel instances and Microsoft Excel Online (Microsoft 365). The app runs inside Excel in a task pane.
- **Microsoft Office 2016** – The Excel and Word Apps use advances in the Apps for Office framework that were introduced in Office 2016. Therefore, Office 2016 is required in order to run the Excel and Word Apps.
- **Authentication** – The Excel and Word Apps run in an Internet Explorer window inside Excel and Word. Even if the user is running the client in an InPrivate Browsing session in Internet Explorer, or in a different browser, such as Chrome, Internet Explorer is still used to run the app inside Excel or Word. Authentication is facilitated by OAuth, and the user can select accounts and sign in within the app. Internet Explorer will first try to automatically sign the user in. Therefore, if you aren't signed in as the correct user, or if you have trouble signing in, you might have to force a sign-out from the app by using the sign-out link on the user menu in the lower-right corner of the app. After sign-out, right-click in the app, and try to sign in again.
- **Excel App** – In addition to facilitating refresh and publish data operations, the Excel App also provides source and field information, lookups, filtering, error messaging, and a design experience for adding or removing fields, table columns, or labels from entity data sources.

# Setup

## Load the Fleet data set

During this tutorial, we will mainly use forms, entities, and data in the Fleet Management model. Therefore, we must first load the Fleet data set.

1. Navigate to **Fleet Management > Setup > Fleet setup**.
2. Click **Create**.

## Static Export to Excel experiences

### Static Export to Excel

Static Export to Excel provides a quick mechanism for getting data into grids on a page. The standard mechanism for triggering Export to Excel is the **Open in Microsoft Office** menu. Static Export to Excel is also available via a shortcut menu on the grid.

1. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
2. Click **Open in Microsoft Office > Export to Excel > Customers**.
3. Download and open the workbook that is generated. Note that the columns in the workbook match the columns in the grid.
4. Select ("mark") the first two rows by clicking in the left edge of the row, below the "Select all" check mark.
5. Right-click the grid header to open the shortcut menu. Note that both **Export all rows** and **Export marked rows** are available as commands.
6. Click **Export marked rows**. Note that the columns in the workbook match the columns in the grid, and that the rows that are exported match the rows that you marked.

### Modify the static Export to Excel experience

You can suppress the static Export to Excel mechanism for a grid or change the label that appears on the **Open in Microsoft Office** menu.

1. Start Visual Studio 2015. Make sure that it's running as an administrator.
2. Click **View > Application Explorer** (or press Ctrl+E, Ctrl+E).
3. Navigate to **AOT > User Interface > Forms > FMCustomer**.
4. Right-click **FMCustomer**, and then click **Add to new project**.
5. In Solution Explorer, double-click the **FMCustomer** form to open the designer view.
6. Select **FMCustomerDesignTab(Tab)TabPageGrid(TabPage)MainGrid(Grid)**.
7. In the **Properties** window, find the **Export Label** property.
8. Set the **Export Label** property to **Fleet Customers**.
9. Save the form. If you're asked whether you want to overwrite the existing form or save it as a new form, click **Overwrite**.
10. Build the solution (press Ctrl+Shift+B).
11. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
12. Click **Open in Microsoft Office**. Note that the **Customers** option has changed to **Fleet Customers**.

## Generated Open in Excel experiences

### Generated Open in Excel

Generated Open in Excel options are automatically added to forms when the system finds data entities that have the same root data source as the form. The workbook that is generated will contain a single table data source where the data from that entity is loaded. The Open in Excel experiences are listed on the **Open in Microsoft Office** menu. (When an entity has the same root data source as a form, it's added as an option in the **Open in**

Excel section of the **Open in Microsoft Office** menu. This option is referred to as a “generated” option.)

1. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
2. Click **Open in Microsoft Office > Open in Excel > Fleet Management Customers (unfiltered)**.
3. Download and open the workbook that is generated. This workbook contains the Excel Data Connector App, a binding to the **Fleet Management Customer** entity, and a pointer to the server that the workbook was generated from.
4. Click **Enable editing** to enable the Excel Data Connector App to load. Customer data is read from the OData service on the server and added to the table.
5. In Internet Explorer, on the **Customer** page, click **Edit** (or press F2), and change the email address of one of the customers.
6. In the Excel App, click **Refresh**. Note that the new email address is shown in Excel.
7. In Excel, change the email address of one of the customers.
8. In the Excel App, click **Publish**.
9. In Internet Explorer, click **Refresh** in the upper right of the page (or press Shift+F5). Note that the new email address is shown on the **Customer** page.
10. In Excel, click the **Settings** (gear) button in the lower-right corner of the Excel App. You can use the dialog box that appears to adjust the settings in the current workbook. Note that the **Server URL** value matches the start of the URL that is shown in Internet Explorer. Also note that the data refresh and data publish operations are listed.
11. Click **Cancel** to close the **Settings** dialog box.
12. Click the **Message Center** (flag) button in the lower-right corner of the Excel App. The message center dialog box that appears provides information about what is occurring in the Excel App.

### **Add and remove table columns from an existing table data source in the Excel App**

The Excel App has a design experience that lets users add and edit bindings to entity data sources and labels. To add and remove fields from an existing binding, you use the edit experience that is outlined in the following steps.

1. Get a workbook that has an existing table data source:
  - a. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
  - b. Click **Open in Microsoft Office > Open in Excel > Fleet Management Customers (unfiltered)**.
  - c. Download and open the workbook that is generated. This workbook contains the Excel Data Connector App, a binding to the Fleet Management Customer entity, and a pointer to the server that the workbook was generated from.
  - d. Click **Enable editing** to enable the Excel Data Connector App to load. Customer data is read from the OData service on the server and added to the table.
2. Open the data source for editing:
  - a. In Excel, in the Excel App, click **Design**. A list of table and field data sources appears.
  - b. Click the **Edit** (pencil) button next to the existing table data source. The data source details are shown.
3. Remove fields. In the **Selected** list, double-click a field. Alternatively, click a field, and then click **Remove**. To select multiple fields, keep the Ctrl key held down while you click them. To select all fields, press Ctrl+A.
4. Add fields. In the **Available** list, double-click a field. Alternatively, click a field, and then click **Add**. To select multiple fields, keep the Ctrl key held down while you click them. To select all fields, press Ctrl+A.
5. Change the field order. In the **Selected** list, click a field, and then click **Up** or **Down**.
6. Change a field label. In the **Selected** list, click a field, and then click in the **Column label** field below the list. You can change the label to either a static string or a label identifier that will be translated to the active language (for example, @SYS129977).



7. Apply the changes that you made to data source fields:
  - a. Click **Update** to return to the data source list.
  - b. Click **Refresh** to make sure that any new fields are filled with data.

### **Change an automatically generated Open in Excel experience**

The automatically generated Open in Excel experiences that are created for entities have a single table binding. The list of fields that are added to that table binding is defined by the **AutoReport** field group if the table binding contains fields. Otherwise, the key and mandatory fields for the entity are automatically added. The order of fields in the **AutoReport** group determines the order of fields in the table binding.

1. Start Visual Studio 2015. Make sure that it's running as an administrator.
2. Click **View** > **Application Explorer** (or press Ctrl+E, Ctrl+E).
3. Navigate to **AOT** > **Data Model** > **Data Entities** > **FMCustomerEntity**.
4. Right-click **FMCustomerEntity**, and then click **Add to project**.
5. Expand **FMCustomerEntity** > **Field Groups** > **AutoReport**.
6. Reverse the order of the **First name** and **Last name** fields by clicking the **Last name** field and moving it up (press Alt+Up arrow key).
7. Save the entity. If you're asked whether you want to overwrite the existing entity or save it as a new entity, click **Overwrite**.
8. Build the solution (press Ctrl+Shift+B).
9. In Internet Explorer, navigate to **Fleet Management** > **Customers** > **Customer**.
10. Click **Open in Microsoft Office** > **Open in Excel** > **Fleet Management Customers**.
11. Open the workbook that is generated.
12. Click **Enable editing** to enable the Excel Data Connector App to load. Note that the **Last name** column appears before the **First name** column.

### **Open in Excel Online**

The Excel App is built by using a new Apps for Office framework. This framework provides a JavaScript-based web application programming interface (API) that enables apps to communicate with Office applications. The biggest advantage of this new framework is that apps can run in on-premises Excel instances (Win32), Excel Online (Microsoft 365), and Excel on the Apple iPad. They will also be able to run in other Excel apps in the future.

1. Navigate to **Fleet Management** > **Customers** > **Customer**.
2. Click **Open in Microsoft Office** > **Open in Excel** > **Fleet Management Customers**.
3. Click **SharePoint**.
4. Browse to the desired Microsoft SharePoint folder.
5. Click **Save**. The default behavior is to open the file after it's saved. Note that the workbook opens in Excel Online. In Excel Online, capabilities of the Excel App, such as refresh and publish, and the design experience, should work just as they work in on-premises Excel instances.

## Template Open in Excel experiences

### **Template Open in Excel**

Template options resemble the generated Open in Excel options. They are automatically added to forms when the system finds templates that have the same first data source as the root data source in the form. A workbook template can have multiple data sources. It can also have unbound content. The Open in Excel experiences are listed on the **Open in Microsoft Office** menu. The **Excel workbook designer** page provides an easy way to get a generated Open in Excel experience for an entity. It also provides a mechanism getting a blank workbook that contains just the Excel App and a pointer to the server.

1. In Internet Explorer, navigate to **Common > Common > Office integration > Excel workbook designer**.
2. Select the **FleetCustomer** entity.
3. Add all fields in the list of available fields to the list of selected fields.
4. Click **Create workbook**.
5. Open the workbook that is generated. This workbook contains the Excel Data Connector App, a binding to the **Fleet Management Customer** entity, and a pointer to the server that the workbook was generated from.
6. Click **Enable editing** to enable the Excel Data Connector App to load. Customer data is read from the OData service on the server and added to the table.
7. Insert a blank row above the table, and enter **Fleet Customers** as the title.
8. Rename the worksheet **FleetCustomers**.
9. Rearrange some of the fields in the table. Click **Design** to open the design experience.
10. Next to the **FleetCustomer** data source, there are buttons for editing and deleting the data source. Click **Edit** to see the field list.
11. Select fields, and move them as you require. Set the order for the first three fields to **FirstName**, **LastName**, and **DriverLicense**.
12. Click **Update**. Note that the field order is changed.
13. Click **Done**.
14. Click the **Settings** (gear) button.
15. Click **Clear binding data** so that the workbook contains no bound data.
16. Click **OK**.
17. Save the workbook as **FleetCustomersBasic.xlsx**.
18. In Internet Explorer, navigate to **Common > Common > Office integration > Document templates**.
19. Click **New**.
20. Browse to the file that you just saved.
21. Click **OK**. The template is added as a line in the templates table.
22. In the **FleetCustomersBasic** row, clear the **Apply current record filter** check box, so that an unfiltered list of customers will be loaded after the template is opened.
23. Change the **Template display name** value to **Fleet Customers Basic**.
24. Navigate to **Fleet Management > Customers > Customer**.
25. Click **Open in Microsoft Office**. Note that **Fleet Customers Basic** is now an option in the **Open in Excel** section. Click that option.
26. Open the workbook that is generated.
27. Click **Enable editing** to enable the Excel Data Connector App to load. Customer data is read from the OData service on the server and added to the table binding that you created.

### **Register a template as a system-defined template**

Templates that are registered as system-defined templates are loaded at deployment. This behavior is useful for independent software vendors (ISVs) and partners that want to package templates together with other model artifacts.

1. Start Visual Studio 2015 by opening the previously created project where the model is set to **Fleet Management**, or create a new project.
2. Right-click the project, and then click **Add > New item**.
3. Select the **Resource** item type.
4. Set the name to **FleetCustomersBasicTemplate**.
5. Make sure that the **FleetCustomersBasic.xlsx** file is closed.

6. Click **Add**.
7. Select the **FleetCustomersBasic.xlsx** file. Note that the resource is added to the project.
8. Click **View** > **Application Explorer** (or press Ctrl+E, Ctrl+E).
9. Navigate to **AOT** > **Classes** > **Code** > **FMTemplateRegistrations**.
10. Right-click **FMTemplateRegistrations**, and then click **Add to project**.
11. Open **FMTemplateRegistrations**. The **FMTemplateRegistrations.xpp** code file should be shown.
12. Copy one of the existing lines, and change it by providing the template name, resource name, description, display name, and **Apply current record filter** and **List in Open in Office menu** values. The display name is the text that appears as an Open in Excel option. The description appears when the user holds the pointer over that item. The display name and description can be either labels or static strings. The code should resemble the following example.

```
this.addTemplate(  
    OfficeAppApplicationType::Excel,  
    resourceStr(FleetCustomersBasicTemplate),  
    resourceStr(FleetCustomersBasicTemplate),  
    "Template for fleet customers", "Fleet customers basic", NoYes::No, NoYes::Yes);
```

13. Save the code. If you're asked whether you want to overwrite the existing code or save it as a new file, click **Overwrite**.
14. Build the solution (press Ctrl+Shift+B).
15. Verify that the change was successful. In Internet Explorer, navigate to **Common** > **Common** > **Office integration** > **Document templates**.
16. Click **Reload system templates**.
17. Click **Yes** to confirm that you want to reload the system templates.
18. Verify that the new system-defined template is loaded, and that the template name is **FleetCustomersBasicTemplate**.

### **Journal Entry in Excel experience powered by a template**

1. In Internet Explorer, navigate to **General ledger** > **Journal entries** > **General journals**.
2. Make sure that you're in company **USMF**.
3. Create a new journal by clicking **New**.
4. Set the name to **GenJrn**.
5. Click **Open lines in Excel**.
6. Open the workbook that is generated, and enable editing as required. Note that header fields are filled with data.
7. Enter a new line, and set the **MainAccount** field to **110110**. Enter a description, a currency, and a debit amount. Note that lookups are provided for the company and currency fields, because those relationships are defined for this entity.
8. Click **Publish**. Note that the line is updated with the current date and a debit amount of 0 (zero).
9. In Internet Explorer, click **Lines**. Note that line that you entered in Excel is shown.

## Lookups in Excel experiences

### **Lookups in the Excel App**

To facilitate data entry, the Excel App provides lookups and data assistance. Date fields provide a date picker,

enumeration (enum) fields provide an enum list, and relationships provide a relationship lookup.

1. In Internet Explorer, navigate to **Fleet Management > Rentals > Rental**.
2. Click **Open in Microsoft Office > Open in Excel > Fleet Management Rentals**.
3. Open the workbook that is generated.
4. Click **Enable editing** to enable the Excel Data Connector App to load and read in data.
5. Click a **Drivers license** value. Note that a relationship lookup now appears in the Excel App and shows a list of customers. Because relationship lookups are in their first generation, no filtering or sorting is currently supported.
6. Click another customer in the lookup, and note that the **Drivers license** value changes. Because this field is part of the key, click the original **Drivers license** value to reset it. Note that the **Drivers license**, **First name**, and **Last name** fields form a multi-part key, but the Excel App doesn't immediately change all parts of the multi-part key.
7. Click a **Start date** value. Note that a date picker now appears in the Excel App.
8. Click another date to change the **Start date** value.
9. Click **Design**, and edit the FleetRental data source by adding the **Status** field as a column in the table binding.
10. When you've finished adding the **Status** column, click a **Status** value. Note that an enum list now appears in the Excel App.
11. While focus is in the **Status** column, move up and down the list of rentals to see how quickly the enum list changes to reflect the current value. The whole enum list is shown, so that the user can quickly see all the available values.
12. Click a different **Status** value to see how an enum value can be changed by using a single click.

### **Create a relationship lookup**

When relationships exist between entities, a relationship lookup is shown.

1. Start Visual Studio 2015 by opening the previously created project where the model is set to **Fleet Management**, or create a new project.
2. Click **View > Application Explorer** (or press Ctrl+E, Ctrl+E).
3. Navigate to **AOT > Data Model > Tables > FMCustGroup**.
4. Right-click, and then click **Open designer**.
5. In the designer, right-click **FMCustGroup**, and then click **Add-ins > Create data entity**. Artifacts are added to the project.
6. Open the designer view for **FMCustGroupEntity**.
7. In the property sheet for **FMCustGroupEntity**, set **Public Collection Name** to **FleetCustomerGroups** and **Public Entity Name** to **FleetCustomerGroup**.
8. Add the **CustGroup** and **Description** fields to the **AutoLookup** field group.
9. If **FMCustomerEntity** isn't already in the project, add it.
10. Open the designer view for **FMCustomerEntity**.
11. Right-click **Relations**, and then click **New > Relation**.
12. On the new relation, set **Name** to **CustomerGroup**, **Cardinality** to **ZeroMore**, **RelatedDataEntity** to **FMCustGroupEntity**, **RelatedDataEntityCardinality** to **ZeroOne**, **RelationshipType** to **Association**, **Role** to **CustomerGroupSource**, and **RelatedDataEntityRole** to **CustomerGroupTarget**.
13. Build the solution (press Ctrl+Shift+B).
14. Verify that the change was successful. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
15. Click **Open in Microsoft Office > Open in Excel > Fleet Management Customers**.
16. Open the workbook that is generated.
17. Click a **Customer group** value.

18. Change the **Customer group** value for a customer.
19. Publish the change.
20. Change the value back, and publish that change.

### Create a custom lookup

You can create custom lookups to show data options when an enum or relationship isn't sufficient. The main use case is when data must be retrieved from an external service and presented in real time.

1. Start Visual Studio 2015 by opening the previously created project where the model is set to **Fleet Management**, or create a new project.
2. Open the designer view for **FMCustomerEntity**.
3. Right-click **Methods**, and then click **New Method**.
4. Add the **lookup\_Country** code from the following example.

```
public class FMCustomerEntity extends common
{
    [SysODataActionAttribute("FMCustomerEntityCountryCustomLookup", false), //Name in $metadata
    SysODataCollectionAttribute("_fields", Types::String), //Types in context
    SysODataFieldLookupAttribute("Country")] //Name of field
    public static str lookup_Country(Array _fields)
    {
        OfficeAppCustomLookupListResult result = new OfficeAppCustomLookupListResult();

        result.items().value(1, "US");
        result.items().value(2, "AU");
        result.items().value(3, "FR");
        result.items().value(4, "GR");
        result.items().value(5, "NZ");

        return result.serialize();
    }
}
```

5. Save the code. If you're asked whether you want to overwrite the existing code or save it as a new file, click **Overwrite**.
6. Build the solution (press Ctrl+Shift+B).
7. Verify that the change was successful. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
8. Click **Open in Microsoft Office > Open in Excel > Fleet Management Customers**.
9. Open the workbook that is generated.
10. Click a **Country** value.
11. Change the **Country** value for a customer.
12. Publish the change.
13. Change the value back, and publish that change.

## Export to Word experiences

### Export to Word

Export to Word experiences can be used for lightweight reporting. They are powered by pre-built templates. The Export to Word experiences are listed on the **Open in Microsoft Office** menu. Let's look at an example

experience that has been created for Fleet Management Customers.

1. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
2. Click **Open in Microsoft Office > Export to Word > Customer information Fleet Management Customers (unfiltered)**.
3. Download and open the document that is generated. The document contains data from the record that is currently selected.

### Create a Word template

The Microsoft Dynamics App for Office can be run in Word to enable the creation of templates that can then be used for document generation.

1. Add a trusted catalog that points to the file share that contains the Microsoft Dynamics App manifest:
  - a. In Word, click **File > Options**.
  - b. Click **Trust Center > Trust Center Settings**.
  - c. Click **Trusted Add-in Catalogs**.
  - d. In the **Catalog URL** field, enter the file share location of the manifest.
  - e. Click **Add catalog**.
  - f. Click **OK**.
  - g. Click **OK**.
  - h. Restart Word.
2. Add the Microsoft Dynamics App to a document:
  - a. In Word, click **Insert > My Add-ins > Shared Folder > Microsoft Dynamics**.
  - b. Click **Insert**.
  - c. In the app, click **Add server information**.
  - d. In the **Server URL** field, enter the start of the URL (protocol + hostname). For example, enter `https://topo00dfa4stbobaos.cloudax.test.dynamics.com`.
  - e. Click **OK**.
  - f. Click **Yes** to apply the settings change and reload the app.
3. Sign in to the app:
  - a. Click **Sign In**. The Azure Active Directory sign-in screen should provide a list of credentials. If you encounter an error, force a sign-out (by using the sign-out link in the lower-right corner of the app), and then sign in again.
  - b. Select the appropriate account, or click **Use another account**.
  - c. Enter the credentials for that environment, and then click **Sign in**.
4. Load the template designer applet:
  - a. After sign-in, click **Load applets**.
  - b. Select **Template Designer**.
  - c. Click **OK**.
  - d. Click **Yes** to confirm.
  - e. Click **OK** to close the settings page. The latest OData metadata is loaded.
5. Follow one of these steps:
  - Add a fields data source:
    - a. In the app, click **Design**.
    - b. Click **Add fields**.
    - c. Select **FleetCustomer**.
    - d. Click **Next** to go to the field selection page.

- e. In the document, add a title and/or some blank lines at the top of the document.
  - f. In the app, in the **Available** list, select the **FirstName** field.
  - g. Click **Add label** to add a content control that references the "First name" label.
  - h. In the document, click to put focus into the document, click again to put focus at the end of the label, and then press the Right arrow key until the cursor is outside the content control (the control box will disappear).
  - i. Add a separator, such as space+hyphen+space (" - ") or space+colon+space (" : ").
  - j. In the app, click **Add value** to add a content control that references the **FirstName** field.
  - k. Repeat the process for the **LastName** field label and value.
  - l. Continue to add fields as desired.
- Add a table data source:
    - a. In the app, click **Design**.
    - b. Click **Add table**.
    - c. Select **FleetCustomer**.
    - d. Click **Next** to go to the field selection page.
    - e. In the document, add a title and/or some blank lines at the top of the document.
    - f. In the app, in the **Available fields** list, add the **FirstName**, **LastName**, and **City** fields.
    - g. Click **Done**.
6. In Word, save the template document.

### Create a Word template and use it for document generation

After you've built a Word template, you can upload it to create an Export to Word experience.

1. Upload a template:
  - a. Navigate to **Common > Common > Office integration > Document templates**. Alternatively, search for the page.
  - b. Click **New**.
  - c. Click **Browse**.
  - d. In the dialog box, select a previously created template, and then click **Open**. Note that the Root data entity is obtained from the template and appears near the bottom of the dialog box.
  - e. Click **OK**.
  - f. Scroll down the list of templates to confirm that the template was added.
  - g. Optional: If the template should not be filtered to the user's current record, clear the **Apply current record filter** check box.
  - h. Optional: If the template should not be filtered to the user's current company, clear the **Apply company filter** check box.
2. Use the uploaded template for document generation:
  - a. Navigate to a page that shares the same root data source as the template's root data entity. For **FleetCustomer (FMCustomerEntity)**, that page is **Fleet Management > Customers > Customer**.
  - b. Click **Open** in **Microsoft Office > Export to Word**, and click the template.
  - c. Download and open the document that is generated.

## Document Management and SharePoint experiences

### Add a SharePoint document type

The Document Management subsystem can be used to attach files to records. Most non-executable file types are supported as attachments. A document preview is provided for Office document files and PDFs. Administrators

create document types to indicate where attachments should be stored. When administrators use SharePoint as the storage location, they must provide a specific folder that the files should be put in. Security of that SharePoint folder is a separate administration responsibility.

1. In Internet Explorer, navigate to **Organization administration > Document management > Document management parameters**.
2. Click **SharePoint**.
3. Make sure that the **Default SharePoint server** field is set to a default value for the tenant, such as **contosoax7.sharepoint.com**.
4. Click **Test SharePoint connection**. Note a successful connection.
5. Click **Save**.
6. Navigate to **Organization administration > Document management > Document types**.
7. Click **New**.
8. Set **Type** to **SharePointDoc**.
9. Set **Name** to **SharePointDoc**.
10. Set **Location** to **SharePoint**.
11. Click the **Edit** (pencil) button next to the **SharePoint Address** field.
12. On the left side of the dialog box, select a site. For the contosoax7 tenant, select **ContosoAX Team Site**.
13. On the right side of the dialog box, select a folder. For the contosoax7 tenant, select **ContosoAX Team Site > Documents > OfficeIntegration > Attachments**.
14. Click **OK**.
15. Click **Save**.
16. Click the **Browse** (globe) button next to the **SharePoint Address** field. Note that a new browser tab that shows the selected folder appears.
17. Use Windows Explorer to create a Word document in the Documents folder, and enter a few words in the document.
18. In Internet Explorer, navigate to **Fleet Management > Customers > Customer**.
19. Put focus on the first customer, and then click the **Attach** (paperclip) button in the upper-right corner of the page.
20. Click **New > SharePointDoc**.
21. Click **Browse**, and select the Word document that you created.
22. Expand the **Preview** FastTab to see a preview of the Word document.
23. Expand the **Attachment** FastTab to see the file location of the Word document.
24. In Internet Explorer, use the previously opened tab that shows the SharePoint folder to double-check that the file has been placed appropriately.

## Email experiences

### Send mail via a local mail client

Email workflows that are enabled via the SysEmail framework can generate email messages (.eml files) that contain attachments. You can then send these messages via Microsoft Outlook or another email client.

1. In Internet Explorer, navigate to **Accounts receivable > Customers > All customers**.
2. Select **US-008 Sparrow Retail**.
3. Click **Collect > Customer balances > Collections** to open the **Collections** page.
4. Click **Communicate > Email > Statements to contact**.
5. Click **OK** to accept the default values in the dialog box.
6. If you're prompted for the mail option to use, clear the **Do not ask again** check box (you can change this option from the user options page), select **Use an email app, such as Outlook**, and then click **OK**.



7. If you're running Internet Explorer on your laptop, open the email (.eml) file that is generated. If you're running Internet Explorer on the VM, copy the file to your laptop, and open it there.
8. Note the email address in the **To** field and the generated workbook attachment.

### Send mail via SMTP

Email workflows that are enabled via the SysEmail framework can also be created in a simple email dialog box and then sent via Simple Mail Transfer Protocol (SMTP).

1. In Internet Explorer, navigate to **System administration > Setup > Email > Email parameters**.
2. Click **SMTP settings**.
3. Set the **Outgoing mail server** to the desired SMTP server:
  - For [Microsoft 365 production](#) (including \*.onmicrosoft.com accounts): smtp.office365.com (Find this setting via outlook.office.com, at **Settings > Mail > POP and IMAP**.)
  - For Outlook/Hotmail: smtp-mail.outlook.com
4. Set the user name and password to an appropriate email account and password.
5. Leave **SSLRequired** turned on, and leave **SMTP port number** set to **587**.
6. Click **Save**.
7. In Internet Explorer, navigate to **Accounts receivable > Customers > All customers**.
8. Select **US-008 Sparrow Retail**.
9. Click **Collect > Customer balances > Collections** to open the **Collections** page.
10. Click **Communicate > Email > Statements to contact**.
11. Click **OK** to accept the default values in the dialog box.
12. If you're prompted for the mail option to use, select **Use the Microsoft Dynamics 365 for Finance and Operations email client**, and then click **OK**.
13. To receive the test message, change the **To** address to your email address.
14. Enter a subject and body for the message.
15. Click **Send**. The message should be delivered in one to five minutes. Note that the message will appear to be sent from the email account that is set on the **Email parameters** page. If that email account is given "Send As" (or "Send email from this mailbox") permissions for the From address that is used in the **Send email** dialog box, messages will appear to come from that address.
  - You can configure "Send As" permissions in the Microsoft 365 admin center (portal.office.com/Admin), at **Users > Active users > User > Edit mailbox permissions > Send email from this mailbox**. For more information, see [Enable sending email from another user's mailbox in Microsoft 365](#).
  - Before users can send email messages, "Send As" permissions for each user email account in the client must be given to the email account that is set on the **Email parameters** page. For more information, see [How to set up a multifunction device or application to send email using Microsoft 365](#).
16. Email that is sent directly from the server, without user interaction, is sent via a batch process and requires that the **Email distributor batch** process be started. Follow these steps to start the process:
  - a. Navigate to **System administration > Periodic tasks > Email processing > Batch**.
  - b. Turn on **Batch processing**.

# Additional resources

[Office integration overview](#)

[Troubleshoot the Office integration](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# View and update entity data with Excel

2/18/2021 • 9 minutes to read • [Edit Online](#)

## Applies to these Dynamics 365 apps:

Commerce, Finance, Supply Chain Management

### IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [Service update availability](#).

This topic explains how to open entity data in Microsoft Excel, and then view, update, and edit the data by using the Microsoft Dynamics Excel add-in. To open entity data, you can start from either Excel or Finance and Operations apps.

By opening entity data in Excel, you can quickly and easily view and edit the data by using the Excel add-in. This add-in requires Microsoft Excel 2016 or later.

### NOTE

If your Microsoft Azure Active Directory (Azure AD) tenant is configured to use Active Directory Federation Services (AD FS), you must make sure that the May 2016 update for Office has been applied, so that the Excel add-in can correctly sign you in.

To learn more about how to use the Excel add-in, watch the short [Create an Excel template for header and line patterns](#) video.

## Open entity data in Excel when you start from a Finance and Operations app

1. On a page in a Finance and Operations app, select **Open in Microsoft Office**.

If the root data source (table) for the page is the same as the root data source for any entities, default **Open in Excel** options are generated for the page. **Open in Excel** options can be found on frequently used pages, such as **All vendors** and **All customers**.

2. Select an **Open in Excel** option, and open the workbook that is generated. This workbook has binding information for the entity, a pointer to your environment, and a pointer to the Excel add-in.
3. In Excel, select **Enable editing** to enable the Excel add-in to run. The Excel add-in runs in a pane on the right side of the Excel window.
4. If you're running the Excel add-in for the first time, select **Trust this Add-in**.
5. If you're prompted to sign in, select **Sign in**, and then sign in by using the same credentials that you used to sign in to the Finance and Operations app. The Excel add-in will use a previous sign-in context from the browser and automatically sign you in, if it can. (For information about the browser that is used based on the operating system, see [Browsers used by Office add-ins](#)) To ensure that sign-in was successful, verify the user name in the upper-right corner of the Excel add-in.

The Excel add-in automatically reads the data for the entity that you selected. Note that there will be no data in

the workbook until the Excel add-in reads it in.

## Open entity data in Excel when you start from Excel

1. In Excel, on the **Insert** tab, in the **Add-ins** group, select **Store** to open the Office Store.
2. In the Office Store, search on the keyword **Dynamics**, and then select **Add** next to **Microsoft Dynamics Office Add-in** (the Excel add-in).
3. If you're running the Excel add-in for the first time, select **Trust this Add-in** to enable the Excel add-in to run. The Excel add-in runs in a pane on the right side of the Excel window.
4. Select **Add server information** to open the **Options** pane.
5. In your browser, copy the URL of your target Finance and Operations app instance, paste it into the **Server URL** field, and then delete everything after the host name. The resulting URL should have only the host name.

For example, if the URL is `https://xxx.dynamics.com/?cmp=usmf&mi=CustTableListPage`, delete everything except `https://xxx.dynamics.com`.

6. Select **OK**, and then select **Yes** to confirm the change. The Excel add-in is restarted and loads metadata.  
  
The **Design** button is now available. If the Excel add-in has a **Load applets** button, you probably aren't signed in as the correct user. For more information, see "The Load applets button is shown" in the [Troubleshooting](#) section of this topic.
7. Select **Design**. The Excel add-in retrieves entity metadata.
8. Select **Add table**. A list of entities appears. The entities are listed in "Name - Label" format.
9. Select an entity in the list, such as **Customer - Customers**, and then select **Next**.
10. To add a field from the **Available fields** list to the **Selected fields** list, select the field, and then select **Add**. Alternatively, double-click the field in the **Available fields** list.
11. After you've finished adding fields to the **Selected fields** list, make sure that the cursor is in the correct place in the worksheet (for example, cell A1), and then select **Done**. Then select **Done** to exit the designer.
12. Select **Refresh** to pull in a set of data.

## View and update entity data in Excel

After the Excel add-in reads entity data into the workbook, you can update the data at any time by selecting **Refresh** in the Excel add-in.

## Edit entity data in Excel

You can change entity data as you require and then publish it back to Finance and Operations apps by selecting **Publish** in the Excel add-in. To edit a record, select a cell in the worksheet, and then change the cell value. To add a new record, follow one of these steps:

- Click anywhere in the data sources table, and then select **New** in the Excel add-in.
- Click anywhere in the last row of the data sources table, and then press the Tab key until the cursor moves out of the last column of that row and a new row is created.
- Click anywhere in the row immediately below the data sources table, and start to enter data in a cell. When you move the focus out of that cell, the table expands to include the new row.
- For field bindings of header records, select one of the fields, and then select **New** in the Excel add-in.

Note that a new record can be created only if all the key and mandatory fields are bound in the worksheet, or if default values were filled in by using the filter condition.

To delete a record, follow one of these steps:

- Right-click the row number next to the worksheet row that should be deleted, and then select **Delete**.
- Right-click anywhere in the worksheet row that should be deleted, and then select **Delete > Table Rows**.

If data sources have been added as related data sources, the header is published before the lines. If there are dependencies between other data sources, you might have to change the default publishing order. To change the publishing order, in the Excel add-in, select the **Options** button (the gear symbol), and then, on the **Data Connector** FastTab, select **Configure publish order**.

## Add or remove columns

You can use the designer to adjust the columns that are automatically added to the worksheet.

### NOTE

If the **Design** button doesn't appear below the **Filter** button in the Excel add-in, you must enable the data source designer. Select the **Options** button (the gear symbol), and then select the **Enable design** check box.

1. In the Excel add-in, select **Design**. All the data sources are listed.
2. Next to the data source, select the **Edit** button (the pencil symbol).
3. In the **Selected fields** list, adjust the list of fields as you require:
  - To add a field from the **Available fields** list to the **Selected fields** list, select the field, and then select **Add**. Alternatively, double-click the field in the **Available fields** list.
  - To remove a field from the **Selected fields** list, select the field, and then select **Remove**. Alternatively, double-click the field.
  - To change the order of fields in the **Selected fields** list, select a field, and then select **Up** or **Down**.
4. To apply your changes to the data source, select **Update**. Then select **Done** to exit the designer.
5. If you added a field (column), select **Refresh** to pull in an updated set of data.

## Change the publish batch size

When users publish changes to data records by using the Excel add-in, the updates are submitted in batches. The default publish batch size is 100 rows. In version 10.0.17 and later, the **Allow configuration of the publish batch size in the Excel add-in** feature gives you flexible control over the publish batch size.

System administrators can specify a system-wide limit on the publish batch size for "Open in Excel" workbooks by setting the **Publish batch limit** field in the **App parameters** section of the **Office app parameters** page.

The publish batch size can also be changed for an individual workbook by using the Excel add-in.

1. Open the workbook in Excel.
2. Select the **Option** (gear) button in the upper right of the Excel add-in.
3. Set the **Publish batch size** field as desired. The value that you set must be less than the system-wide publish batch limit.
4. Select **OK**.
5. Save the workbook. If you don't save the workbook after you make changes to the add-in settings, those changes won't persist when the workbook is reopened.

Excel workbook template authors can use the same procedure to set the publish batch size for templates before they upload them into the system.

## Copy environment data

The data that is read into the workbook from one environment can be copied to another environment. However, you can't just change the connection URL, because the data cache in the workbook will continue to treat the data as existing data. Instead, you must use the Copy Environment Data functionality to publish the data to a new environment as new data.

1. Select the **Options** button (the gear symbol), and then, on the **Data Connector** FastTab, select **Copy Environment Data**.
2. Enter the server URL for the new environment.
3. Select **OK**, and then select **Yes** to confirm the action. The Excel add-in is restarted and connects to the new environment. Any existing data in the workbook is treated as new data.

After the Excel add-in is restarted, a message box states that the workbook is in Environment copy mode.

4. To copy the data into the new environment as new data, select **Publish**. To cancel the environment copy operation and review the existing data in the new environment, select **Refresh**.

## Troubleshooting

There are a few issues that can be resolved through some easy steps.

- **The Load applets button is shown** – If the Excel add-in has a **Load applets** button after sign-in, you probably aren't signed in as the correct user. To resolve this issue, verify that the correct user name appears in the upper-right corner of the Excel add-in. If an incorrect user name appears, select it, sign out, and then sign back in.
- **You receive a "Forbidden" message** – If you receive a "Forbidden" message while the Excel add-in is loading metadata, the account that is signed in to the Excel add-in doesn't have permission to use the targeted service, instance, or database. To resolve this issue, verify that the correct user name appears in the upper-right corner of the Excel add-in. If an incorrect user name appears, select it, sign out, and then sign back in.
- **A blank webpage is shown over Excel** – If a blank webpage is opened during the sign-in process, the account requires AD FS, but the version of Excel that is running the Excel add-in isn't recent enough to load the sign-in dialog box. To resolve this issue, update the version of Excel that you're using. To update the version of Excel when you're in an enterprise that is on the deferred channel, use the [Office deployment tool to move from the deferred channel to the current channel](#).
- **You receive a time-out while you publish data changes** – If you receive time-out messages while you're trying to publish data changes to an entity, consider reducing the publish batch size for the affected workbook. Entities that trigger larger amounts of logic on record changes might require updates to be sent in smaller batches to help prevent time-outs.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create Open in Excel experiences

2/18/2021 • 18 minutes to read • [Edit Online](#)

Applies to these Dynamics 365 apps:

Commerce, Finance, Human Resources, Supply Chain Management

Learn about creating Open in Office experiences for Excel and Word.

## What are Open in Excel experiences?

Open in Excel experiences are:

- Based on entities and the OData services that they create.
- Dynamically-generated or based on a pre-defined template.
- Editable and refreshable via the Excel Add-in.

The following image shows the **Excel Add-in** being used for Journal entry.

The screenshot displays an Excel spreadsheet titled "LedgerJournalLineEntryTemplate\_635702638412497567(1) [Read-Only] - Excel". The spreadsheet is titled "General Ledger Journal Entry" and contains a table of journal entries. The table has columns for Date, Account type, MainAccount, BusinessUnit, Department, Description, Currency, Debit, and Credit. The data shows various benefits and payroll tax entries for January 2011. A summary row at the bottom shows Totals with a Debit of 3,575,931.49 and a Credit of 710,016.87. On the right side of the spreadsheet, there is a Microsoft Dynamics Data Connector panel with options for Refresh, Publish, Filter, and Design.

Date	Account type	MainAccount	BusinessUnit	Department	Description	Currency	Debit	Credit
1/20/2011	Ledger	602180	002	026	Benefits	USD	2,808.00	\$ -
1/20/2011	Ledger	602180	002	025	Benefits	USD	312.00	\$ -
1/20/2011	Ledger	602180	002	024	Benefits	USD	4,992.00	\$ -
1/20/2011	Ledger	602180	002	023	Benefits	USD	19,656.00	\$ -
1/20/2011	Ledger	602180	002	022	Benefits	USD	3,432.00	\$ -
1/20/2011	Ledger	602180	001	026	Benefits	USD	4,212.00	\$ -
1/20/2011	Ledger	602180	001	025	Benefits	USD	468.00	\$ -
1/20/2011	Ledger	602180	001	024	Benefits	USD	7,488.00	\$ -
1/20/2011	Ledger	602180	001	023	Benefits	USD	29,484.00	\$ -
1/20/2011	Ledger	602180	001	022	Benefits	USD	5,148.00	\$ -
1/20/2011	Ledger	602120	002	026	Payroll Tax	USD	2,246.40	\$ -
1/20/2011	Ledger	602120	002	025	Payroll Tax	USD	249.60	\$ -
1/20/2011	Ledger	602120	002	024	Payroll Tax	USD	3,993.60	\$ -
1/20/2011	Ledger	602120	002	023	Payroll Tax	USD	15,724.80	\$ -

## Where are the Open in Excel experiences?

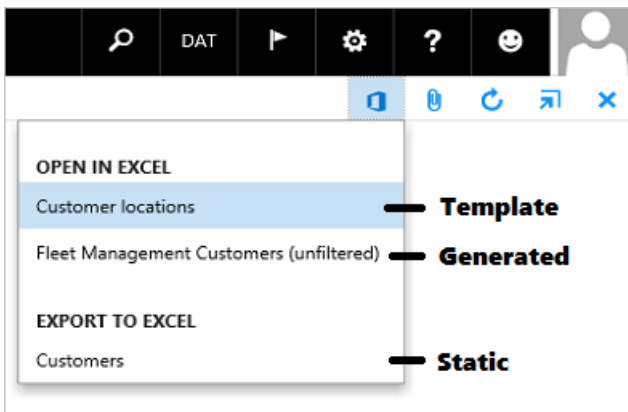
Open in Excel experiences are usually found under in the Open in Excel section of the Open in Microsoft Office menu, but an explicit button can be added for these experiences.

## What's the difference between Export to Excel and Open in Excel?

The Export to Excel options and experiences are both found in the Open in Microsoft Office menu:

- The Export to Excel options are static exports of grid data. Each one corresponds to a visible grid. All the grid data for the current filter is placed into a workbook.
- The Open in Excel experiences utilize the Excel Add-in to facilitate refresh and publish.

The following image shows the **Open in Microsoft Office** menu on the **Fleet Customers** form with a template **Open in Excel** option, a generated **Open in Excel** option, and a static **Export to Excel** option.

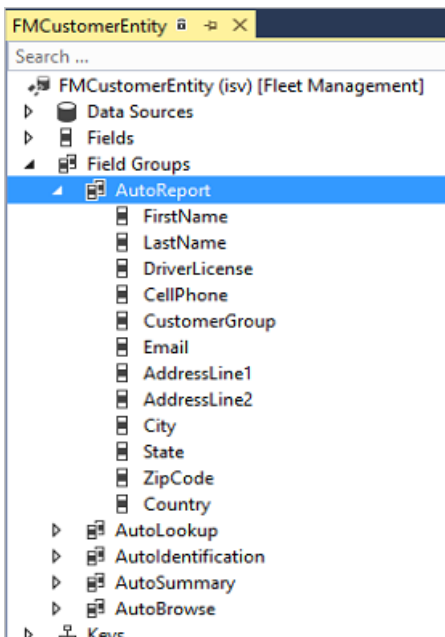


## When will an entity show as an Open in Excel option?

When an entity has the same root datasource (table) as a form, it will be added as an option in the Open in Excel section of the Open in Microsoft Office menu. This is referred to as a "generated" option.

## What fields will be shown in the workbook?

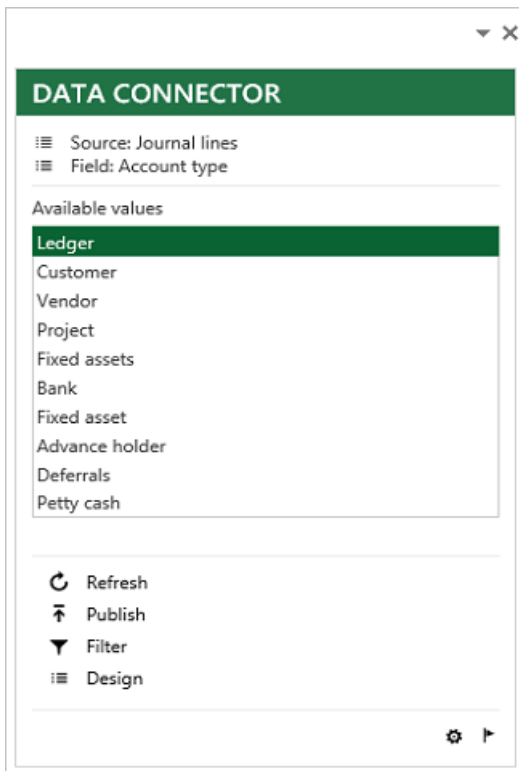
The default fields that will be added into the workbook are the key and mandatory fields of the entity. If a different set of fields should be provided by default, then those fields can be added into the **AutoReport field group** on the entity. The following image shows the Visual Studio view of the AutoReport field group for the FMCustomerEntity.



## What fields will be shown when an entity is the target of a lookup?

When a relationship is defined between two entities, if the identifier for one entity is shown on the other entity, then the fields that will be shown in that lookup are either the key fields, or the fields in the **AutoLookup field group** if it is not empty. Relationship lookups are not currently supported, but they will eventually be displayed in the app in a similar way to the enumeration lookups. The Excel Add-in with an enumeration lookup is shown below.





## What should be done to make an entity ready for use in Excel?

Define the AutoReport and AutoLookup field groups and test them using the Excel App design experience.

## Why does an automatically added entity option have “(unfiltered)” after the entity name?

Currently, a filter is not added to these options, hence the term “(unfiltered)”. In the future, an attempt will be made to apply the filter from the form to these options. For example, if a list of Customers was filtered to just Customers in the state of California, then, in the future, the entity will be scanned for the state field and if it is found then a filter would be added automatically.

## How can an entity be added as an Open in Excel option on a form that doesn't share the same root datasource?

A generated Open in Excel option can be added on any form by implementing the `OfficeGeneratedWorkbookCustomExporter` interface. When adding a generated option programmatically, the set of fields can be explicitly specified. For more information, see [Modifying the Open in Office menu through interfaces](#).

## What are the region-specific considerations for defining entities?

The Open in Excel generated experiences can be made region-specific by adding region-specific fields into the AutoLookup group. These region-specific fields will then be included in the generated workbook.

## How can I create a custom lookup for an entity field in Excel?

A custom lookup can be shown for an Entity field.

- Name - The method needs to have a name that is “lookup\_<fieldname>” e.g. a field “MyField” could have a lookup method “lookup\_MyField”.
- Attributes – Attributes need to be added to the method:

- SysODataActionAttribute(str <name>, Boolean <isInstanceMethod>)
- SysODataCollectionAttribute(str <name>, Types <type>, "Value")
- Return – The method should return a list of strings.

## Example

```
public class ExportToExcel_SimpleEntity extends common
{
    [SysODataActionAttribute("Lookup_StringLookupField", true),
    SysODataCollectionAttribute("return", Types::String, "Value")]
    public List lookup_StringLookupField()
    {
        List lookupList = new List(Types::String);
        const int items = 5;

        for (int item = 0; item < items; item++)
        {
            lookupList.addEnd(strfmt('%1 - %2 (%3)', this.StringField, this.IntField, item));
        }

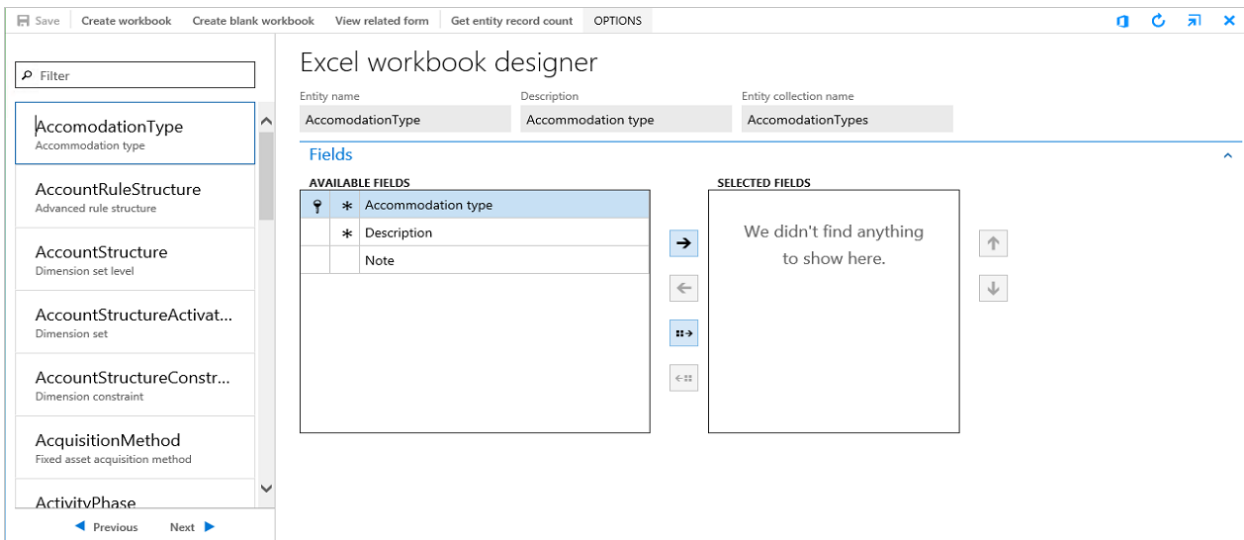
        return lookupList;
    }
}
```

## How does the app get injected into a workbook to start building a template?

The Excel Add-in is injected into a workbook when a generated Open in Excel experience is triggered or when a workbook is created using the **Common > Common > Office integration > Excel workbook designer** form.

- The **Create workbook** button will add the selected entity and fields, a pointer to the server, and the app into a workbook.
- The **Create blank workbook** button will simply add a pointer to the server and the app into a workbook.
- The **View related** form will navigate to the form relating to the currently selected entity to more easily review the effect of data changes made in Excel.
- The **Get entity record count** button will show the record count for the currently selected entity. The Excel Add-in will handle large sets of data within the memory limits of a user's machine. By default, the Excel Add-in has a data governor that restricts the data size to one million cells but, depending on the performance abilities of the user's machine, this can usually be extended to around 2.5 million cells.

The following image shows the **Excel workbook designer** form.



After obtaining a workbook containing the Excel Add-in, additional datasources can be added using the **Design** button. Currently, datasources cannot be removed.

## When will a template show as an Open in Excel option?

When a template listed in the **Common > Common > Office integration > Document templates** form (DocuTemplate) has `ShowInOpenInOfficeMenu` set to `Yes` and has the same root datasource (table) as the current form, it will be added as an option in the **Open in Excel** section of the **Open in Microsoft Office** menu. The following image shows the **Document templates** form.

TEMPLATE TYPE	TEMPLATE NAME	COMPANY	LANGUAGE	ROOT DATA ENTITY	TEMPLATE DISPLAY NAME	DESCRIPTION
Excel	FMTemplateCustomersWithLocati			FMCustomerEntity	Customer locations	Template provides a map of custo
Excel	FMTemplateRentalsByStatus			FMRentalEntity	Rentals by status	Template provides rental status cc
Excel	LedgerJournalLineEntryTemplate			LedgerJournalHeaderEntity	LedgerJournalLineEntryTemplate	General journal line entry using an
Word	DocuGenerationTestStringKeyTem			ExportToExcel_StringKeyEntity	DocuGenerationTestStringKeyTem	This is a test description

## Will a filter be added to the template?

In the **Document Templates** form, the standard filter for "current record" can be turned on and off. If the filter is on, when the template is invoked as an **Open in Excel** option, then a filter for the current record will be added to the workbook. The filter will be the key fields and their values.

## How can templates be defined in metadata and code and loaded automatically?

When adding a template into the **Document Templates** form, it is added for that instance and is referred to as a "user-defined" template. Templates can also be defined in metadata and code and loaded automatically, thus making them "system-defined" templates. To create a system-defined template using metadata and code, you need to do the following:

- Define the template.
- Create a new resource in a project.
- Define a new class that extends the `DocuTemplateRegistrationBase` class and add an implementation of the `registerTemplates` method.

The `LedgerJournalLineEntryTemplateRegistration` and `FMTemplateRegistrations` classes are good examples of

template registrations defined in code. The `LedgerJournalLineEntryTemplate` and `FMTemplateCustomersWithLocations` resources are the corresponding templates stored in metadata as resources. When a template has a registration class, it will be loaded when the **Reload system templates** button is clicked in the **Document Templates** form.

## How do templates get loaded into a fresh deployment?

To load system defined templates, click the **Reload system templates** button in the **Common > Common > Office integration > Document templates** form, as shown below.



In the future, we will do the equivalent of clicking that button during deployment.

## How do I decide if I should create a template?

A template is an artifact that needs to be maintained and versioned. If you can avoid defining a template without sacrificing much from the user experience, then you probably should use a template. Create a template if:

- You need additional content or formatting in the template.
- You want to combine multiple entities/datasources in the same workbook.

Don't create a template if:

- You can just specify a set of fields to show in a table binding.

## What are the region-specific considerations for templates?

When creating a template for an entity that has region-specific fields, you should leave those region-specific fields out of the template, otherwise all users will see the region-specific fields. Templates should cater to the majority of users by default and region-specific users can add those fields using the easy-to-use design experience of the Excel Add-in. The region-specific fields and columns can be added by users as needed. That template can be either saved to local computer for reuse by a single user or uploaded via the Document Templates form for reuse by any user of that instance. A couple of other considerations:

- If a region has a region-specific entity, then a region-specific template could be created.
- If a region is important enough, then you could define a region-specific template as well as a region-generic template.

## How do I add an explicit button for a template Open in Excel option?

An explicit button can be added for Open in Excel experiences. The label shown on the button should usually be "Open target in Excel" where target is the name of the target data like "lines" or "catalog". The code behind such a button will:

- Obtain the template to be used.
- Add a filter.
- Pass the template to the user.

An example of this code can be found on the `LedgerJournalTable` form (**General ledger > Journals > General journal**) in the `Clicked` method on the `OpenLinesInExcel` button.

```

[Control("Button")]
class OpenLinesInExcel
{
    /// <summary>
    /// Opens the current journal in Excel for line entry and editing
    /// </summary>
    public void clicked()
    {
        super();

        const str templateName = resourceStr(LedgerJournalLineEntryTemplate);
        DocuTemplate template = DocuTemplate::findTemplate(OfficeAppApplicationType::Excel,
templateName);

        // Ensure the template was present
        if (template && template.TemplateID == templateName)
        {
            Map filtersToApply = new Map(Types::String, Types::String);

            // Create lines filter
            ExportToExcelFilterBuilder filterBuilder = new
ExportToExcelFilterBuilder(tablestr(LedgerJournalLineEntity));
            str filterString = filterBuilder.areEqual(fieldstr(LedgerJournalLineEntity, JournalBatchNumber),
LedgerJournalTable.JournalNum);
            filtersToApply.insert(tablestr(LedgerJournalLineEntity), filterString);

            // Create header filter
            filterBuilder = new ExportToExcelFilterBuilder(tablestr(LedgerJournalHeaderEntity));
            filterString = filterBuilder.areEqual(fieldstr(LedgerJournalHeaderEntity, JournalBatchNumber),
LedgerJournalTable.JournalNum);
            filtersToApply.insert(tablestr(LedgerJournalHeaderEntity), filterString);

            // Generate the workbook using the template and filters
            DocuTemplateRender renderer = new DocuTemplateRender();
            str documentUrl = renderer.renderTemplateToStorage(template, filtersToApply);

            // Pass the workbook to the user
            if (documentUrl)
            {
                Browser b = new Browser();
                b.navigate(documentUrl, false, false);
            }
            else
            {
                error(strFmt("@ApplicationFoundation:DocuTemplateGenerationFailed", templateName));
            }
        }
        else
        {
            warning(strFmt("@ApplicationFoundation:DocuTemplateNotFound", templateName));
        }
    }
}
}

```

The following image shows the **General ledger > Journals > General journal** form with the **Open lines in Excel** button highlighted.

General journal

Show: All  Show user-created only

LIST GENERAL SETUP BLOCKING FINANCIAL DIMENSIONS HISTORY

✓	NAME	JOURNAL BAT... ↑	DESCRIPTION	POSTED	LOG	I...	REVERSING ENTRY	REVERSING DATE
	GenJrn	00001	Payroll - Jan	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	
	GenJrn	00002	Payroll - Feb	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	
	GenJrn	00342	Payroll - Mar	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	
	GenJrn	00343	Payroll - Apr	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	
	GenJrn	00344	Payroll - May	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	
	GenJrn	00345	Payroll - Jun	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	
	GenJrn	00346	Payroll - Jly	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	
	GenJrn	00347	Payroll - Aug	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	

◀ Previous      Next ▶

To programmatically add generated and template Open in Excel options, Open in Excel options can be added by implementing the `ExportToExcelGeneratedCustomExport` and `ExportToExcelTemplateCustomExport` interfaces. This allows the addition of options to forms where the entity or template doesn't have the same table as the root datasource. An example of when you would use this capability is on forms without a datasource, potentially containing only a collection of form parts. The following example adds generated and template Open in Excel options programmatically to the **FM Rental** form.

```

[Form]
public class FMRental extends FormRun implements ExportToExcelGeneratedCustomExport,
ExportToExcelITemplateCustomExport
{
...

    public List getExportOptions()
    {
        List exportOptions = new List(Types::Class);

        ExportToExcelExportOption exportOption =
ExportToExcelExportOption::construct(ExportToExcelExportType::CustomGenerated, int2str(1));
        exportOption.setDisplayNameWithDataEntity(tablestr(FMRentalEntity));
        exportOptions.addEnd(exportOption);

        ExportToExcelExportOption exportOption2 =
ExportToExcelExportOption::construct(ExportToExcelExportType::CustomTemplate, int2str(2));
        exportOption2.displayName("Analyze rentals");
        exportOptions.addEnd(exportOption2);

        return exportOptions;
    }

    public ExportToExcelDataEntityContext getDataEntityContext(ExportToExcelExportOption _exportOption)
    {
        ExportToExcelDataEntityContext context = null;

        if (_exportOption.id() == int2str(1))
        {
            context = ExportToExcelDataEntityContext::construct(tablestr(FMRentalEntity),
tablefieldgroupstr(FMRentalEntity, AutoReport));
        }

        return context;
    }

    public System.IO.Stream getTemplate(ExportToExcelExportOption _exportOption)
    {
        System.IO.Stream stream = null;

        if (_exportOption.id() == int2str(2))
        {
            stream =
Microsoft.Dynamics.Ax.Xpp.MetadataSupport::GetResourceContentStream(resourcestr(FMRentalEditableExportTempla
te));
        }

        return stream;
    }

    public void updateTemplateSettings(ExportToExcelExportOption _exportOption,
Microsoft.Dynamics.Platform.Integration.Office.ExportToExcelHelper.SettingsEditor _settingsEditor)
    {
    }
...

```

## How do I add a filter for a programmatically-added template Open in Excel option?

A template Open in Excel option can be programmatically added by implementing the `ExportToExcelITemplateCustomExport` interface and providing a template in the `getTemplate` method. A filter for that option can be programmatically added by using the `ExportToExcelFilterBuilder` API in the `updateTemplateSettings` method.

```

public void updateTemplateSettings(ExportToExcelExportOption _exportOption,
Microsoft.Dynamics.Platform.Integration.Office.ExportToExcelHelper.SettingsEditor _settingsEditor)

{

_settingsEditor.SetFilterExpression(tableStr(RetailTmpBulkProductAttributeValueEntity),
element.getExportToExcelFilterExpression());

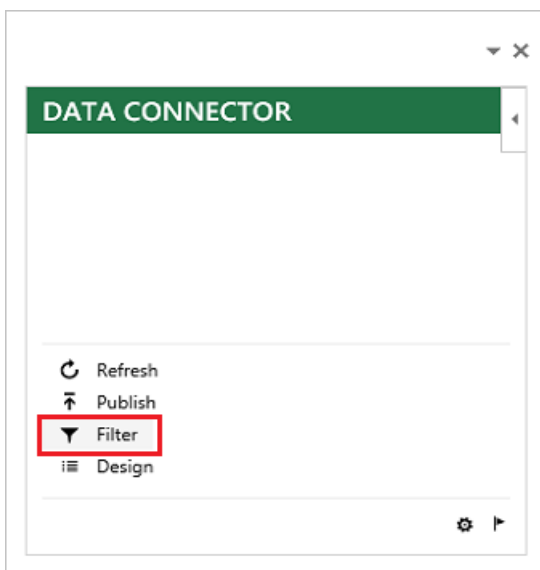
DictDataEntity dictDataEntity = new DictDataEntity(tableNum(RetailTmpBulkProductAttributeValueEntity));

_settingsEditor.SetFilterExpressionByPublicName(dictDataEntity.publicEntityName(),
element.getExportToExcelFilterExpression());

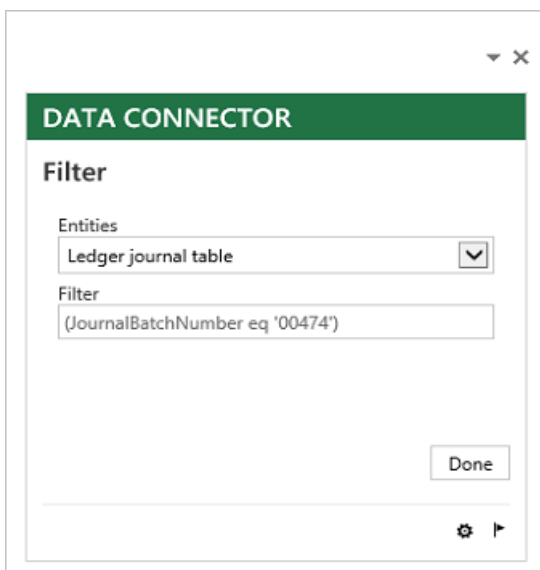
}

```

After a filter has been added programmatically, the resulting filter can be viewed in the Excel Add-in using the **Filter** button. The following image shows the Excel Add-in with the **Filter** button highlighted.



The following image shows the Excel Add-in with the **Filter** dialog box opened.



## How do I enable relationship lookups in Excel?

To enable relationship lookups in the Excel Data Connector, you must ensure that the following metadata is set.

- The Role and Related Data Entity Role defined on the relationship need to be unique among all



relationships on both the source and target entity. Also, the relation role properties must be unique across all entities. This is particularly important for relationships involving entities with many relationships, such as DimensionCombinationEntity. If you're not seeing an expected lookup, try changing the role names to the following format:

- **Role:** [this entity's public name] + [target entity's public name] + [target entity field] + "Source"
- **Related Data Entity Role:** [this entity's public name] + [target entity's public name] + [target entity field] + "Target"
- The Cardinality and Related Data Entity Cardinality need to be set appropriately.
- At least one constraint must be added to the relationship. With the exception of dimension relationships, which are a special case, the properties constrained must both be public.

## How can I enable users to create new header records as well as lines in a workbook?

To enable creation of header records and related lines, the header data source must be added as a set of "fields" and the lines data source must be added as a related table. This pattern can work well for document data entry scenarios such as Journal entry.

To learn more about header records and related lines, watch the short [Create an Excel template for header and line patterns in Dynamics 365 for Finance and Operations](#) video.

To design a workbook with header fields and a lines table that enables header creation:

1. In the Excel Add-in, click **Design** to open the Designer. Select **Add fields** to add a header data source.
2. Select the header fields that you want to use. Be sure to include all the key fields or the **New** button won't be enabled.
3. For all of the string header value fields, manually apply "Text" format for that cell using **Excel ribbon > Home tab > Number group > set "Number"** in the format drop-down menu. If the Text format isn't manually set on a string field and there's a string value with leading zeros like "00045", then Excel will automatically change it to "45" and an error will be shown like: *"Unable to change the value of PurchaseOrderHeader's PurchaseOrderNumber field as it is a key field"*. Currently, the API doesn't allow for automatically applying the text formatting on individual cells (versus table columns).
4. In the Designer, on the header data source, click the **Add related table** button represented by a double plus icon.
5. Select the line fields that you want to use.

Here's an example of a header data source with a related table data source.

- PurchaseOrderHeader (Fields)
  - dataAreald
  - PurchaseOrderNumber
  - PurchaseOrderName
  - OrderVendorAccountNumber
- PurchaseOrderLine (Table - related)
  - LineNumber
  - ItemNumber
  - LineDescription
  - OrderedPurchaseQuantity
  - LineAmount

To use a header and lines workbook to create a new header and lines, follow these steps:

1. In the workbook, move the focus to a cell with a header value.
2. In the Excel Add-in, click **New**.
3. Enter header values and lines as needed.
4. Select **Publish**.

## How can fields be added, removed, or moved within an existing template workbook?

Fields can be added into an existing template workbook by editing the workbook stored in **Document Templates**.

1. Get the original template workbook.
  - a. Open the **Document Templates** form.
  - b. Find the existing template workbook.
  - c. Download the workbook.
  - d. Open the workbook and enable editing so that the Excel Add-in runs.
2. Make changes to the template.
  - a. In the Excel Add-in, select **Design**.
  - b. Click the **Edit** button (pencil icon) next to the datasource that you want to add a field into.
  - c. Add fields by moving them from the **Available fields** list into the **Selected fields** list. Double-clicking a field will move it. Remove fields by moving them from the **Selected fields** list into the **Available fields** list. Move fields using the **Up** and **Down** buttons.
  - d. After changes are complete, select **Update**, select **Yes** to confirm, and then select **Done** to exit the Designer (if appropriate, select **Refresh** to verify that the data is correctly populated).
  - e. Clear the data from the template before upload by clicking **Options** (gear icon), expand the **Data Connector** section, then click the **Clear binding data** button.
  - f. Use **Save As** to store the template somewhere temporarily.
3. Upload the changed template.
  - a. Return to the **Document Templates** form and upload the changed template.
  - b. Click **New** and browse to find the changed template.
  - c. Select the saved template file and click **Open**.
  - d. In the **Upload template** dialog box, remove the underscore and trailing random number from the name. For example, "CustInvoiceJournalTemplate\_636564840743000567" becomes "CustInvoiceJournalTemplate".
  - e. A confirmation dialog box should show that "A template with this name already exists...", click **Yes** to confirm replacement of the previous template. Note that if this confirmation is not shown, then the template name is different and it is being uploaded as a new template.
4. Open the form that the template is used on and use the changed template.

## Troubleshooting

If you are not seeing an expected lookup, validate relationship metadata by checking the metadata feed available at [YourSiteURL]/data/\$metadata. Search the \$metadat feed for the public name of your entity to find its EntityType element, then make sure there is a child NavigationProperty element with a name equal to the Role value of the relationship. If the navigation property exists, it will be used by the Excel Data Connector to show a relationship lookup. Lookups are not shown under the following conditions:

- All of the entity's key fields are included as constraints in the relationship.
- The selected field is a key and the selected record is not new.

- The authenticated user does not have permission to access the entity targeted by the lookup.

## How do dimensions work?

The easiest way to set up dimension metadata on data entities is to use the data entity creation wizard, which will automatically create the private relationships and public display value fields exactly as the dimensions framework needs them. If you want to customize your dimensions setup, see [Add dimensions to Excel templates](#). Lookups, are only generated automatically for non-ledger dimensions. Custom dimensions are not supported currently. If you want to enable lookups for ledger dimensions (MainAccount, Department, CostCenter, etc.), see [Add dimensions to Excel templates](#) for guidance on creating relationships on DimensionCombationEntity and DimensionSetEntity fields. When those relationships are present, relationship lookups will be displayed in the Excel Data Connector. The Excel Data Connector supports two types of dimension data entry: editing the display value directly or editing each attribute of the display value in a separate column. If both the display value column and the individual attribute columns are bound, they can both be edited and published separately. If both the display value and an individual attribute are edited in the same row, the individual attribute change overrides the display value change.

## How do I create formula table columns?

If a formula is needed in a table, then add a formula column. When in the field selection page for a table binding, click the **Formula** button above the Selected fields list to add a new formula column. The label and value for the formula are entered in the fields immediately below the Selected fields list. After adding a new formula column, leave the value empty and click **Update**. After the field has been added to the table, use standard Excel capabilities to create a formula, then copy the formula and paste it into the formula column value field. When defining a formula, make sure there is more than one row in the table, otherwise the formula that Excel provides may be for ALL rows instead of THAT row. To specify just the current row, the at sign (@) is needed. For example, sum of four columns for all rows `"=SUM(Table1[[ColumnA]:[ColumnD]])"` versus sum of four columns for the current row `"=SUM(Table1[@[ColumnA]:[ColumnD]])"`.

## Known issues

### Refresh doesn't automatically occur in old templates

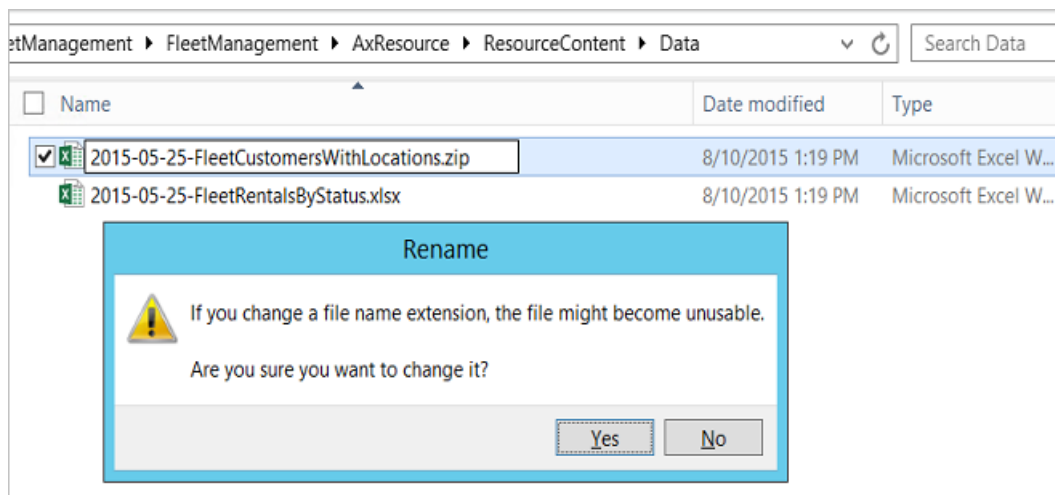
The ability to control "refresh on open" was added as a setting. To add this to the default behavior, existing templates and workbooks need to have the **Refresh on open** check box selected in **Options > Data Connector > Refresh Options**.

### Error finding entity

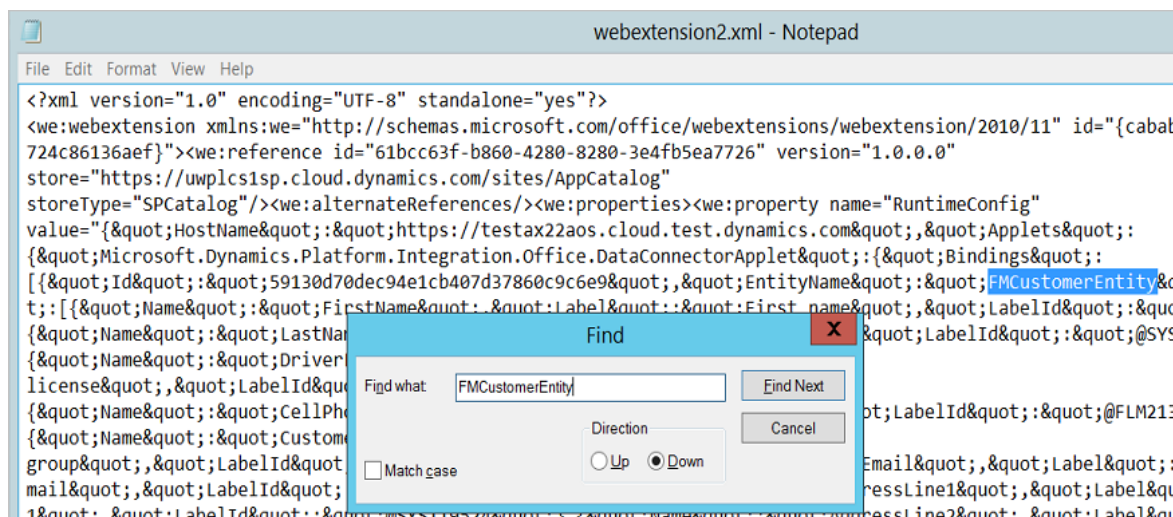
The reference to entities changed from using the Private Entity Name (DataEntity.Name) to Public Entity Name (DataEntity.PublicEntityName). If the public and private names for an entity were different and that entity was used in an Excel template or workbook, then this will cause the following error to be displayed in the Excel App: "Error Finding Entity. Details: Entity "<DataEntity.Name>" not found".

To resolve this, change the binding information in the affected template so that it points to DataEntity.PublicEntityName instead of DataEntity.Name.

1. For the DataEntity.Name that needs to be replaced, determine the DataEntity.PublicEntityName, for example replace FMCustomerEntity with FleetCustomer.
2. Find the affected template.
3. Change the file extension on the template from .xlsx to .zip.



4. The file to be changed will be one of the webextension\*.xml files in the xlwebextensions directory, such as 2015-05-25-FleetCustomersWithLocations.zipxlwebextensionswebextension2.xml.
5. Open the file to ensure that you have the correct location.
6. Find the DataEntity.Name, such as FMCustomerEntity.



7. Extract the zip file.
8. Open the webextension xml file.
9. Replace the DataEntity.Name with the corresponding DataEntity.PublicEntityName.
10. Save the webextension .xml file changes.
11. Rename the old zip file, for example, add ".old" to the name.
12. Create a new zip file of all the previously extracted content. This usually involves highlighting the content inside the archive/zip folder and creating a zipped folder using that content.
13. Verify that the zip file has the "\_rels", "docProps", and "xl" folders in the root of the zip file.
14. Rename the zip file as needed, for example rename the file 2015-05-25-FleetCustomersWithLocations.zip.
15. Change the zip file extension to .xlsx.
16. Re-publish the workbook .xlsx file, if needed.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

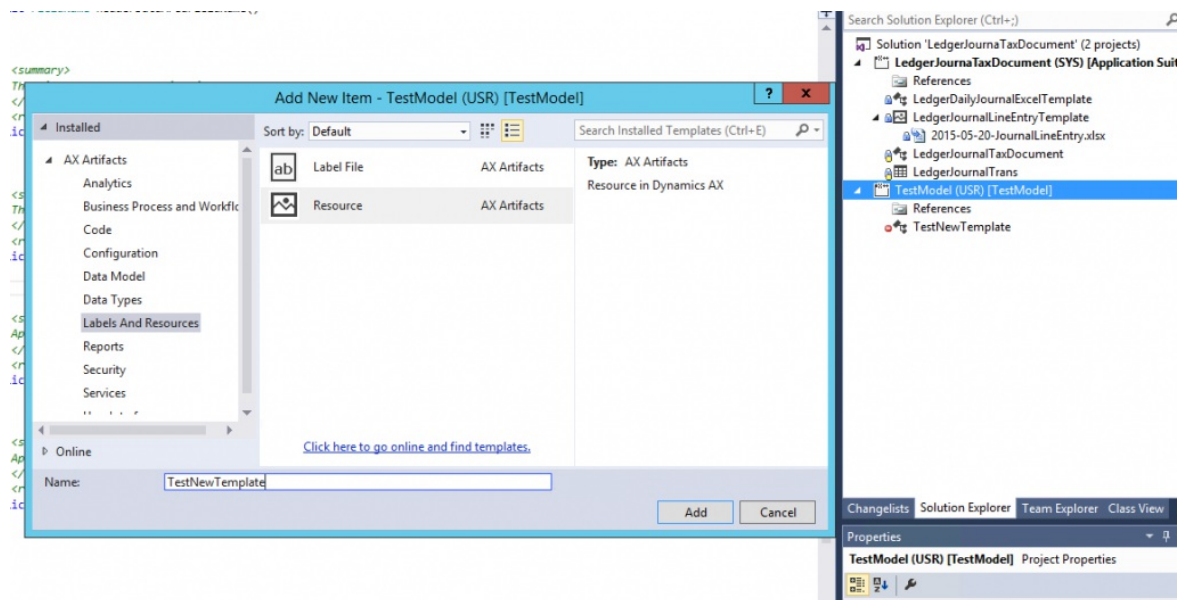
# Add templates to the Open lines in Excel menu

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how you can promote a template to the Open lines in the Excel menu that is available on journal pages.

Some of the most frequently used templates are the journal templates. Some of these journal templates have been promoted so that they appear on the **Open lines in Excel** menu by default. However, when you add a new template to the system, it's available on the **Open in Office** menu by default. To promote the template so that it's available on the **Open lines in Excel** menu, follow these steps.

1. Create a Microsoft Excel template, and save it locally. For more information, see the "Create Open in Excel experiences" article.
2. In Microsoft Visual Studio, create a new project for a model that has a reference to the ApplicationSuite model.



3. Create a new class, implement the **LedgerJournalExcelTemplate** interface, and extend **DocuTemplateRegistrationBase**. Your implementation (supported journal type, and so on) defines the context that your template will be available as an option for in the Open in Excel experience. This example uses **LedgerJournalHeaderEntity** and **LedgerJournalLineEntity**, but you aren't limited to these entities. You can define your own entities, provided that they entities follow the journal header/line entity pattern. Here is an example from the **LedgerDailyJournalExcelTemplate** class.

```
using Microsoft.Dynamics.Platform.Integration.Office;
public class TestNewTemplate extends DocuTemplateRegistrationBase implements
LedgerJournalExcelTemplate
{
    private const DocuTemplateName ExcelTemplateName = resourceStr(TestNewTemplate);
    private const EntityName LineEntityName = tableStr(LedgerJournalLineEntity);
    private const FieldName LineEntityJournalNum = fieldStr(LedgerJournalLineEntity,
JournalBatchNumber);
    private const FieldName LineEntityDataAreaId = fieldStr(LedgerJournalLineEntity, dataAreaId);
    private const FieldName HeaderEntityName = tableStr(LedgerJournalHeaderEntity);
    private const FieldName HeaderEntityJournalNum = fieldStr(LedgerJournalHeaderEntity,
JournalBatchNumber);
    private const FieldName HeaderEntityDataAreaId = fieldStr(LedgerJournalHeaderEntity, dataAreaId);
    /// <summary>
```

```

/// A boolean value which indicates whether the journal type is supported for the Excel template.
/// </summary>
/// <param name = "_ledgerJournalType">The ledger journal type.</param>
/// <returns>True if the journal type is supported; otherwise, false.</returns>
public boolean isJournalTypeSupported(LedgerJournalType _ledgerJournalType)
{
    return _ledgerJournalType == LedgerJournalType::Daily;
}
/// <summary>
/// Gets the document template name.
/// </summary>
/// <returns>The document template name</returns>
public DocuTemplateName documentTemplateName()
{
    return ExcelTemplateName;
}
/// <summary>
/// Gets a collection of the supported account types for the entity.
/// </summary>
/// <returns>A collection of <c>LedgerJournalACType</c> values.</returns>
public Set supportedAccountTypes()
{
    Set accountTypeSet = new Set(Types::Integer);
    accountTypeSet.add(LedgerJournalACType::Ledger);
    return accountTypeSet;
}
/// <summary>
/// Gets a collection of the supported offset account types for the entity.
/// </summary>
/// <returns>A collection of <c>LedgerJournalACType</c> values.</returns>
public Set supportedOffsetAccountTypes()
{
    Set offsetAccountTypeSet = new Set(Types::Integer);
    offsetAccountTypeSet.add(LedgerJournalACType::Ledger);
    return offsetAccountTypeSet;
}
/// <summary>
/// Validates the journal is valid for the template.
/// </summary>
/// <param name = "_ledgerJournalTable">The <c>LedgerJournalTable</c> record.</param>
/// <returns>True if the journal is valid for the template; otherwise, false.</returns>
public boolean validateJournalForTemplate(LedgerJournalTable _ledgerJournalTable)
{
    return LedgerJournalExcelTemplate::validateJournalForTemplate(_ledgerJournalTable, this);
}
public void registerTemplates()
{
    this.addTemplate(
        OfficeAppApplicationType::Excel,
        ExcelTemplateName,
        ExcelTemplateName,
        'Test new template',
        'Test new template',
        NoYes::No,
        NoYes::No,
        NoYes::No);
}
/// <summary>
/// The resource name of the header entity.
/// </summary>
/// <returns>The resource name of the header entity.</returns>
public EntityName headerEntityName()
{
    return HeaderEntityName;
}
/// <summary>
/// The resource name of the line entity.
/// </summary>
/// <returns>The resource name of the line entity.</returns>

```

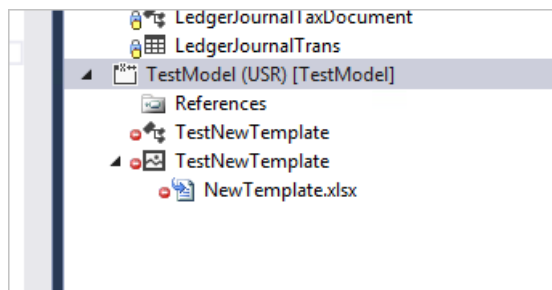
```

public EntityName lineEntityName()
{
    return LineEntityName;
}
/// <summary>
/// The field name for the header journal batch number.
/// </summary>
/// <returns>The field name for the header journal batch number.</returns>
public FieldName headerJournalBatchNumberFieldName()
{
    return HeaderEntityJournalNum;
}
/// <summary>
/// The field name for the header data area.
/// </summary>
/// <returns>The field name for the header data area.</returns>
public FieldName headerDataAreaFieldName()
{
    return HeaderEntityDataAreaId;
}
/// <summary>
/// The field name for the line journal batch number.
/// </summary>
/// <returns>The field name for the line journal batch number.</returns>
public FieldName lineJournalBatchNumberFieldName()
{
    return LineEntityJournalNum;
}
/// <summary>
/// The field name for the line data area.
/// </summary>
/// <returns>The field name for the line data area.</returns>
public FieldName lineDataAreaFieldName()
{
    return LineEntityDataAreaId;
}
/// <summary>
/// Append additional filter to the default filtering behavior.
/// </summary>
/// <returns>The original filter with new filter(s) appended; Otherwise, the original
filter</returns>
public FilterCollectionNode appendHeaderEntityFilters(FilterCollectionNode _headerFilter,
ExportToExcelFilterTreeBuilder _headerFilterBuilder)
{
    return _headerFilter;
}
/// <summary>
/// Append additional filter to the default filtering behavior.
/// </summary>
/// <returns>The original filter with new filter(s) appended; Otherwise, the original
filter</returns>
public FilterCollectionNode appendLineEntityFilters(FilterCollectionNode _lineFilter,
ExportToExcelFilterTreeBuilder _lineFilterBuilder)
{
    FilterCollectionNode lineFilter = _lineFilterBuilder.and(
        _lineFilterBuilder.areEqual(fieldStr(LedgerJournalLineEntity, AccountType),
LedgerJournalACType::Ledger),
        _lineFilterBuilder.areEqual(fieldStr(LedgerJournalLineEntity, OffsetAccountType),
LedgerJournalACType::Ledger));
    return _lineFilterBuilder.and(_lineFilter, lineFilter);
}
}

```

4. Build the project/model that has the new resources. You should have one new resource and one new class.





5. In the client, go to **Common > Common > Office integration > Document templates > Reload system templates**. You will see the new template in the list, and if you open the journal page that you added the template to, you will also see that template on the **Open lines in Excel** menu.

## Additional resources

[Create Open in Excel experiences](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Customize the Open in Microsoft Office menu

2/18/2021 • 10 minutes to read • [Edit Online](#)

## Applies to these Dynamics 365 apps:

Commerce, Finance, Supply Chain Management

Most pages include an Open in Microsoft Office menu. This topic provides information about the Open in Office menu, and explains how to customize it by adding, removing, and changing options.

## Overview

The **Open in Microsoft Office** menu button (**Open in Office** menu) is a system-defined button that appears on pages. The **Open in Office** menu contains menu items that let you export data to various Office products, such as Microsoft Excel and Microsoft Word. The following table describes the menu items on the **Open in Office** menu.

MENU ITEM	DESCRIPTION
Export to Excel	The data is exported to an Excel workbook. The workbook contains no references back to Finance and Operations, and the data can't be refreshed.
Export to Word	The data is exported to a Word document. The document contains no references back to Finance and Operations, and the data can't be refreshed.
Open in Excel	A workbook is created that contains the Microsoft Dynamics Office add-in. The workbook contains a reference back to Finance and Operations, and the data can be refreshed, updated, and published from the Data Connector that is hosted in the add-in.

## How menu items are added to the Open in Office menu

The export options are added to the **Open in Office** menu in the following manner:

1. In the **Export to Excel** group, a menu item is added for each visible grid.
2. For each root data source on the page, the set of data entities that have the same root data source is determined. For each of these data entities, the following menu items are added:
  - In the **Open in Excel** group, a menu item is added for a default export of the data entity.
  - In the **Open in Excel** group, a menu item is added for each Document Template record of the **Excel** type that has the same root data entity and is marked for inclusion on the **Open in Office** menu.
  - In the **Export to Word** group, a menu item is added for each Document Template record of the **Word** type that has the same root data entity and is marked for inclusion on the **Open in Office** menu.

## Default exports

The **Open in Office** menu provides a default export for each data entity. This export includes all the fields in the **AutoReport** group on the data entity. If **SaveDataPerCompany** is set to **Yes** for the data entity, a filter is applied to limit the data to the current company.

# Document templates

Document templates can be added from the **Document templates** page. Several fields that are associated with each Document Template record control the behavior of that template on the **Open in Office** menu.

FIELD	DESCRIPTION
Root data entity	The root data entity of the template. The root data entity is used to determine which pages the template can be included on.
List in Office menu	If this field is selected, the template will be included on the <b>Open in Office</b> menu on applicable pages. (The applicable pages depend on the root data entity).
Apply record filter	If this field is selected, the data will be filtered, based on the record that is currently selected on the page.
Apply company filter	If this field is selected, the data will be filtered, based on the current company.

## Trimmed template columns and fields

For Excel templates that are included on the **Open in Office** menu, columns and fields will be trimmed from the workbook, based on the configuration keys that are applied to the system and the applicable country/region context. If a configuration key is associated with a column or field in the workbook, the column or field will be removed if the configuration key is disabled. If a set of country/region codes is associated with the column or field in the workbook, the column or field will be removed if the country/region code isn't in scope.

## Open in Office menu customization

There are several methods for customizing the content that appears on the **Open in Office** menu on a given page. For example, you can customize the content statically through metadata properties on model element and code attributes. However, customization via code gives you the finest level of control. In code, you can either implement one or more interfaces on a page, or use extensions and event subscriptions. The following sections describe the customization scenarios that are most often used.

### Modifying the Open in Office menu through interfaces

If you must modify a page that you own, interfaces are the most appropriate customization method, because they give access to all private members of the page and allow for deeper customization. You can apply the following interfaces to the code for a page.

INTERFACE	DESCRIPTION
OfficeMenuCustomizer	Use this interface to modify the set of data entities that is considered for a page and add custom menu items.
OfficeGeneratedWorkbookCustomExporter	Use this interface to do a custom export of a workbook that is generated at run time.
OfficeTemplateCustomExporter	Use this interface to do a custom export that is based on a Document Template record.

### Modifying the Open in Office menu through extensions and event subscriptions

If you must modify a page that you don't own, you should avoid using interfaces, because that approach will

require over-layering. Instead, you should do the customization through extensions and event subscriptions. To use this approach, implement an extension class that subscribes to the **OnInitializing** event of the page that you're customizing. From this event handler, get the **OfficeFormRunHelper** for the page, and subscribe to its **OfficeMenuInitializing** event. The following example shows sample code for this approach.

```
public static class MyForm_Extension
{
    [FormEventHandler(formStr(MyForm), FormEventType::Initializing)]
    public static void ExportToExcel_DataEntityCustom_OnInitializing(xFormRun sender, FormEventArgs e)
    {
        FormRun formRun = sender as FormRun;
        if (formRun)
        {
            OfficeFormRunHelper officeHelper = formRun.officeHelper();
            if (officeHelper)
            {
                officeHelper.OfficeMenuInitializing +=
                eventhandler(MyForm_Extension::officeMenuInitializingHandler);
            }
        }
    }
    private static void officeMenuInitializingHandler(FormRun _formRun, OfficeMenuEventArgs _eventArgs)
    {
        // Modify the OfficeMenuOptions available on the OfficeMenuEventArgs.menuOptions() as necessary.
    }
}
```

## Typical customization scenarios

The following examples assume that the **\_menuOptions** variable contains the **OfficeMenuOptions** instance that you're customizing.

### Modifying the set of data entities that is considered for a page

Many of the menu items on the **Open in Office** menu are added automatically, based on the data entities that are considered for the page. However, in some cases, the algorithm that is used to determine the set of data entities might not determine the correct set. To modify the set of data entities that is considered for the page, you can use the **OfficeMenuOptions** that is available from either the **OfficeMenuCustomizer.customizeMenuOptions** method or the **OfficeFormRunHelper.OfficeMenuInitializing** delegate.

```
// Add an entity to the list
OfficeMenuDataEntityOptions entityOptions = OfficeMenuDataEntityOptions::construct(tableStr(MyEntity));
_menuOptions.dataEntityOptions().addEnd(entityOptions);
// Remove an entity from the list
ListIterator dataEntityOptionsIterator = new ListIterator(_menuOptions.dataEntityOptions());
while (dataEntityOptionsIterator.more())
{
    OfficeMenuDataEntityOptions dataEntityOptions = dataEntityOptionsIterator.value();
    if (dataEntityOptions.dataEntityName() == tableStr(MyOtherEntity))
    {
        dataEntityOptionsIterator.delete();
    }
    else
    {
        dataEntityOptionsIterator.next();
    }
}
```

### Specifying the default data entity–related options that are included

The **OfficeMenuDataEntityOptions** class lets you specify whether to include a menu item for a default export or a menu item that is related to a document template.

```
// Find the entity options if they were included by default.
OfficeMenuDataEntityOptions entityOptions = _menuOptions.getOptionsForEntity(tableStr(MyEntity));
if (!entityOptions)
{
    // The entity options were not included. Add them.
    entityOptions = OfficeMenuDataEntityOptions::construct(tableStr(MyEntity);
    _menuOptions.dataEntityOptions().addEnd(entityOptions);
}
entityOptions.includeDefault(false); // Don't include the default export menu item.
entityOptions.includeTemplates(false); // Don't include Document Template related menu items.
```

### Adding a custom export menu item – Generating a workbook

To explicitly add a menu item, you must add it to the **OfficeMenuOptions.customMenuItems()** list. To add a menu item that corresponds to a workbook that is generated at run time, use an

**OfficeGeneratedExportMenuItem**.

```
OfficeGeneratedExportMenuItem menuItem = OfficeGeneratedExportMenuItem::construct(tableStr(MyEntity),
"MyCustomGeneratedExportId");
menuItem.setDisplayNameWithDataEntity();
_menuOptions.customMenuItems().addEnd(menuItem);
```

To define what is actually exported, use an **ExportToExcelDataEntityContext**. The method for specifying the **ExportToExcelDataEntityContext** depends on whether you're using interfaces or extensions and event subscriptions to customize the **Open in Office** menu.

#### Using interfaces

If you're using interfaces, you must implement the

**OfficeGeneratedWorkbookCustomExporter.getDataEntityContext()** method.

```
public ExportToExcelDataEntityContext getDataEntityContext(OfficeGeneratedExportMenuItem _menuItem)
{
    ExportToExcelDataEntityContext context = null;
    if (_menuItem.id() == "MyCustomGeneratedExportId")
    {
        context = ExportToExcelDataEntityContext::construct(tableStr(MyEntity), tableFieldGroupStr(MyEntity,
MyFieldGroup));
    }
    return context;
}
```

#### Using extensions and event subscriptions

If you're using extensions and event subscriptions, the

**OfficeGeneratedExportMenuItem.getDataEntityContext** delegate should be subscribed to before you add the menu item to the **OfficeMenuOptions.customMenuItems()** list. The code for the event handler should resemble the preceding code for the interface. The following example shows how to do the event subscription.

```
menuItem.getDataEntityContext += eventhandler(MyForm_Extension::getDataEntityContextHandler);
```

### Adding a custom export menu item – Specifying a document template

To explicitly add a menu item, you must add it to the **OfficeMenuOptions.customMenuItems()** list. To add a menu item that corresponds to a Document Template record, use an **OfficeTemplateExportMenuItem**.

```
OfficeTemplateExportMenuItem menuItem =
OfficeTemplateExportMenuItem::construct(OfficeAppApplicationType::Excel, "MyTemplateId",
"MyCustomTemplateExportId");
_menuOptions.customMenuItems().addEnd(menuItem);
```

To modify the template at run time, you can supply a set of initial filters. These filters will replace any filters in the template for the specified data entities. Additionally, you can modify filters and specify many settings by using **WorkbookSettingsManager**. The following sections show examples.

#### Using interfaces

If you're using interfaces, you must implement the

**OfficeTemplateCustomExporter.getInitialTemplateFilters()** and **OfficeTemplateCustomExporter.updateTemplateSettings()** methods.

```
public Map getInitialTemplateFilters(OfficeTemplateExportMenuItem _menuItem)
{
    Map initialFilters = new Map(Types::String, Types::Class);
    if (_menuItem.id() == "MyCustomTemplateExportId")
    {
        // Add an initial filter.
        ExportToExcelFilterTreeBuilder bldr = new ExportToExcelFilterTreeBuilder(_menuItem.dataEntityName());
        FilterNode filter = // create the filter..
        initialFilters.insert(_menuItem.dataEntityName(), filter);
    }
    return initialFilters;
}
public void updateTemplateSettings(OfficeTemplateExportMenuItem _menuItem,
Microsoft.Dynamics.Platform.Integration.Office.SettingsManager _settingsManager)
{
    if (_menuItem.id() == "MyCustomTemplateExportId")
    {
        // Set a new filter.
        ExportToExcelFilterTreeBuilder bldr = new ExportToExcelFilterTreeBuilder(_menuItem.dataEntityName());
        FilterNode filter = // create the filter..
        Excel.WorkbookSettingsManager workbookSettingsManager = _settingsManager as Excel.WorkbookSettingsManager;
        workbookSettingsManager.SetEntityFilter(entityMetadata.PublicEntityName, filter);
        // Adjust settings.
        DataConnectorAppletSettings settings = settingsManager.DataConnectorSettings;
        DataConnectorAppletUserOptions options = settings.DataOptions;
        options.RefreshOnOpen = true;
        options.EnableDesign = false;
        workbookSettingsManager.DataConnectorSettings = settings;
    }
}
```

#### Using extensions and event subscriptions

If you're using extensions and event subscriptions, the

**OfficeTemplateExportMenuItem.getInitialTemplateFilters** and **OfficeTemplateExportMenuItem.updateTemplateSettings** delegates should be subscribed to before you add the menu item to the **OfficeMenuOptions.customMenuItems()** list. The code for the event handlers should resemble the preceding code for the interface. The following example shows how to do the event subscription.

```
menuItem.getInitialTemplateFilters += eventhandler(MyForm_Extension::getInitialTemplateFiltersHandler);
menuItem.updateTemplateSettings += eventhandler(MyForm_Extension::updateTemplateSettingsHandler);
```

## Additional customizations

The following customizations let you modify the contents of the **Open in Office** menu for a page without using

interfaces or extensions and event handlers.

### Removing an Export to Excel menu item for a grid

On the **Open in Office** menu, the **Export to Excel** group will contain a menu item for each visible grid. To remove a grid from the **Open in Office** menu, set the **ExportAllowed** metadata property on the grid to **No**. After you make this change, users won't be able to export the grid by using the shortcut menu for the grid.

### Renaming an Export to Excel menu item for a grid

To rename the **Export to Excel** menu item that is related to a grid, set the **ExportLabel** metadata property on the grid to the appropriate label.

### Removing all menu items for a data entity from all pages

Integration scenarios require that some data entities be publicly available via the OData Service. However, it isn't always appropriate that these data entities appear on the **Open in Office** menu by default. In this scenario, you can add the **OfficeMenuOmit** code attribute to the entity declaration.

```
[OfficeMenuOmit]
public class MyEntity extends common
{
    // Entity code...
}
```

After you make this change, by default, the entity won't appear on the **Open in Office** menu on pages that have a matching root data source. However, if the entity should be added to a specific page, you can use other customization mechanisms to add it.

### Adding a menu item button to a page that corresponds to an Open in Office menu entry

Sometimes, it's appropriate that the Action Pane of a page have a menu item button that corresponds to a custom menu item on the **Open in Office** menu. In this case, you can model a menu item button that has the following properties:

- **Menu Item Type:** Action
- **Menu Item Name:** ExportToExcelExportForm
- **Parameters:** The ID of the menu item

Then, a mouse click of this menu item button is equivalent to a mouse click of the corresponding menu item on the **Open in Office** menu.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and send email

2/18/2021 • 13 minutes to read • [Edit Online](#)

The behavior of the email subsystem is influenced by a combination of administrator configuration, user configuration, and user choices. This topic is divided into sections for administrators and users to make it easy to find relevant information.

Both administrators and users set the behavior of the email subsystem.

## [Administrator] Email parameters page

### Configuration tab

On the **Email parameters** page, note the following settings on the **Configuration** tab.

FIELD	DESCRIPTION
Batch email provider	Specifies which email provider will be used to send emails that are sent by processes in a batch or non-interactive manner. The Exchange provider will use the account associated with the batch process.
Attachment size limit	Specifies the maximum size of a single email that can be sent via the email subsystem.

In Platform update 32, an **Email history** page was added to allow administrators to review all sent emails, including any errors that might have prevented an email from being sent. By default, the last 30 days of email history is retained. This can be configured by changing the **Number of days to retain email history** to a non-zero amount. Zero provides the default amount and behavior.

In version 10.0.16/Platform 40, an **Email throttling** section is visible, if your environment has enabled the **Email throttling** feature in Feature management. This feature allows non-interactive email providers (such as the batch email provider) to adhere to a per-minute sending limit. This can prevent errors from the system attempting to send more emails than the provider allows. The sending limits for Microsoft 365 email providers are set automatically according to [Exchange Online sending limits](#). Manual configuration is required for all other email providers. The per-minute sending limit can be removed from a provider by resetting the **per-minute email sending limit** field to 0.

### SMTP settings tab

On the **Email parameters** page, note the following settings on the **SMTP settings** tab.

FIELD	DESCRIPTION
Outgoing mail server	The host name of the desired SMTP server. <ul style="list-style-type: none"><li>For <a href="#">Microsoft 365 production</a> (including *.onmicrosoft.com accounts) use smtp.office365.com. (You can find this setting at outlook.office.com at <b>Settings &gt; Mail &gt; POP and IMAP</b>.)</li><li>For Outlook/Hotmail use smtp-mail.outlook.com.</li></ul>
SMTP port number	Typically, the port number should be set to 587 for secure transport.



FIELD	DESCRIPTION
User name and Password	Specify, as needed, to send the email via the appropriate mail account. All users need to provide the SMTP account <b>Send As</b> and <b>Send On Behalf Of</b> permissions to enable the ability to send Simple Mail Transfer Protocol (SMTP) mail. You can configure Send As permissions in the Microsoft 365 admin center (portal.office.com/Admin), at <b>Users &gt; Active users &gt; User &gt; Edit mailbox permissions &gt; Send email from this mailbox</b> . For more information, see <a href="#">Enable sending email from another user's mailbox in Microsoft 365</a> .
Specify if SSL is required	Determines whether secure transport is used. Typically, this is <b>Yes</b> , except for internal or troubleshooting scenarios.

## [Administrator] Email distributor batch process

Email that is sent directly from the server, without user interaction, via SMTP is sent by the **Email distributor batch** process. That batch process must be started to process the email queue. To start the process, open the **Email distributor batch** pane (**System administration > Periodic tasks > Email processing > Batch**) and turn on **Batch processing**.

If the Exchange provider is used, then the user account associated with the batch process (usually admin) will be sender.

## [Administrator] User email

The default **send from** address for each user is pulled from the **Email** field on the **Users** page (**System administration > Users > Users**). Administrators can override this **send from** default if needed using the **Sender email** field on the **Options** page.

## [User] Email provider selection section on the Options page

The **Options** page can be opened via **Settings > User options**. The **Email provider selection** section is on the **Account** tab.

FIELD	DESCRIPTION
Email provider ID	Allows the user to select the email provider that should be used when sending an email. Selecting an option here is the equivalent of selecting <b>Do not ask again</b> in the <b>How would you like to send email</b> dialog box. Selecting the blank option <b>Prompt for which email provider to use</b> will cause the <b>How would you like to send email</b> dialog box to display when an email is going to be sent.

FIELD	DESCRIPTION
Sender email	<p>Allows the administrator to provide an email address override for the user in the <b>From</b> field of the email. By default, the email alias that is associated with the user account is used as the <b>From</b> field in new emails, but this user option email address will override that. When sending email via SMTP, the user needs to have appropriate <b>Send As</b> and <b>Send On Behalf Of</b> permissions configured in Exchange or on the SMTP server.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>[!NOTE] You can configure <b>Send As</b> and <b>Send On Behalf Of</b> permissions in the Microsoft 365 admin center (portal.office.com/Admin) at <b>Users &gt; Active users &gt; User &gt; Edit mailbox permissions &gt; Send email from this mailbox</b>. For more information, see <a href="#">Enable sending email from another user's mailbox in Microsoft 365</a>.</p> </div>

## [User] How would you like to send email dialog box (optional)

When an email is going to be sent, the user will see the **How would you like to send email** dialog box that will list the available options for sending email.

FIELD	DESCRIPTION
Use an email app, such as Outlook	Provides the user with a generated email (.eml) file.
Use Exchange email server	Uses the Exchange Online server associated with the tenant. On-premises Exchange servers are currently not supported for the <b>Exchange</b> mail provider.
Use the system email client	Opens the <b>Send email</b> composition dialog box and then sends the resulting email via SMTP.
Do not ask again	If this field is not selected, the next time an email is sent the most recently selected option will be used and the dialog box will not open.

## [User] Send email dialog box (optional)

The **Send email** dialog box is opened to allow the user to edit the contents of the email that will be sent. Some of the following fields will be pre-populated in this window.

FIELD	DESCRIPTION
From	Populated from the <b>Email</b> field on the <b>Options</b> page.
To, Cc, Bcc, Subject, and Body	Populated with values specified by the process that initiated the sending of the email. These fields can be edited as needed by the user.
Attachments list	May be populated with attachments specified by the process that initiated the sending of the email. This list can be edited as needed by the user.

When the email is ready to be sent, the **Send** button will cause the email to be sent via SMTP.

## Usage scenarios to verify if email is configured correctly

### Send mail via a local mail client

Email workflows that are enabled via the SysEmail framework can generate email messages (.eml files) that contain attachments. You can then send these messages via Microsoft Outlook or another email client.

1. In Internet Explorer, navigate to **Accounts receivable > Customers > All customers**.
2. Select **US-008 Sparrow Retail**.
3. Click **Collect > Customer balances > Collections** to open the **Collections** page.
4. Click **Communicate > Email > Statements to contact**.
5. Click **OK** to accept the default values in the dialog box.
6. If you're prompted for the mail option to use, clear the **Do not ask again** check box (you can change this option from the **User options** page), select **Use an email app, such as Outlook**, and then click **OK**.
7. If you're using Internet Explorer on your computer, open the email (.eml) file that is generated. If you're using Internet Explorer on the VM, copy the file to your computer, and open it there.
8. Note the email address in the **To** field and the generated workbook attachment.

### Send mail via SMTP

Email workflows that are enabled via the SysEmail framework can also be created in a simple email dialog box and then sent via Simple Mail Transfer Protocol (SMTP).

1. Go to the **Email parameters** page.
2. Click **SMTP settings**.
3. Set the **Outgoing mail server** to the desired SMTP server:
  - For **Microsoft 365 production** (including \*.onmicrosoft.com accounts) use smtp.office365.com. (Find this setting via outlook.office.com, at **Settings > Mail > POP and IMAP**.)
  - For Outlook/Hotmail use smtp-mail.outlook.com.
4. Set the user name and password to an appropriate email account and password.
5. Leave **SSLRequired** turned on, and leave **SMTP port number** set to **587**.
6. Click **Save**.
7. In Internet Explorer, navigate to **Accounts receivable > Customers > All customers**.
8. Select **US-008 Sparrow Retail**.
9. Click **Collect > Customer balances > Collections** to open the **Collections** page.
10. Click **Communicate > Email > Statements to contact**.
11. Click **OK** to accept the default values in the dialog box.
12. If you're prompted for the mail option to use, select **Use the Microsoft Dynamics 365 for Finance and Operations email client**, and then click **OK**.
13. To receive the test message, change the **To address** to your email address.

Ensure that the account specified in the SMTP settings is able to **Send As** and **Send On Behalf Of** your email account. The easiest way to ensure this is to use your email account in the SMTP settings.

14. Enter a subject and body for the message.

15. Click **Send**. The message should be delivered in one to five minutes.

## [Administrator] Workflow email notifications

Workflow email configuration is a collection of related settings that work in conjunction.

### Workflow email notification setup

1. Verify email settings:
  - a. Go to **System administration > Setup > Email > Email parameters**.
  - b. Verify that SMTP is enabled.
  - c. Set the SMTP mail server settings.
2. Verify that the email batch process is running:
  - a. Go to **System administration > Periodic tasks > Email processing > Email distributor batch**.
  - b. Enable the **Batch processing** option.
  - c. Optionally, adjust the recurrence of the email process:
    - a. Select **No end date** to adjust all recurrences of the email batch process.
    - b. Adjust the count.
    - c. Adjust to run every minute if needed.
3. Verify workflow notification system email templates:
  - a. Go to **System administration > Setup > Email > System email templates** (for system-wide templates).
  - b. Verify that the **Sender email** field is set and valid.
4. Verify workflow notification organization email templates:
  - a. Go to **Organization administration > Setup > Organization email templates** (for organization-specific templates).
  - b. Verify that the **Sender email** field is set and valid.
5. Verify that the user can receive email notifications:
  - a. Go to **Settings > User options**.
  - b. Go to the **Account** tab.
    - a. Set the **Email provider ID** (for example, SMTP).
    - b. Optionally, set a **Sender email** override if the default **send from** address should not be used for the current user.
  - c. Navigate to the **Workflow** tab. Set the option to send notifications in email to **Yes**.
6. Verify that the workflow system will send email notifications:

For each workflow that should have a notification, open the workflow properties in the workflow editor.

  - a. Click **Basic settings**. Adjust the email template for the workflow notifications.
  - b. Click **Notifications**.
    - a. Enable the events for which a user should be notified.
    - b. Set the recipient of the workflow notification for each event notification that is enabled.
  - c. On a workflow approval element for which a user should be notified:

- a. Go to **Properties**.
- b. Enable the events for which a user should be notified.
- c. Set the recipient of the workflow notification for each event notification that is enabled.

### **Workflow email notification testing**

The testing for email notifications is to simply trigger the notification and then check for it.

1. Submit a workflow that has been set up for email notifications.
2. Check the workflow history to make sure that a workflow work item was assigned to the expected user.
3. Check the status of the pending email notification in **System administration > Periodic tasks > Email processing > Batch email sending status**.

If the email fails to send, make sure that the SMTP mail account can be opened.

4. Check for the email notification in the appropriate inbox.

## **Troubleshoot email**

There are a few standard steps that can help you troubleshoot the configuration of email settings.

1. Verify email settings:
  - a. Go to **System administration > Setup > Email > Email parameters**.
  - b. Verify that SMTP is enabled.
  - c. Verify the settings of the SMTP mail server.
  - d. Sign in to the SMTP account in a separate window to make sure that the account and password are correct.
  - e. Send a test email using **System administration > Setup > Email > Email parameters > Test email**.
2. Verify that the email batch process is running:
  - a. Go to **System administration > Periodic tasks > Email processing > Batch**.
  - b. Make sure that the **Batch processing** option is set to **Yes**.
  - c. Review the recurrence of the email process:
    - a. Select **No end date** to adjust all recurrences of the email batch process.
    - b. Adjust the count as you require.
3. To review the contents and status of batch emails, go to **System administration > Periodic tasks > Email processing > Batch email sending status**.
  - a. If you're using a release that is earlier than Platform update 28, personalize the form to add the email sender for easy review. To do this, right-click the grid header, select **Add columns**, select **Email**, and then click **Insert**. If the **Email** field isn't added into the grid, you can view the sender by selecting **Show message**, and then selecting the **Email** field.
  - b. Verify that emails are being sent from the correct account. If the account is incorrect, you need to adjust settings such as user options, system templates, or organization templates, as needed.
  - c. Verify that all email user accounts have been granted permission to **Send As** for the configured SMTP account (see step 4 for details).
4. In Platform update 32, an **Email history** page was added to allow administrators to review all sent emails, including any errors that might have prevented an email from being sent. The **Email history** page will show interactive as well as non-interactive/batch emails. For any emails that have an **Email status** of **Failed**, review the error message on the **Failure details** tab and determine if corrective

actions should be taken.

5. In the Microsoft 365 admin center, verify that all user mail accounts that will be used to send emails have **Send As** and **Send On Behalf Of** permissions for the configured SMTP account. For more information, see [Enable sending email from another user's mailbox in Microsoft 365](#).
6. Sign in to all user mailboxes to verify that they are valid and can be accessed using sign in.
7. Send a test email using **System administration > Setup > Email > Email parameters > Test email**.
8. If the SMTP settings were migrated from another environment, clear the password field and re-enter the password to ensure that the field encryption hasn't negatively affected the stored value.
9. If you continue to experience issues when email is sent via SMTP, enter the SMTP account information in a tool such as [SMTPer.net](#) to verify that the SMTP server and account are valid and working correctly.

## Troubleshoot the Exchange mail provider

The **Email parameters** page allows an administrator to select Exchange as an interactive email provider and as the Batch email provider. The Exchange mail provider will use the current user's Exchange Online account to send emails. When used as the Batch email provider, the batch account will be used. No additional configuration is needed. If troubleshooting is needed, ensure that the current user's account can be signed into and that emails can be sent from that account to the intended recipients.

### **Exchange mail provider not supported for external users**

Users that are external to the primary tenant will not have exchange accounts on that tenant, so the Exchange mail provider is not supported for external users.

## Other notes

The system communicates with Exchange or an SMTP server like a typical email client, so standard behavior and limits apply. For example, standard [Exchange Online receiving and sending limits](#) apply.

## Troubleshooting

### **Where do workflow email templates come from?**

The email templates will be sourced from either system email templates or organization email templates depending on whether the workflow is a system-level (not company specific) or organization-level (company specific) workflow.

## Additional resources

[Troubleshoot the Office integration](#)

[Office integration tutorial](#)

[Configure email functionality in Microsoft Dynamics AX \[AX 2012\]](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create email templates for transactional events

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes how to create, upload, and configure email templates for transactional events in Microsoft Dynamics 365 Commerce.

## Overview

Dynamics 365 Commerce provides an out-of-box solution for sending emails that alert customers about transactional events (for example, when an order is placed, an order is ready for pickup, or an order has been shipped). This topic describes the steps for creating, uploading, and configuring the email templates that are used to send transactional emails.

## Create an email template

Before you can map a specific transactional event to an email template, you must create the template.

To create an email template, follow these steps.

1. In Commerce headquarters, go to **Organization email templates**, which is under **Retail and Commerce > Headquarters setup > Organization email templates** or **Organization administration > Setup > Organization email templates**.
2. Select **New**.
3. Under **General**, set the following fields:
  - **Email ID** – The email ID is the unique identifier for a template, and it's the value that is shown when you select a template to map to an event.
  - **Email description** – You can use this optional field to provide a description of the template. The value that you enter appears only in Commerce headquarters.
  - **Sender name** – The name that you enter appears in the "from" field of most email clients.
  - **Sender email** – Enter the email address that should be used for emails that are sent by using this template.
  - **Default language code** – This field specifies the localized version of the email that is sent by default, if no language is provided by the channel that invokes this template.
4. Under **Email message content**, select **New**.
5. In the **Language** field, enter the language for the email template. You can add more languages and localized templates later.
6. In the **Subject** field, enter the email subject that should appear in the subject field of the email.
7. Select **Edit** to upload your email template.

## Create an email message body by using HTML

The message body of your email is composed in HTML. You can use any layout, styling, and branding that HTML and inline Cascading Style Sheets (CSS) allow for. You can also use images, if you host them on a publicly available web endpoint. To add an image, enter the image's URL in the `src` attribute of the HTML `<img>` tag.

#### NOTE

Email clients impose layout and style limitations that might require adjustments to the HTML and CSS that you use for the message body. We recommend that you familiarize yourself with the best practices for creating HTML that will be supported by the most popular email clients.

## Add placeholders to the email message body

Your email can contain placeholders that are replaced with customer-specific and transaction-specific values when the email is generated. Placeholders are always surrounded by percent signs (%) and are inserted directly into the HTML document.

Here is an example.

```
<p>  
  Hello %customername%,<br />  
  Order number %salesid%, can be picked up from the <b>%pickupstorename%</b> store.  
</p>
```

### Order placeholders (sales order level)

The following placeholders retrieve and show data that is defined at the sales order level (as opposed to the sales line level).

PLACEHOLDER NAME	PLACEHOLDER VALUE
customername	The name of the customer who placed the order.
salesid	The sales ID of the order.
deliveryaddress	The delivery address for shipped orders.
customeraddress	The address of the customer.
deliverydate	The delivery date.
shipdate	The ship date.
modeofdelivery	The delivery mode of the order.
charges	The total charges for the order.
tax	The total tax for the order.
total	The total amount for the order.
ordernetamount	The total amount for the order, minus the total tax.
discount	The total discount for the order.
storename	The name of the store where the order was placed.
storeaddress	The address of the store that placed the order.



PLACEHOLDER NAME	PLACEHOLDER VALUE
storeopenfrom	The opening time of the store that placed the order.
storeopento	The closing time of the store that placed the order.
pickupstorename	The name of the store where the order will be picked up.
pickupstoreaddress	The address of the store where the order will be picked up.
pickupopenstorefrom	The opening time of the store where the order will be picked up.
pickupopenstoreto	The closing time of the store where the order will be picked up.

### Order line placeholders (sales line level)

The following placeholders retrieve and show data for individual products (lines) in the sales order.

PLACEHOLDER NAME	PLACEHOLDER VALUE
productid	The product ID for the line.
lineproductname	The name of the product.
lineproductdescription	The description of the product.
linequantity	The number of units that were ordered for the line, plus the unit of measure (for example, <b>ea</b> , or <b>pair</b> ).
lineunit	The unit of measure for the line.
linequantity_withoutunit	The number of units that were ordered for the line, without the unit of measure.
linequantitypicked	When the <b>PickOrder</b> event is used, the number of units that were picked. Otherwise, <b>0</b> (zero).
linequantitypicked_withoutunit	When the <b>PickOrder</b> event is used, the number of units that were picked, without the unit of measure. Otherwise, <b>0</b> (zero).
linequantitypacked	When the <b>PackOrder</b> and <b>Order ready for pickup</b> events are used, the number of units that were packed. Otherwise, <b>0</b> (zero).
linequantitypacked_withoutuom	When the <b>PackOrder</b> and <b>Order ready for pickup</b> events are used, the number of units that were packed, without the unit of measure. Otherwise, <b>0</b> (zero).
linequantityshipped	Always <b>0</b> , except when specific events are used, as described in the next row.

PLACEHOLDER NAME	PLACEHOLDER VALUE
linequantityshipped_withoutuom	When the <b>ShipOrder</b> event is used, the number of units that were picked, without the unit of measure. Otherwise, 0 (zero).
lineprice	The price of a single unit.
linenetamount	The price of the line after the number of units and discount are applied.
linediscount	The discount for an individual unit.
lineshipdate	The ship date for the line.
linedeliverydate	The delivery date for the line.
linedeliverymode	The delivery mode for the line.
linedeliveryaddress	The delivery address for the line.
giftcardnumber	The gift card number, for products of the gift card type.
giftcardbalance	The gift card balance, for products of the gift card type.
giftcardmessage	The gift card message, for products of the gift card type.
giftcardpin	The personal identification number (PIN) of the gift card, for products of the gift card type. (This placeholder is specific to external gift cards.)
giftcardexpiration	The expiration date of the gift card, for products of the gift card type. (This placeholder is specific to external gift cards.)
giftcardrecipientname	The name of the gift card recipient, for products of the gift card type.
giftcardbuyername	The name of the gift card buyer, for products of the gift card type.

### Format of order line placeholders in the email message body

When you create the HTML for the individual order lines in the email message body, surround the repeating block of HTML and placeholders for the lines with the following placeholders that are put inside HTML comment tags.

```
<!--%tablebegin.salesline%-->

(Insert the repeating block of HTML and placeholders for individual lines here.)

<!--%tableend.salesline%-->
```

Here is an example.

```
<table>
  <tr>
    <td>Product name</td>
    <td>Quantity</td>
    <td>Price</td>
  </tr>
<!--%tablebegin.salesline-->
  <tr>
    <td>%lineproductname%</td>
    <td>%linequantity_withoutunit%</td>
    <td>%lineprice%</td>
  </tr>
<!--%tableend.salesline-->
</table>
```

## Create a template for emailed receipts

Receipts can be emailed to customers who make purchases at a retail point of sale (POS). In general, the steps for creating the emailed receipt template are the same as the steps for creating templates for other transactional events. However, the following changes are required:

- The email ID of the email template must be **emailRecpt**.
- The text of the receipt is inserted into the email by using the **%message%** placeholder. To ensure that the receipt body is correctly rendered, surround the **%message%** placeholder with HTML **<pre>** and **</pre>** tags.
- Line breaks in the HTML for the header and footer of the email are converted to HTML **<br />** tags so that the receipt body is rendered correctly. To eliminate unwanted vertical space in your receipt emails, remove line breaks from any place in the HTML where vertical space isn't required.

For more information about how to configure email receipts, see [Set up email receipts](#).

## Upload the email HTML

After you've created and tested the HTML for your message body, it must be uploaded to Commerce headquarters. Currently, email HTML can't be exported. Therefore, you must maintain the master copy of your HTML outside Commerce headquarters.

To upload a new or edited email template HTML, follow these steps.

1. In Commerce headquarters, go to **Retail and Commerce > Headquarters setup > Organization email templates**.
2. Select the row for the language that you want to add or replace HTML for. Alternatively, select **New** to create a row for a new language.
3. Select **Edit**.
4. In the dialog box that appears, select **Browse**. Browse to the HTML document that you want to upload, select it, and then select **Open**.
5. Select **Upload**.
6. After your email HTML appears in the preview window, select **OK**.
7. Make sure that the **Has body** check box is selected for the row.

If you've already configured Commerce headquarters to send email, your new or updated email will be sent to all customers who perform a transaction that invokes the event that is mapped to the template.

For more information about how to configure emails in Dynamics 365 Commerce, see [Configure and send email](#).

## Additional resources

[Set up an email notification profile](#)

[Configure and send email](#)

[Set up email receipts](#)

[Send email receipts from Modern POS](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Develop email experiences by using the SysMailer framework

2/18/2021 • 6 minutes to read • [Edit Online](#)

## Sending emails

The SysMailer framework is a new, extensible way to send email. It replaces all previous application programming interfaces (APIs) for mail, such as CDO.Messaging (SysMailer), MAPI (SysINetMail), and Outlook COM (SmmOutlookEmail). Those older mail APIs won't work correctly in Finance and Operations applications. By taking advantage of the SysPlugin framework and several .NET technologies, SysMailer provides a configurable experience for users and enables the application consumers to remain agnostic to the email option that users use to send email.

The SysMailer framework consists of a factory class that is used to retrieve an email provider, a set of email providers that send messages, a message builder that builds the messages, and the forms that are related to configuring and interacting with the email providers. To consume the SysMailer framework, an application developer primarily uses the **SysMailerFactory** and **SysMailerMessageBuilder** classes. The email provider factory is used to retrieve interactive or non-interactive email providers, so that multiple messages can be sent at the same time, or so that a message can be sent directly. The email providers expect the messages that they send to be encapsulated in .NET **System.Net.Mail.MailMessage** objects. The message builder class is used to build the .NET object that is passed to the email provider.

### Scenarios

This topic describes three scenarios:

- Sending an interactive message
- Sending a non-interactive (batch) message
- Sending multiple non-interactive (batch) messages

#### **Sending an interactive message**

The following example is taken from the **CustCollectionsEmail** class. It demonstrates multiple features of the framework, such as the ability to chain message builder calls, conditionally set the sender address ("from" address), and add attachments.

```

using (System.IO.Stream attachmentStream = this.generateAttachment())
{
    var messageBuilder = new SysMailerMessageBuilder();
    messageBuilder.addTo(context.parmEmailAddress())
        .setSubject(emailSubject)
        .setBody(SysEmailMessage::stringExpand(emailBody,
        SysEmailTable::htmlEncodeParameters(templateTokens)));
    if (emailSenderAddr)
    {
        messageBuilder.setFrom(emailSenderAddr, emailSenderName);
    }
    else if (custParameters.CollectionsOMTeam)
    {
        var collectionsEmail = OMTeam::find(custParameters.CollectionsOMTeam).primaryEmail();
        if (strLen(collectionsEmail) > 0)
        {
            messageBuilder.setFrom(collectionsEmail);
        }
    }
    if (attachmentStream != null)
    {
        messageBuilder.addAttachment(
            attachmentStream,
            strFmt('%1%2', strReplace(DateTimeUtil::toStr(DateTimeUtil::utcNow()), ':', ''), '.xlsx'));
    }
    SysMailerFactory::sendInteractive(messageBuilder.getMessage());
}

```

### **Sending a non-interactive (batch) message**

The following example is taken from the **VendRequestCompanyWorkflowManager** class. It shows how you can send a single message non-interactively.

```

// The vendor <vendor name> has been approved and has been added to the vendor master.
messageText = strFmt("@SYS134393", DirPartyTable::findRec(vendRequestCompany.VendParty).Name);
// Request
var messageBuilder = new SysMailerMessageBuilder();
messageBuilder.setFrom(senderEmail)
    .addTo(recipientEmail)
    .setSubject("@SYS130372")
    .setBody(messageText);
SysMailerFactory::sendNonInteractive(messageBuilder.getMessage());

```

### **Sending multiple non-interactive (batch) messages**

The following example is taken from the **UserAdAddManager** class. It shows how you can get an instance of a batch email provider before you iterate over a list of emails to send.

```

var mail = SysMailerFactory::getNonInteractiveMailer();
var messageBuilder = new SysMailerMessageBuilder();
for (i = 1; i <= conLen(_notifyCon); i++)
{
    notifyEmailsStr = conPeek(_notifyCon, i);
    select firstonly RecId, Email from sysUser where sysUser.Id == notifyEmailsStr;
    if (sysUser.RecId && sysUser.Email != '')
    {
        messageBuilder.reset()
            .setFrom(_emailFrom)
            .addTo(sysUser.Email)
            .setSubject("@SYS129183")
            .setBody(errorStr);
        mail.sendNonInteractive(messageBuilder.getMessage());
        delete_from tmpUserErrorNotification;
    }
}
}

```

### Important considerations

- A sender address ("from" address) is required when messages are sent to an email provider. A receiver address ("to" address) is required when messages are sent non-interactively. If these conditions aren't met, the framework throws an exception. If **getMessage** is called on the message builder before any call to **setFrom** is made, the builder tries to set the sender address to the current user's email address or network alias.
- When messages are sent, the way that the sender address ("from" address) is used depends on the provider:
  - **EML provider**: The sender address is removed from the message before the message is opened in the user's email client. Therefore, the email client can set the sender address to the default account that is configured for sending mail.
  - **SMTP provider**: The Simple Mail Transfer Protocol (SMTP) server must be configured to allow messages to be sent by using the sender address. In other words, the SMTP server must allow the impersonation of emails that are sent from it. Otherwise, the SMTP server might prevent the messages from being sent, flag them as spam, and so on.
- When messages are sent, the framework returns a Boolean value that indicates whether the operation to send the message was successful. However, it doesn't report any messages to the Action Center when the operation is successful. You decide whether messages are shown in the Action Center.
- By default, the body of all messages that are sent is in rich-text (HTML) format. If an application scenario requires that plain text be used to maintain newline spacing, you can pass **false** to the optional **\_isHtml** parameter of the **setBody** method on the message builder.

## Adding a new email provider

You can add email providers by using the pluggable framework. When you use the factory class and interfaces, new email providers immediately become available for use across all relevant application scenarios. Examples of email providers can be found in the existing provider implementations, `SysMailerEML` and `SysMailerSMTP`, and also in an existing tutorial implementation, `Tutorial_SysMailerMailTo`. The examples that follow are excerpts from the implementation of the `SysMailerEML` email provider.

To implement an email provider, you must create an implementation class that has the following properties:

- The class must have the appropriate **Export** attributes.
- The class must implement the base **SysIMailer** methods, **getId** and **getDescription**.
- The class must implement the **SysIMailerInteractive** interface, the **SysIMailerNonInteractive** interface, or both interfaces.

## Specify attributes for the implementation class

The first step is to specify attributes for the implementation class. The class must have two attributes:

- **ExportAttribute** – The framework uses this attribute to discover the plug-in. The attribute specifies that the plug-in is an implementation of **SysIMailer** and therefore part of the SysMailer framework.
- **ExportMetadataAttribute** – The framework uses this attribute to find specific instances of a plug-in that have specific metadata. The attribute specifies the ID of the email provider, so that a single provider can be discovered quickly. **No two email providers should ever have the same ID.**

```
using System.IO;
using System.Net.Mail;
using System.Text.RegularExpressions;
#define SysMailerEML_ID('EML')
/// <summary>
/// The <c>SysMailerEML</c> class is an interactive email provider implementation that sends messages by
generating
/// an EML file, uploading it to Azure temporary blob storage, and then redirecting the user's browser to
/// the file to save or open for sending using their default email client.
/// </summary>
// This is a framework class. Customizing this class may cause problems with future upgrades to the
software.
[System.ComponentModel.Composition.ExportAttribute(IdentifierStr(Dynamics.AX.Application.SysIMailer)),
System.ComponentModel.Composition.ExportMetadataAttribute(extendedTypeStr(SysMailerId), #SysMailerEML_ID)]
public class SysMailerEML implements SysIMailerInteractive
{
```

## Implement the SysIMailer interface

The **SysIMailer** interface includes identifiable information about an email provider, regardless of the capabilities of the email provider itself. To implement this interface, you must create two methods:

- **getId** – This method returns the same ID that is specified in the **ExportMetadataAttribute** attribute.
- **getDescription** – This method returns a non-technical description that indicates how the email will be sent.

```
public SysMailerId getId()
{
    return #SysMailerEML_ID;
}
public SysMailerDescription getDescription()
{
    // Use an email app, such as Outlook
    return "@ApplicationFoundation:EmailProviderEMLDescription";
}
```

## Implement a combination of the SysIMailerInteractive and SysIMailerNonInteractive interfaces

The **SysIMailerInteractive** and **SysIMailerNonInteractive** interfaces are very simple. They implement the **sendInteractive** and **sendNonInteractive** methods, respectively. An email provider might implement one or both methods, depending on its capabilities. Each method that is implemented takes a **.NET System.Net.Mail.MailMessage** object that you use to construct and send the email.



```
public boolean sendInteractive(System.Net.Mail.MailMessage _message)
{
    using (var emlStream = this.generateEmIFile(_message))
    {
        Dynamics.AX.Application.File::SendFileToUser(emlStream, 'message.eml');
    }
    return true;
}
```

The **System.Net.Mail.MailMessage** object contains a large amount of advanced functionality that is related to MIME messages. You can build a relatively complex message object and pass it to an email provider. If there is specific functionality that an email provider doesn't support, the email provider is expected to handle the functionality actively (by modifying the message), passively (by making an internal call to another email provider), or not at all (by throwing an error).

## Migration from Microsoft Dynamics AX 2012 to Finance and Operations applications

Migration involves using the **SysMailerMessageBuilder** object to build a message and then using the **SysMailerFactory** to send it, as outlined in the examples in this topic.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Troubleshoot the Office integration

2/18/2021 • 16 minutes to read • [Edit Online](#)

## Applies to these Dynamics 365 apps:

Commerce, Finance, Human Resources, Supply Chain Management

This topic provides answers to questions, tips, and troubleshooting information about the capabilities of the Microsoft Office integration. The questions and issues that are discussed range across user, administration, and development scenarios.

## Frequently asked questions

### What platforms do the Office Add-ins support?

The Microsoft Excel Add-in and Microsoft Word Add-in are built by using the Office Web/JavaScript Add-in framework. This framework was originally released for Microsoft Office 2013 but received significant updates in Microsoft Office 2016. For more information, see [Office Add-in host and platform availability](#). The Excel Add-in requires ExcelAPI 1.2. Therefore, use the [Office Add-in host and platform availability](#) matrix to determine which platforms support the Excel Add-in. For many users, the phrase "Excel 2016 with the latest updates" is sufficient.

### Are the Office Add-ins safe?

In an age of malware, full connectivity, and compliance risks, nothing is completely secure. However, the web add-ins, like other websites, are basically a web application that interacts with the Office client products via a limited application programming interface (API). For more details, see [Office Add-ins platform overview](#)

### Does the Excel Add-in support Office for Mac?

No. Support for Apple Mac and iOS is currently under development. The Office JavaScript (JS) APIs work differently in Apple Safari and Internet Explorer, especially in respect to authentication. For details about platform support for the Office JS APIs, see [Office Add-in host and platform availability](#).

### What version of Office is required for the Excel Add-in to support AD FS?

For more information, see the "Troubleshooting issue" section later in this topic.

### How can I force an update of Office?

If your Office build isn't updated, you might be on the deferred track ([Overview of update channels for Microsoft 365 Apps](#)). In this case, you can [use the Office Deployment Tool to move to the Current channel](#) or sign up for the [Office Insider program](#) to help guarantee that you have the latest updates. The easiest method is to use the Office Deployment Tool to switch to the Current channel. In this case, the latest updates will be installed immediately.

### Why can't you tell me what version of Office or Excel a particular issue is fixed in?

Office has many releases. These releases receive updates at different times and have different version numbers that don't correspond. Some frequently used Office versions and update methods are Click to Run (C2R) Current channel, C2R Deferred, C2R First Update Deferred, Office Insider Fast, Office Insider Slow, and MSI/MSO (install from DVD). For more information about Office versions, see the [Release information for updates to Microsoft 365 Apps](#) page.

### Why am I having trouble signing into the Excel Add-in?

The Excel Add-in runs inside an Internet Explorer window. By default, the Excel Add-in picks up stored credentials from Internet Explorer, and Internet Explorer provides the current Microsoft Windows credentials if there are no stored credentials. Make sure that you're using the correct credentials to sign in. In the Excel Add-in, explicitly

sign out, and then sign in to help guarantee that the correct credentials are used.

### **The Excel Add-in seems to be slow when it publishes records. How can I learn more about what is occurring?**

Most of the work that the Excel Add-in does should occur on the server. To learn where the time is being spent, you can use [Fiddler \(a free download\)](#) to make sure that the Excel Add-in works as you expect.

The Excel Add-in sends the published records as a request. When those records are processed, the response is sent back from the server. The Excel Add-in then creates another message that contains the next set of records to publish, and sends that request. Five to ten seconds of processing time in the Excel Add-in should be required between the previous response from the server and the next request to the server.

To check processing time in the Excel Add-in versus the server/service, follow these steps.

1. Start [Fiddler](#).
2. Publish a few records to test the process.
3. Make sure that you can view that request and response in Fiddler ([make sure that HTTPS traffic is being decrypted](#)).
4. Publish a larger number of records.
5. In Fiddler, watch the time that is required from a request to its response, and from a response to the next request.
  - If the time from a request to its response is large, the bottleneck is the server/service.
  - If the time from a response to the next request is large, the bottleneck is the Excel Add-in (that is, the client).

### **Why is the Export to Excel functionality limited to 10,000 records (prior to Platform update 22)?**

Prior to Platform update 22, the Export to Excel functionality is limited to 10,000 records. This limitation is in place because the export process uses the form from which data is being exported to provide the following records with fields and data that can't be obtained otherwise: formatted values, calculated values, and temporary table data. Because the form is used during the export, it occurs inside the client process that is shared by all the users on a given computer. During the export, those other users are blocked from interacting with the client.

With Platform update 22 and later, Export to Excel has a progress dialog box and is no longer a blocking process for other users, so larger datasets can be exported. Exporting data via Export to Excel will be slower than using the Excel Add-in or the Data Management framework, but it will return exactly the data shown in the grid. This is useful for filtered datasets. The user is presented with a dialog box that allows them to stop at any point. Because the export can take some time, it is recommended that the export is done with the Chrome or Edge browsers, with the automatic download option enabled. The automatic download option will ensure that the browser downloads the file as soon as the export is complete to ensure that the download link is used within the 15-minute time limit.

The ideal alternative to Export to Excel is to use Open in Excel and the Excel Add-in. The Excel Add-in retrieves data by using the OData service, and takes advantage of the security that the entities provide. The import and export capabilities in the Data management framework (DMF) and Data import/export framework (DIXF) can also be used. However, DMF/DIXF is often limited to administrators.

If you have concerns about giving users access to the data via the Excel Add-in, because they should not be able to update records, consider the following points:

- The entities should have all the validation and logic that the forms have. If they don't, it's a bug.
- The way that entities are secured resembles the way that forms are secured. Therefore, if a user should not have permission to update or write data by using a form that exposes that data, the user should not have permission to update or write data by using an entity that exposes that data.

### Why is the Publish button in the Excel Add-in unavailable?

All key and mandatory fields must be present to publish data back to the entity. Try to edit the design to add more fields to the binding.

### Why are the Excel Add-in, the Word Add-in, and the Open in Excel options only available when the Internet is available?

For all environments, including on-premises, the Excel and Word Add-ins, and the libraries they use, are loaded from multiple Internet locations and therefore will only run when the Internet is available. For on-premises environments, when the Internet is not available, the Open in Excel options are hidden because the Excel Add-in will not run without access to the Content Delivery Network (CDN) that houses the Excel Add-in.

### Can the Excel Add-in and Word Add-in be made available to users using Centralized Deployment?

Yes, Centralized Deployment is supported. For more information, see [Centralized Deployment](#).

To use Centralized Deployment, on the **App parameters** tab on the **Office App Parameters** page change the **App ID**, **Store**, and **Store Type**:

- **App ID:** 61bcc63f-b860-4280-8280-3e4fb5ea7726
- **Store:** EXCatalog
- **Store Type:** Centralized Deployment

In case a change back to Office Store is needed, the standard values are:

- **App ID:** WA104379629
- **Store:** en-US
- **Store Type:** Office Store

#### NOTE

- **Name**, **Version**, and **Notes** are values that provide information but they are not needed to run the Excel Add-in.
- These same values are also used for the Word Add-in when it is run from the Document Templates form.

If you encounter issues with Centralized Deployment for some users, it could be one of these problems:

- One or more users are members in a group that is more restrictive than others
- The user referenced is on a different Microsoft 365 account (such as a personal account)

### What is the cell limit for the Excel Add-in?

The default Excel Add-in cell limit is about half the limit of what the Excel Add-in can handle on a reasonably fast machine. The speed of the machine is the limitation. If problems are encountered, then the cell limit should be reduced and/or the filter should be adjusted to reduce the data set. A common workaround is to use a filter to manage the data in smaller pieces instead of all at once.

### How do I make an entity available in the Excel Add-in and/or as an Open in Excel option?

If the entity is marked as "IsPublic=Yes" and has unique PublicEntityName and PublicCollectionName values, then it will be available via the OData service. Check that there aren't any existing entities with the same PublicEntityName and PublicCollectionName values by looking at the \$metadata feed for the environment (preferably in Google Chrome): [https://SomeFullEnvironmentURL.dynamics.com/data/\\$metadata](https://SomeFullEnvironmentURL.dynamics.com/data/$metadata)

### Why are date and time values in UTC in the Excel Add-in?

The Excel Add-In, Data Management Framework, and Power BI reporting are all designed to interact with data directly on the database level. Because there is no client to adjust Date and Time data to the time zone of the user, all Date and Time values are in UTC.

# Troubleshooting issues

**[Fixed] Issue: During sign-in to the Excel Add-in, I receive the following error message: "AADSTS65001: The user or administrator has not consented to use the application with ID XYZ"**

**Issue:** During sign in to the Excel Add-in, you receive the following error message: "AADSTS65001: The user or administrator has not consented to use the application with ID XYZ."

**Explanation:** Typically, this issue occurs because Microsoft Azure Active Directory (Azure AD) can't find the Azure AD application that represents the Excel Add-in. That issue occurs because, during the [configuration of Microsoft Power BI](#), an Azure AD application was added that has the App ID URI set to the environment URL.

**Fix:** Make sure that no Azure AD apps have the App ID URI set to the environment URL. App ID URIs should be fabricated, unique URIs, such as `https://contosoAXPowerBI`.

**[Fixed] Issue: During sign-in to the Excel Add-in, I receive the following error message: "AADSTS50001: The application named ABC was not found in the tenant named XYZ"**

**Issue:** During sign-in to the Excel Add-in, you receive the following error message: "AADSTS50001: The application named ABC was not found in the tenant named XYZ."

**Explanation:** This issue probably occurs because an error in the deployment system caused the environment to get a URL that wasn't added to the configured list of service principals for the tenant.

**Fix:** File a support issue for your environment, so that the problem can be investigated and the configuration can be adjusted.

**[Fixed] Issue: After the Excel Add-in starts and updates data, I receive the following error message: "An error occurred while writing to the data cache"**

**Issue:** After the Excel Add-in starts and updates data, you receive the following error message: "An error occurred while writing to the data cache." The details of the error state, "The argument is invalid or missing or has an incorrect format."

**Explanation:** You receive this error message if the client is open in Internet Explorer, and the user clicks **Open** immediately after selecting the **Open in Excel** option. The way that Internet Explorer handles temporary Internet files causes an issue in Excel. This issue, in turn, causes API calls to fail.

**Workaround:** In Internet Explorer, when you open a workbook, click **Save** first, and then click **Open**. The file will then be opened from your Downloads folder. Alternately, use the Edge or Google Chrome browser. By default, both these browsers save files to a Downloads folder. Therefore, the issue doesn't occur.

**Long-term fix:** We are working with the Office team to understand this issue so that it can be fixed in Excel.

**Issue: When I send email by using SMTP, the server response is "5.7.60 SMTP; Client does not have permissions to send as this sender"**

**Issue:** When you send email by using Simple Mail Transfer Protocol (SMTP), you might receive an error message that states that the server response was "5.7.60 SMTP; Client does not have permissions to send as this sender." Alternatively, the error message might state, "Something went wrong while generating the report."

**Explanation:** This issue is usually caused by incorrect setup of the Send As permissions for the email account.

**Fix:** You can configure Send As permissions in the Microsoft 365 admin center ([portal.office.com/Admin](https://portal.office.com/Admin)). Click **Users > Active users > User > Edit mailbox permissions > Send email from this mailbox**. For more information, see [Give mailbox permissions to another user in Microsoft 365 - Admin Help](#).

The following illustration shows the setup of SMTP on the **Email parameters** page. Here, you must provide the outgoing mail server, port, user name, password, and Secure Sockets Layer (SSL) requirements.

## Email parameters

Email providers

SMTP settings

### SMTP settings

Outgoing mail server: smtp.office365.com  
SMTP port number: 587  
User name: serviceacct@d365forops.onmicro  
Password: [REDACTED]  
Specify if SSL is required: Yes

### IMPORTANT

All users must give the SMTP account Send As permissions on their email setup in Microsoft 365. This configuration is done in the mailbox permissions in Microsoft Exchange or in the Microsoft 365 Admin portal. The following illustration shows the setup for the Test User account, where the SMTP service account is added in the **Send As** section.

### Test User

- general
- mailbox usage
- contact information
- organization
- email address
- mailbox features
- member of
- MailTip
- ▶ mailbox delegation

#### Send As

The Send As permission allows a delegate to send email from this

+ -

USER PRINCIPAL NAME

SMTP AUTHORITY

serviceacct@d365forops.onmicrosoft.com

#### Send on Behalf

The Send on Behalf permission allows the delegate to send email

+ -

DISPLAY NAME

#### Full Access

The Full Access permission allows a delegate to open this mailbox

+ -

DISPLAY NAME

### [Fixed] Issue: The Office Add-ins don't yet support AD FS

Affected versions: CTP8 and the February 2016 releases

**Issue:** When users from an Azure AD tenant that uses Active Directory Federation Services (AD FS) try to sign in to the Office Add-ins (in other words, when the users enter their account, and then press Tab or click in the field to enter their password), a separate browser window opens. This browser window usually has a URL that starts with `https://az689774.vo.msecnd.net/dynamicsofficeapp/v1.2.1.0/App/DynamicsApp.html?id_\token=`. The user can't sign in.

**Explanation:** During sign-in to the Office add-ins (both the Excel Add-in and the Word Add-in), a redirect to the AD FS site for the tenant occurs. However, that site is an unknown and therefore disallowed application domain (AppDomain).

**Long-term fix:** The long-term fix for this issue was put in place on May 10, 2016. The Office Add-ins now use a new Dialog API that the Office team added.

**Taking advantage of the add-in updates that support AD FS:** All Office installations should be updated via **File > Account > Updates** (for click-to-run installations) or via Windows Update (for MSI installations). The AD FS Dialog API was included in the May update (16.0.6868.2060). For information about updates, see the [Microsoft 365 client update channel releases](#) page.

If your Office build isn't updated, you might be on the deferred track ([Overview of update channels for Microsoft 365 Apps](#)). In this case, you can [use the Office Deployment Tool to move to the Current channel](#) or sign up for the [Office Insider program](#) to help guarantee that you have the latest updates. Additionally, see [Install the latest version of Office](#) and [Office 2016 Deployment Guides for Admins](#).

If Office updates can't be installed, the following workaround can unblock users.

**Workaround: Use Internet Explorer to sign in to the client before you use the Office Add-ins**

This workaround requires user knowledge and extra steps. After users have been educated about this workaround, it should be straightforward for them.

**User steps:** Before users open Excel (or Word), they should sign in to the client by using Internet Explorer.

**Explanation:** The Excel or Word Add-in will use the sign-in context, and no redirect will be required. The earlier sign-in must occur in Internet Explorer, because the Office Add-ins run inside an Internet Explorer window in Excel and Word. The sign-in context lasts 6 to 24 hours, depending on policies. Therefore, a new sign-in through Internet Explorer is required only occasionally.

1. Exit Internet Explorer and Excel.
2. Start Internet Explorer, and sign in to the client.
3. Test the Excel Add-in by using an Open in Excel experience. (For example, click **Fleet Management > Customers > Customer > Open in Microsoft Office > Open in Excel > Fleet Management Customers**.)

**[Fixed] Issue: The Excel Add-in doesn't correctly run or enable sign-in**

**Issue:** When users try to sign into the Excel Add-in, a blank authentication dialog box appears, or an error message is shown instead of the authentication page. The user can't sign in.

**Explanation:** The Excel Add-in relies on the Office Web JS Add-in platform and uses Azure AD for authentication. If a proxy is used, several URLs must be accessible for users to run and sign in to the Excel Add-in. Additionally, if AD FS is used, the AD FS URL must use HTTPS.

**Solution:** Because this issue is a customer-specific network issue, it requires a customer-specific resolution. If AD FS is used, make sure that the AD FS URL uses HTTPS. Additionally, make sure that all the following URLs are accessible from the user's computer.

The following URLs are accessed for loading.

- `https://az689774.vo.msecnd.net:443`
- `https://az689774.vo.msecnd.net`
- `https://appsforoffice.microsoft.com:443`
- `https://appsforoffice.microsoft.com`
- `https://secure.aadcdn.microsoftonline-p.com:443`
- `https://secure.aadcdn.microsoftonline-p.com`
- `https://az416426.vo.msecnd.net:443`
- `https://az416426.vo.msecnd.net`
- `https://telemetryservice.firstpartyapps.oaspapps.com:443`

- <https://telemetryservice.firstpartyapps.oaspapps.com>
- <https://nexus.officeapps.live.com:443>
- <https://nexus.officeapps.live.com>
- <https://browser.pipe.aria.microsoft.com:443>
- <https://browser.pipe.aria.microsoft.com>
- <http://schemas.microsoft.com>

The following URLs are accessed for authentication.

- <https://login.windows.net:443>
- <https://login.windows.net>
- <https://login.microsoftonline.com:443>
- <https://login.microsoftonline.com>

### **Issue: The Excel Add-in needs an explicit sign out after encountering an AADSTS50058 "silent sign in failed" error**

**Issue:** When users try to sign in to the Excel Add-in after some period of inactivity, the user encounters the AADSTS50058 "silent sign in failed" error and is forced to sign out before signing back in.

**Explanation:** The Excel Add-in uses Azure AD for authentication. When authentication occurs, a token is created for the user. That token has an expiration period. After the token has expired, an AADSTS50058 error will occur indicating that "silent sign in failed".

**Solution:** The user needs to sign out and sign back in. We will improve this behavior in the future by automatically signing the user out to enable faster sign in.

### **Issue: When trying to use a document template with Open in Excel a "Record for id GUID not found" error displays**

**Issue:** The "Record for id GUID not found" error can display when copying a database from one environment to another.

**Explanation:** Copying the database is problematic for document templates, record attachments, and other files that are stored in Azure blob storage. When the database is copied from one environment to another, the files are not copied along with the records, so the files that the application tries to access are not found.

**Solution:** For document templates, the solution is to identify the templates that are needed and load a copy of those template files into the target environment.

## Additional resources

[Office integration](#)

[Office integration tutorial](#)

[Configuring Power BI integration](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Organization administration home page

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic points to content that will help power users and administrators configure the system to work smoothly and effectively for your organization and business.

Much of the content listed here applies to features in the **Organizational administration** module. However, there are a couple of tasks, such as creating and using a record template, that can be performed in any module to help your organization run more efficiently.

## Number sequences

Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require identifiers. A master data record or transaction record that requires an identifier is referred to as a *reference*. Before you can create new records for a reference, you must set up a number sequence and associate it with the reference.

- [Number sequences overview](#)
- [Set up number sequences using a wizard](#) (Task guide)
- [Set up number sequences on an individual basis](#) (Task guide)

## Organizations

An organization is a group of people who are working together to carry out a business process or achieve a goal. Organizational hierarchies represent the relationships between the organizations that make up your business.

Before you set up organizations and organization hierarchies, make sure that you plan how your business will be modeled. The organization model has a significant effect on implementation and business processes.

- [Organizations and organizational hierarchies overview](#)
- [Plan your organizational hierarchy](#)
- [Create an organization hierarchy](#) (Task guide)
- [Create a legal entity](#) (Task guide)
- [Create an operating unit](#) (Task guide)

## Address books

The global address book is a centralized repository for master data that must be stored for all internal and external persons and organizations that the company interacts with. The data that is associated with party records includes the party's name, address, and contact information.

After you create the global address book, you can create additional address books as you require, such as a separate address book for each company in your organization or for each line of business.

- [Global address book overview](#)
- [Plan for the global address book and other address books](#)
- [Configure the global address book](#)
- [Address books FAQ](#)

# Workflow

Workflow is a system that you can use to create individual workflows, or business processes. When you create a workflow, you specify how a document flows, or moves, through the system by showing who must complete a task, make a decision, or approve a document.

- [Workflow system overview](#)
- [Workflow elements](#)
- [Actions in workflow approval processes](#)
- [Create workflows overview](#)

# Electronic signatures

An electronic signature confirms the identity of a person who is about to start or approve a computing process. In some industries, an electronic signature is as legally binding as a handwritten signature. Electronic signatures are a regulations compliance requirement for several regulated industries, such as pharmaceuticals, food and beverage, and aerospace and defense.

You can use electronic signatures for critical business processes. Some processes have built-in electronic signature capabilities. You can also create custom signature requirements for any database table and field.

- [Electronic signatures overview](#)
- [Set up electronic signatures](#) (Task guide)

# Case management

By planning, tracking, and analyzing cases, you can develop efficient resolutions that can be used for similar issues. For example, when customer service representatives or Human Resources generalists create cases, they can find information in knowledge articles to help them work with or resolve a case more efficiently.

- [Case management overview](#)
- [Plan case category security, case processes, and case categories](#)

# Record templates

Record templates can help you to create records more quickly. You can create a record template so that field values that are used often do not have to be entered explicitly for each new record.

- [Record templates overview](#)
- [Create a record template to facilitate data entry](#) (Task guide)
- [Use record template to create a new record](#) (Task guide)

# General organization administration

- [Change the banner or logo](#) (Task guide)
- [Configure document management](#)
- [Configure and send email](#)
- [Date/time data and time zones](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Number sequences overview

2/18/2021 • 4 minutes to read • [Edit Online](#)

Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require identifiers. A master data record or transaction record that requires an identifier is referred to as a *reference*.

Before you can create new records for a reference, you must set up a number sequence and associate it with the reference. We recommend that you use the pages in **Organization administration** to set up number sequences. If module-specific settings are required, you can use the parameters page in a module to specify number sequences for the references in that module. For example, in **Accounts receivable** and **Accounts payable**, you can set up number sequence groups to allocate specific number sequences to specific customers or vendors.

When you set up a number sequence, you must specify a scope, which defines which organization uses the number sequence. The scope can be **Shared**, **Company**, **Legal entity**, or **Operating unit**. **Legal entity** and **Company** scopes can be combined with **Fiscal calendar period** to create even more specific number sequences.

Number sequence formats consist of segments. Number sequences with a scope other than **Shared** can contain segments that correspond to the scope. For example, a number sequence with a scope of **Legal entity** can contain a legal entity segment. By including a scope segment in the number sequence format, you can identify the scope of a particular record by looking at its number.

In addition to segments that correspond to scopes, number sequence formats can contain **Constant** and **Alphanumeric segments**. A **Constant** segment contains a set of letters, numbers, or symbols that does not change. An **Alphanumeric** segment contains a set of letters or numbers that increment every time that a number is used. Use a number sign (#) to represent incrementing numbers and an ampersand (&) to represent incrementing letters. For example, the format #####\_2017 creates the sequence 00001\_2017, 00002\_2017, and so on.

## Number sequence examples

The following examples show how to use segments to create number sequence formats. In particular, the examples demonstrate the effects of using scope segments.

### Expense report numbers

In the following example, expense report numbers are set up for the legal entity that is titled **CS**.

- **Area:** Travel and expense
- **Reference:** Expense report number
- **Scope:** Legal entity
- **Legal entity:** CS

SEGMENTS	SEGMENT TYPE	VALUE
Segment 1	Legal entity	CS
Segment 2	Constant	-EXPENSE-

SEGMENTS	SEGMENT TYPE	VALUE
Segment 3	Alphanumeric	####

**Example of formatted number:** CS-EXPENSE-0039

You can set up a similar number sequence format for other legal entities. For example, for a legal entity that is named **RW**, if you change only the value of the legal entity segment, the formatted number is RW-EXPENSE-0039. You can also change the whole number sequence format for other legal entities. For example, you can omit the legal entity scope segment to create a formatted number such as Exp-0001.

### Sales order numbers

In the following example, sales order numbers are set up for the company ID CEU.

- **Area:** Sales
- **Reference:** Sales order
- **Scope:** Company
- **Company:** CEU

SEGMENTS	SEGMENT TYPE	VALUE
Segment 1	Constant	SO-
Segment 2	Alphanumeric	####

**Example of formatted number:** SO-0029

Even though a scope segment is not included in the format, numbering restarts for each company ID. If you use the same format for all company IDs, the same numbers are used in each company. For example, sales order number SO-0029 is used in each company. You can also change the whole number sequence format for other company IDs.

### Purchase requisition numbers

In the following example, purchase requisition numbers are organization-wide.

- **Area:** Purchase
- **Reference:** Purchase requisition
- **Scope:** Shared

SEGMENTS	SEGMENT TYPE	VALUE
Segment 1	Constant	Req
Segment 2	Alphanumeric	####

**Example of formatted number:** Req0052

Because the scope is **Shared**, the number sequence format is used across the organization. You cannot set up different number sequence formats for different parts of the organization.

## Performance considerations for number sequences

Consider the following information about how the configuration of number sequences can affect system performance before you set up number sequences.

## Continuous and non-continuous number sequences

Number sequences can be continuous or non-continuous. A continuous number sequence does not skip any numbers, but numbers may not be used sequentially. Numbers from a non-continuous number sequence are used sequentially, but the number sequence may skip numbers. For example, if a user cancels a transaction, a number is generated, but not used. In a continuous number sequence, that number is recycled later. In a non-continuous number sequence, the number is not used.

Continuous number sequences are typically required for external documents, such as purchase orders, sales orders, and invoices. However, continuous number sequences can adversely affect system response times because the system must request a number from the database every time that a new document or record is created.

If you use a non-continuous number sequence, you can enable **Preallocation** on the **Performance** FastTab of the **Number sequences** page. When you specify a quantity of numbers to preallocate, the system selects those numbers and stores them in memory. New numbers are requested from the database only after the preallocated quantity has been used.

Unless there is a regulatory requirement that you use continuous number sequences, we recommend that you use non-continuous number sequences for better performance.

## Automatic cleanup of number sequences

In case of a power failure, an application error, or other unexpected failure, the system cannot recycle numbers automatically for continuous number sequences. You can run the cleanup process manually or automatically to recover the lost numbers.

Carefully consider server usage when you plan the cleanup process. We recommend that you perform the cleanup as a batch job during non-peak hours.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up number sequences on an individual basis

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to set up number sequences on an individual basis. Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require them. A master data or transaction record that requires an identifier is referred to as a reference. Before you can create new records for a reference, you must set up a number sequence and associate it with the reference. You can set up all required number sequences at the same time by using the **Set up number sequences** wizard, or you can create or modify individual number sequences by using the **Number sequences** page.

1. Go to **Navigation pane > Modules > Organization administration > Number sequences > Number sequences**.
2. Select **Number sequence**.
3. In the **Number sequence code** field, type a value.
4. In the **Name** field, type a value.
5. On the **Scope parameters** FastTab, select a scope for the number sequence and select scope values from the drop-down list. The scope defines which organizations use the number sequence. In addition, number sequences that have a scope other than **Shared** can have segments that correspond to their scope. For example, a number sequence with a scope of **Legal entity** can have a legal entity segment. For more information about scopes, see [Number sequence overview](#).
6. Expand the **Segments** section.
  - Define the format for the number sequence by adding, removing, and rearranging segments.
  - Number sequences of all scopes can contain *Constant segments* and *Alphanumeric segments*. Constant segments contain a set of alphanumeric characters that do not change. Use this segment type to add a hyphen or other separators between number sequence segments. Alphanumeric segments contain a combination of number signs (#) and ampersands (&). These characters represent letters and numbers that increment every time that a number from the sequence is used. Use a number sign (#) to indicate incrementing numbers and an ampersand (&) to indicate incrementing letters. For example, the format `#####_2014` creates the sequence `00001_2014`, `00002_2014`, and so on. At least one alphanumeric segment must be present. Scope segments, such as company or legal entity, are not required. However, if you do not include scope segments in the format, numbers for the selected reference are still generated per scope.
7. Expand the **References** section. Select the document type or record to assign this number sequence to. This step is optional for sequences that are defined for special application usage patterns. In these scenarios, a new number is generated by using the value of a number sequence code or ID, without using a reference. An example of a special application usage pattern is a voucher series that is used for specific journal names. However, we do not recommend that you use such patterns.
8. Expand the **General** section. On the General FastTab, specify whether the number sequence is manual, and continuous or non-continuous. In addition, enter the lowest and highest numbers that can be used in the number sequence. We do not recommend changing a non-continuous number sequence to a continuous number sequence. The number sequence will not be truly continuous. This change may also cause duplicate key violations in the database. In addition, continuous number sequences have a larger effect on performance.
9. Click **Save**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up number sequences using a wizard

2/18/2021 • 2 minutes to read • [Edit Online](#)

Number sequences are used to generate readable, unique identifiers for master data records and transaction records that require them. A master data or transaction record that requires an identifier is referred to as a reference. Before you can create new records for a reference, you must set up a number sequence and associate it with the reference. This topic explains how to set up all required number sequences at the same time by using a wizard. The demo data company used to create this procedure is USMF.

1. Go to **Navigation > Modules > Organization administration > Number sequences > Number sequences**.
2. Select **Generate**.
3. Select **Next**.
  - On this page, you can modify the identification code, the lowest value, and the highest value. In addition, you can indicate whether the number sequence must be continuous.
  - Do not select the **Continuous** option if you must preallocate numbers for the number sequence. To add a scope segment to the format of a number sequence, select the format in the list, and then select **Include scope in format**. To remove a scope segment from the format of a number sequence, select the format in the list, and then select **Remove scope from format**. To exclude a number sequence from automatic generation, select the number sequence in the list, and then select **Delete**.
4. Select **Next**.
5. Select **Finish**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Organizations and organizational hierarchies overview

2/18/2021 • 3 minutes to read • [Edit Online](#)

An organization is a group of people who are working together to carry out a business process or achieve a goal. Organizational hierarchies represent the relationships between the organizations that make up your business.

## Organizations

You can define the following types of internal organizations: legal entities, operating units, and teams.

All internal organizations are types of the **Party** entity. Therefore, these organizations use the address book to store address and contact information. A party, which can be either a person or an organization, can belong to one or more address books.

### Legal entities

A legal entity is an organization that has a registered or legislated legal structure. Legal entities can enter into legal contracts and are required to prepare statements that report on their performance.

A company is a type of legal entity. Currently, companies are the only kind of legal entity that you can create, and every legal entity is associated with a company ID. This association exists because some functional areas in the program use a company ID, or DataAreaid, in their data models. In these functional areas, companies are used as a boundary for data security. Users can access data only for the company that they are currently logged on to.

### Operating units

An operating unit is an organization that is used to divide the control of economic resources and operational processes in a business. People in an operating unit have a duty to maximize the use of scarce resources, improve processes, and account for their performance.

The types of operating units include cost centers, business units, value streams, departments, and commerce channels. The following table provides more information about each type of operating unit.

OPERATING UNIT TYPE	DESCRIPTION	PURPOSE
Cost center	An operating unit in which managers are accountable for budgeted and actual expenditures.	Used for the management and operational control of business processes that span legal entities.
Business unit	A semi-autonomous operating unit that is created to meet strategic business objectives.	Used for financial reporting that is based on industries or product lines that the organization serves independently of legal entities.
Value stream	An operating unit that controls one or more production flows.	Commonly used in lean manufacturing to control the activities and flows that are required to supply a product or service to consumers.

OPERATING UNIT TYPE	DESCRIPTION	PURPOSE
Department	An operating unit that represents a category or functional part of an organization that performs a specific task, such as sales or accounting.	Used to report on functional areas. A department may have profit and loss responsibility, and may consist of a group of cost centers.
Commerce channel	An operating unit that represents a brick and mortar store, an online store or an online marketplace.	Used for the management and operational control of one or more stores within or across legal entities.

### Teams

A team is an organization in which the members share a common responsibility, interest, or objective. Teams cannot be used in organizational hierarchies.

## Organizational hierarchies

Set up organizational hierarchies to view and report on your business from different perspectives. For example, you can set up a hierarchy of legal entities for tax, legal, or statutory reporting. Set up a hierarchy that is based on operating units to report financial information that is not legally required, but that is used for internal control. For example, you can create a purchasing hierarchy to control purchasing policies, rules, and business processes.

Each hierarchy is assigned a purpose. The purpose of a hierarchy determines the types of organizations that can be included in the hierarchy. The purpose also determines which application scenarios a hierarchy can be used in.

Organizations in a hierarchy can share parameters, policies, and transactions. An organization can inherit or override the parameters of its parent organization. However, shared master data, such as products and address books, applies to the whole organization and cannot be overridden for individual organizations. Creating organizations and hierarchies requires careful planning. For more information, see [Plan your organizational hierarchy](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Plan your organizational hierarchy

2/18/2021 • 13 minutes to read • [Edit Online](#)

Before you set up organizations and organization hierarchies, make sure that you plan how your business will be modeled. The organization model has a significant effect on the implementation and business processes.

Organizational hierarchies represent the relationships between the organizations that make up a business. Therefore, the most important consideration when you model organizations is the structure of your business. We recommend that you define organization structures based on feedback from executives and senior managers from functional areas, such as finance and accounting, human resources, operations, purchasing, and sales and marketing.

When you are planning hierarchies, it is also important to consider the relationship between the organizational hierarchy and financial dimensions. You can set up multiple organizational hierarchies to represent different views of your business. By using financial dimensions, you can create reports based on these views. Work with your partner to create hierarchies that address both organizational and statutory reporting needs.

## NOTE

Although you can use financial dimensions to represent legal entities without creating the legal entities, financial dimensions aren't designed to address the operational or business needs of legal entities. The interunit accounting functionality is designed to address only the accounting entries that are created by each transaction.

## IMPORTANT

You shouldn't decide how to model organizations based only on the information in this article. This documentation is a guide. You can work with your Partner for additional guidance. Your Partner has gained experience in various industries and across the customer base.

## Decide whether to model internal organizations as legal entities or operating units

You must have at least one legal entity to represent your business. A legal entity can enter legal contracts and is required to prepare financial statements that report on its performance.

Legal entities can be used for transactional business or for consolidation. This means that a legal entity in Finance and Operations does not necessarily represent a real entity in your business. For example, a company that participates in transactions can own subsidiary legal entities. In this scenario, a legal entity is required for transactions, and a virtual legal entity is required to consolidate the results and balances of the subsidiary legal entities.

Internal organizations in your business, such as regional offices, can be represented as additional legal entities, or as operating units of the main legal entity. An operating unit is not required to be a legally defined organization. Operating units are used to control economic resources and operational processes in the business. For example, departments and cost centers are operating units.

Some functionality works differently depending on whether the organization is a legal entity or an operating unit. Carefully consider the functionality described below as you make your decision.

### Master data

**If the organization is modeled as a legal entity**

Some master data, such as customers, payment terms, tax authorities, and site-specific stock ordering, must be set up for each legal entity. Some master data, such as users, products, and most human resources data, is shared among all legal entities.

**If the organization is modeled as an operating unit**

Master data is shared among operating units.

**Module parameters****If the organization is modeled as a legal entity**

Parameters for modules, such as Accounts receivable parameters, Accounts payable parameters, and Cash and bank management parameters, must be set per legal entity. Because the module setup for legal entities is separate, each subsidiary can comply with local statutory requirements and business practices. For example, a professional services legal entity and a manufacturing legal entity can have different module parameters even though they report to the same parent company.

**If the organization is modeled as an operating unit**

Module parameters are shared among operating units.

**Data security****If the organization is modeled as a legal entity**

Most data is automatically secured by company ID. A company ID is a unique identifier for the data that is associated with a legal entity. A company can be associated with only one legal entity, and a legal entity can be associated with only one company. Users can access data only for the companies that they have access to. You do not need to customize to secure data by company ID.

**If the organization is modeled as an operating unit**

Data can be secured per operating unit by creating customized data security policies. Data security policies are used to limit access to data. For example, assume that a user is allowed to create purchase orders only in a particular operating unit. Data security policies can be created to prevent the user from accessing purchase order data from any other operating unit. The volume of transactions and the number of security policies can affect performance. When you design security policies, keep performance in mind.

**Ledgers****If the organization is modeled as a legal entity**

Each legal entity requires a ledger that provides a chart of accounts, accounting currency, reporting currency, and fiscal calendar. A balance sheet can be created only for a legal entity. Main accounts, dimensions, account structures, charts of accounts, and account rules can be used by more than one legal entity.

**If the organization is modeled as an operating unit**

An operating unit can't have its own ledger information. If your internal organizations do not require unique ledgers, you can model them as operating units. Ledger information will be set up for the parent legal entity in the hierarchy. Income statements can be created for operating units within a legal entity or for the parent legal entity.

**Fiscal calendars****If the organization is modeled as a legal entity**

Each legal entity has its own fiscal calendar. If your internal organizations use different fiscal years and fiscal calendars, you must model the organizations as legal entities.

**If the organization is modeled as an operating unit**

Operating units must share a fiscal calendar. If your internal organizations can use the same fiscal years and fiscal calendars, you can model the organizations as operating units.

**Consolidation****If the organization is modeled as a legal entity**

You must consolidate the financial results for regional offices into a single, consolidated company in order to

prepare financial statements.

**If the organization is modeled as an operating unit**

Consolidation is not required, because data is already shared among operating units.

**Centralized payments**

**If the organization is modeled as a legal entity**

Centralized payments must be set up so that invoices for all child legal entities can be paid to or from a single parent legal entity.

**If the organization is modeled as an operating unit**

Centralized payments are not required because all invoices are recorded in a single legal entity.

**Intercompany transactions**

**If the organization is modeled as a legal entity**

Intercompany sales orders, purchase orders, payments, or receipts can be applied to one another. You are not required to use journal vouchers. You can view intercompany transactions at the sub-ledger level (Accounts receivable, Accounts payable). The following examples illustrate how intercompany transactions are handled.

*Example 1: Headquarters provides services to regional offices and must charge the costs of those services to the regional offices*

If you model the regional office as a legal entity, you have the following options:

- Headquarters creates a journal entry to cross-charge the regional office for the expense. The transactions cannot be aged.
- Headquarters sends a purchase order for the services to the regional office. A sales order is automatically created in the legal entity for the regional office, with intercompany sub-ledger transactions.

*Example 2: Headquarters procures and pays for service that is delivered to a regional office*

If you model the regional office as a legal entity, you have the following options:

- The invoice and payment follow the regulatory requirements of headquarters. Headquarters can create a journal entry to cross-charge the regional office for the expense. The transactions cannot be aged.
- The invoice and payment follow the regulatory requirements of headquarters. Headquarters can create an intercompany sub-ledger transaction.

**If the organization is modeled as an operating unit**

Intercompany transactions among operating units are supported only through journal vouchers. An operating unit cannot issue or receive a purchase order, sales order, or invoice from another operating unit in the same legal entity. You cannot view intercompany transactions at the sub-ledger level (Accounts receivable, Accounts payable). The following examples illustrate how intercompany transactions are handled.

*Example 1: Headquarters provides services to regional offices and must charge the costs of those services to the regional offices*

If you model the regional office as an operating unit, headquarters enters an expense transaction and codes it to the regional office.

*Example 2: Headquarters procures and pays for service that is delivered to a regional office*

If you model the regional office as an operating unit, the invoice and payment follow the regulatory requirements of headquarters. The invoice can be coded to the regional office. On the income statement, use a balancing financial dimension to report costs for the regional office.

**Local tax requirements**

**If the organization is modeled as a legal entity**

A legal entity is subject to the tax laws of the tax authority in the country/region where the legal entity is registered. For example, a legal entity that is registered in Denmark is subject to Danish tax laws and regulations. A legal entity can belong to only one country/region. The country/region that you select for the primary address of the legal entity controls the country/region-specific features that are available to that legal entity. For example, if the primary address of the legal entity is in Denmark, features that are related to Danish tax laws and regulations become available. Therefore, if your organizations are in different countries/regions and require

different local tax options, you must set up the organizations as separate legal entities.

**If the organization is modeled as an operating unit**

Operating units use the country context of the parent legal entity. Operating units in the same legal entity cannot have different country/region-specific requirements. If your organizations are in the same country/region and use the same tax options, you can set them up as operating units.

**Statutory reporting for a country/region**

**If the organization is modeled as a legal entity**

For countries/regions that are supported, most statutory reports can be created. For information about which reports are available for each country/region, see the [Microsoft Dynamics Localization Portal](#). (A CustomerSource logon is required.)

**NOTE**

A posting layer in the general ledger allows you to make adjusting entries to a parent company that uses a different accounting standard than the child company. For example, for a company that uses generally accepted accounting practices in the United Kingdom (UK GAAP), you can make adjusting entries in the posting layer. These entries can be consolidated into a parent company that uses generally accepted accounting principles (GAAP) in the United States. The adjusting entries do not affect UK GAAP reporting.

**If the organization is modeled as an operating unit**

Statutory reports must be created by using another application. You must ensure that data is captured in Finance and Operations apps to support the requirements of each operating unit, where they differ from the requirements of headquarters.

**Currency**

**If the organization is modeled as a legal entity**

If your organizations must use different functional currencies, you must model the organizations as legal entities. Functional currencies are set up per legal entity. However, you can enter transactions in multiple currencies.

**If the organization is modeled as an operating unit**

If your organizations can use a single functional currency, you can model the organizations as operating units. Operating units must share a functional currency. However, you can enter transactions and create reports in multiple currencies.

**Year-end closing**

**If the organization is modeled as a legal entity**

If laws and accounting practices differ among the countries/regions where your organizations are located, you may require different year-end procedures per organization. This means that you must model the organizations as legal entities. Each legal entity has its own year-end procedures.

**If the organization is modeled as an operating unit**

If laws and accounting practices are the same among the countries/regions where your organizations are located, you may use a single set of year-end procedures. This means that you can model the organizations as operating units. All operating units must use the same year-end closing procedure.

**Number sequences**

**If the organization is modeled as a legal entity**

Number sequences for some references can be set up per legal entity. Some number sequences can be shared.

**If the organization is modeled as an operating unit**

Number sequences for some references can be set up per operating unit. Some number sequences can be shared.

## Products

### If the organization is modeled as a legal entity

Product definitions are shared, and they must be released to individual legal entities before they can be included in transactions. Each legal entity has its own set of released products that can be included in transaction documents. If your internal organizations must use different sets of products, you must model the organizations as legal entities.

#### NOTE

Even though product definitions are shared, in each legal entity where a product has been released, you can specify different sales, purchase, and stocking parameters for the item at each inventory site.

### If the organization is modeled as an operating unit

All operating units share the same set of products. If your internal organizations can share the same set of products, you can model the organizations as operating units.

## Inquiry and reporting

### If the organization is modeled as a legal entity

You must manually change companies to enter transactions and perform inquiries in multiple legal entities. Because of data security boundaries, consolidated inquiry and reporting can be resource intensive and time-consuming.

### If the organization is modeled as an operating unit

You do not need to change companies to access data from multiple operating units. Consolidated inquiry and reporting and individual regional inquiry is easier and faster.

## Best practices for modeling organizations and hierarchies

Consider the following best practices when you implement an organization hierarchy:

- Create a department to model the intersection between a legal entity and a business unit. You can then roll up data from a department to a legal entity for statutory reporting, and from a department to a business unit for internal reporting. Departments can serve as profit centers. If you use departments, you do not have to use legal entities and business units as dimensions in the account structure. You can use just departments as a dimension. However, you must use both cost centers and departments as dimensions in the account structure if cost centers are used only as cost accumulators, and departments are used for revenue recognition.
- Model multiple hierarchies for operating units if you have complex requirements for reporting profit and loss.
- In a single legal entity, do not model multiple hierarchies for the same hierarchy purpose.
- Do not create a hierarchy for every purpose. Usually, you can use one hierarchy for multiple purposes. For example, one hierarchy of operating units can be assigned to all policy-related purposes.
- Create balanced hierarchies. In a hierarchy, all nodes that are the same distance from the root node are defined as a level. In a balanced hierarchy, only one type of operating unit can occur at each level, and the distance from the root node to each level is consistent. If there are intermediate levels between a department and a legal entity or a business unit, placeholder organizations may be required to create a balanced hierarchy.
- Do not model a separate hierarchy of operating units if the structure for legal entities is also your operating structure. A mixed hierarchy of legal entities and operating units may serve both purposes.
- Before you model major restructuring scenarios, use the hierarchy's effective dates to perform an impact analysis and a validation test.
- Use draft mode to change a hierarchy before you publish a new version in a production environment.
- Limit the number of people who have permissions to add or remove organizations from a hierarchy in a

production environment. A smaller number reduces the chance that costly mistakes can occur and corrections must be made.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Create an organization hierarchy

2/18/2021 • 2 minutes to read • [Edit Online](#)

Use the following procedure to create an organizational hierarchy. You can use organizational hierarchies to view and report on your business from various perspectives. For example, you can set up one hierarchy for tax, legal, or statutory reporting. You can then set up another hierarchy to report financial information that is not legally required, but that is used for internal reporting.

Before you create an organizational hierarchy, you must create organizations. For more information, see the "Create a legal entity" or "Create an operating unit" tasks. You can also create departments and teams.

The demo data company used to create this procedure is USMF.

## Create a hierarchy

1. Go to **Navigation pane > Modules > Organization administration > Organizations > Organization hierarchies**.
2. On the **Action pane**, click **New**.
3. In the **Name** field, type a value.
4. In the **Purpose** section, click **Assign purpose**.
5. In the list, find and select the desired record. Select a purpose to assign to your organization hierarchy.
6. In the **Assigned hierarchies** section, click **Add**.
7. In the list, mark the selected row. Find the hierarchy you just created.
8. Click **OK**.

## Add organizations to the hierarchy

1. In the list, find and select the desired record. Select your hierarchy.
2. In the **Assigned hierarchies** section, click **View hierarchy**.
  - Add organizations, as necessary.
  - To add an organization, click **Edit** and then **Insert** to add the organization. When you are done making changes you can **Save** a draft and/or **Publish** the changes.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a legal entity

2/18/2021 • 2 minutes to read • [Edit Online](#)

A legal entity is an organization that is identified through registration with a legal authority. Legal entities can enter into contracts and are required to prepare statements that report on their performance. The following procedure explains how to create a legal entity. The demo data company used to create this procedure is USMF.

1. Go to **Navigation pane > Modules > Organization administration > Organizations > Legal entities**.
2. Click **New**.
3. In the **Name** field, type a value.
4. In the **Company** field, type a value.
5. In the **Country/region** field, enter or select a value.
6. Click **OK**. In the **General** section, provide the following general information about the legal entity: Enter a search name, if a search name is required. A search name is an alternate name that can be used to search for this legal entity. Select whether this legal entity is being used as a consolidation company. Select whether this legal entity is being used as an elimination company.
7. Expand the **Addresses** section. In the **Addresses** section, click **Edit** to enter address information, such as the street name and number, postal code, and city.
8. Expand the **Contact information** section. In the **Contact information** section, enter information about methods of communication, such as email addresses, URLs, and telephone numbers.
9. Expand the **Statutory reporting** section. In the **Statutory reporting** section, enter the registration numbers that are used for statutory reporting.
10. Expand the **Registration numbers** section. In the **Registration numbers** section, enter any information required by the legal entity.
11. Expand the **Bank account information** section. In the **Bank account information** section, enter bank accounts and routing numbers for the legal entity.
12. Expand the **Foreign trade and logistics** section. In the **Foreign trade and logistics** section, enter shipping information for the legal entity.
13. Expand the **Number sequences** section. In the **Number sequences** section, you can view the number sequences that are associated with the legal entity.
14. Expand the **Images** section. In the **Images** section, view or change the logo and/or dashboard image that are associated with the legal entity.
15. Expand the **Tax registration** section. In the **Tax registration** section, enter the registration numbers that are used to report to tax authorities.
16. Expand the **Tax 1099** section. In the **Tax 1099** section, enter 1099 information for the legal entity.
17. Click **Save**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create an operating unit

2/18/2021 • 2 minutes to read • [Edit Online](#)

An operating unit is an organization that is used to divide the control of economic resources and operational processes in a business. People in an operating unit have a duty to maximize the use of scarce resources, improve processes, and account for their performance. The types of operating units include cost centers, business units, departments, and value streams. Use the following procedure to create an operating unit. The demo data company used to create this procedure is USMF.

1. Go to **Navigation pane > Modules > Organization administration > Organizations > Operating units**.
2. Click **New** to open the drop dialog.
3. In the list, find and select the desired record. Select the type of operating unit you want to create.
4. In the list, click the link in the selected row.
5. In the **Name** field, type a value.
  - Expand the **General** section, if necessary.
  - Provide general information about the operating unit, such as an identification number, DUNS number, and manager.
  - Expand the **Addresses** section, if necessary.
  - Enter address information, such as the street name and number, postal code, and city. Click **Add** to enter a new address record, or click **Edit** to modify an existing address record.
  - Expand the **Contact information** section, if necessary.
  - Enter information about methods of communication, such as email addresses, URLs, and telephone numbers. To enter a new communication record, click **New**. To modify an existing communication record, click **More options > Advanced**.
6. Optionally, change the **Operating unit number** as needed. Note that this number is a unique identifier for the corresponding **Party** record and cannot be the same as any other operating unit.
7. Select **Save**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Global address book overview

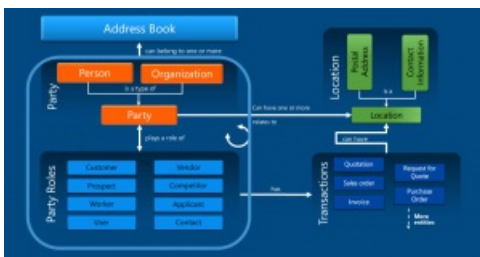
2/18/2021 • 4 minutes to read • [Edit Online](#)

The global address book is a centralized repository for master data that must be stored for all internal and external persons and organizations that the company interacts with. The data that is associated with party records includes the party's name, address, and contact information. Other details vary, depending on whether the party is a person or an organization. Each party record is assigned to a party, and each party can be associated with one or more party roles in a company. Party roles include customer, prospect, worker, user, vendor, competitor, applicant, and contact. For example, the organization party First Up Consultants, can be associated with customer, business relation, and vendor roles in the CEE company, and can also be associated with the vendor role in the CEU company. Here are some of the benefits of this shared data:

- The data shows the relationships that people and organizations have with other areas of the company. The relationship between two organizations changes when one organization has multiple roles, such as vendor and customer. Communication between the two organization also changes. There might be special agreements that can be negotiated to encourage a closer partnership with the other organization.
- Setup and maintenance are easier. For example, when an address changes, the update must be made in only one place. All the other associated records are updated automatically.

## How the global address book works

The following illustration shows how party records, party roles, locations, and transactions interact and relate to an address book. As the illustration shows, a party record can belong to one or more address books. Each party record can store one or more locations, or addresses, and is assigned a party role. The role that is assigned to the party record can have specific transactions types associated with it. The following sections provide more information about party roles, locations, and transaction types. The following image is a graphical representation of the ways that parties, party roles, locations, and transactions interact in relation to the global address book.



### Party roles

Roles that are associated with party records are referred to as party roles. There are several party roles, and they can be assigned to both party types, person and organization. Here are the definitions for each party role:

- **Customer** – Individuals, companies, or other entities who purchase goods and services that are produced by other individuals, companies, or entities.
- **Prospect** – A party that might provide a service or benefit to a legal entity.
- **Worker** – A person who assumes the role of an employee or a contractor, and who is paid in exchange for services.
- **User** – A person who is a user of the system.
- **Vendor** – A party that supplies products to one or more legal entities in exchange for payment.
- **Competitor** – A person or organization that provides goods or services that are similar to the goods or services that your business provides.

- **Applicant** – A person who makes a formal written or electronic request to work for or fill an open position in an organization.
- **Contact** – A person, either inside or outside your organization, that you have created an entry for. In this entry, you can save information such as the person's street and email addresses, telephone and fax numbers, and webpage URLs.

### **Creating new party records**

There are two ways to enter party records in the global address book:

- **Creating a party record when you don't know the role** – When you create a party record and don't know the role type (for example, you don't know whether the party is a customer or an opportunity), you create the record in the global address book. You can select the role type later.
- **Creating a party record when you know the role** – If you know the role type for the party, you can create a record on the appropriate page for that type. For example, if the party is a customer, you create a record on the **Customer** page. When you create and save a record by using the page for the party's role type, the record is automatically created in the global address book.

### **Party roles and transactions**

For transactions that are a part of the business processes, multiple parties might be associated with each transaction. An example is a customer that needs to be referenced on project quotations.

### **Parties locations, addresses, and contact information**

Each party record's addresses, locations, and contact information are shared across all the party roles that are associated with that party. Therefore, when any of this information is changed, all other associated records are updated accordingly.

### **Locations and transactions**

When a party role is included in a transaction, the location, address, or contact information of the party can be accessed when transaction details are entered.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Plan for the global address book and other address books

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the considerations and decisions that you must make during the planning process, before you set up and configure the global address book and any additional address books. Some of the decisions will require that you confirm the decisions that have been made for other areas of the product, such as the organization hierarchy.

## Global address book

Before you begin to work with the global address book, you must determine the default values for it. These default values are then used for any additional address books that you create.

### Decisions

- What sequence should names be displayed in for party records of the **Person** type? For example, one sequence is last name, middle name, first name.
- Should party records be deleted from the address book when the role record is deleted? For example, if a customer record is deleted, should the party record also be deleted?
- When a new record is created, should users be notified if a duplicate record is found in the global address book?
- Should the Data Universal Numbering System (DUNS) number be included in a party record's information?
- If the DUNS number is included in a party record, should the uniqueness of the number be checked?
- When a party record is created in the global address book, do you want a default party type, person, or organization?
- Which user roles should have access to the private addresses and contact information of party records?

## Additional address books

After you create the global address book, you can create additional address books as you require, such as a separate address book for each company in your organization or for each line of business. For example, Fabrikam is an international organization that has multiple companies and multiple lines of business. Fabrikam plans to create an address book for each line of business. For lines of business that occur in more than one location, such as the pneumatic tools business, Fabrikam plans to create an address book for each location. Chris, the IT manager at Fabrikam, has created the following list of address books that are required. This list also describes the party records that each address book must include.

- **Public Sector Contracts (PubSC)** – Party records for all parties that are involved in the public sector contracts that Fabrikam holds.
- **Private Sector Contracts (PriSC)** – Party records for all parties that are involved in the private sector contracts that Fabrikam holds.
- **Electronic Tools (ET)** – Party records for all parties that are involved in the purchase or sale of electronic tools, or that otherwise interact with the electronic tools that are provided by or purchased for Fabrikam in the Fabrikam-Japan company.
- **Pneumatic Tools (PTJPN)** – Party records for all parties that are involved in the purchase or sale of pneumatic tools, or that otherwise interact with the pneumatic tools that are provided by or purchased for Fabrikam in the Fabrikam-Japan company.

- **Pneumatic Tools (PTUSA)** – Party records for all parties that are involved in the purchase or sale of pneumatic tools, or that otherwise interact with the pneumatic tools that are provided by or purchased for Fabrikam in the Fabrikam-US company.

**Decision:**

- How many additional address books will you create?

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure the global address book

2/18/2021 • 2 minutes to read • [Edit Online](#)

Use this procedure to set the default values and security policies for the global address book.

The demo data company used to create this task is USMF. This task is intended for the Planning and configuration team.

1. In the Navigation pane, go to **Modules > Organization administration > Global address book > Global address book parameters**.
2. In the **Name sequence** field, select how names should be shown.
3. In **Delete parties with no roles**, select whether to delete parties with that have not been assigned a role.
4. In **Use duplicate check**, select whether to check for duplicate records.
5. In **Display DUNS number on addresses**, select whether to display the DUNS number on addresses.
6. In **Check for unique DUNS number**, select whether to check for unique DUNS numbers.
7. In the **Party** field, select an option.
8. In the **Customer** field, select an option.
9. In the **Vendor** field, select an option.
10. In the **Prospect** field, select an option.
11. In the **Competitor** field, select an option.
12. Click the **Private location security** tab.
13. In the list, find and select the desired record. Press the Shift key to select multiple roles to add to the **Selected roles** pane and then click the arrow to add the selected roles.
14. Click **Save**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Address books FAQ

2/18/2021 • 5 minutes to read • [Edit Online](#)

## How do I check for duplicate records?

You can check for duplicate records directly from the **Global address book** list page. On the Action Pane, on the **Party** tab, in the **Maintain** group, click **Check for duplicates**. Then select the values to include in the check for duplicates.

## Can I bulk add or delete party records from an address book?

Yes, you can add multiple party records to an address book and also remove multiple party records.

- To add multiple party records to an address book, on the **Global address book** list page, select the parties in the list. Then, on the Action Pane, on the **Party** tab, in the **Maintain** group, click **Assign parties**. Select the address books to add the selected party records to, and then click **OK**. All the selected party records are added to the address books that you selected.
- To remove multiple party records from an address book, on the **Global address book** list page, select the parties in the list. Then, on the Action Pane, on the **Party** tab, in the **Maintain** group, click **Remove parties**. Select the address books to remove the parties from, and then click **OK**. All the selected party records are removed from the address books that you selected.

## Can I change the party type of a record, or do I have to delete the old record and create a new one?

Occasionally, you might have to change the party type of a record from person to organization or from organization to person. For example, Nancy is a member of the sales team for Fabrikam U.K. At a trade show in London, she meets six new prospects. Nancy creates a prospect party record for each prospect. When Nancy saves the records, each record is also created in the global address book. Fabrikam has set the default party type to organization, but two of the new prospects should have a record type of person. Therefore, when Nancy returns from the trade show, she must change the party type of the two prospect records. To change a party record from one party type to another, you must first create a new party record of the correct type in the global address book. You then associate the old party record with this new record. After you have made the new party association, delete the original party record that has the incorrect record type.

## How do I change the name or address of a party record?

You can update the name of a party record, and the addresses that are associated with that record, at any time.

- To update the name of a party record, open the party record, and then, on the Action Pane, click **Edit**. On the **General** FastTab, enter the new name for the party, and then save the record.
- To update an address for a party record, open the party record, and then, on the **Addresses** FastTab, select the address to update. Click **Edit**, and then, on the **Edit address** page, make the required changes to the address or address parameters.

## Can I merge two or more party records into one record?

Occasionally, you might want to merge two or more party records into a single record. This can occur if you create one or more duplicate party records, either on purpose or unintentionally. When you merge party records, you select one record to keep. The information from the other records is then merged into this record.

For example, you discover that information about Fabrikam is stored in three party records: A, B, and C. You decide to keep party record A. Therefore, the information that is stored in party records B and C will be merged into party record A. There are some situations where you can't merge party records:

- You can't merge party records that are associated with the same party role, such as customer or vendor, in the same legal entity. For example, party A is associated with a customer in legal entity 123, and party B is associated with a different customer in legal entity 123. These party records can't be merged, because if they were merged, the merged party record would be associated with multiple customers in the same legal entity, and this isn't allowed. However, the records can be merged if party B is associated with a vendor in legal entity 123 or with a customer in a different legal entity.
- You can't merge internal party organization records in the same legal entity, team, or operating unit.

## Should I create a party record in the global address book or in another place, such as the Customer or Vendor page?

You can enter party records either in the global address book or on the appropriate entity page. When you add a record in one location, the same record is always added in the other location. For example, if you add a party record for a customer in the global address book, the record is also added on the **Customer** page. Likewise, if you add a party record for a customer on the **Customer** page, the record is also added in the global address book. Use the following guidelines to decide where you should enter new party records:

- **Creating a party record when you don't know the entity type** – If you must create a party record but don't know the entity type (for example, you don't know whether the entity is a customer or an opportunity), create the record in the global address book. You can select the entity type later.
- **Creating a party record when you know the entity type** – If you know the entity type for the party, you can create a record on the applicable page for that type. For example, create a record for a customer on the **Customer** page. When you create and save a record by using the appropriate entity page, the record is automatically created in the global address book.

## Can I translate address information for party records?

You can set up translations of address information, so that the information appears in your user language (system language) in your program, but in another language on documents such as sales orders. You can enter translations for country/region names, address purposes, and name sequences. For example, your system language is Danish, and you create a sales order for a customer in France. In this case, you can view the customer record in Danish in the program but display the address information in French on the printed sales order. When you set up translations, you should enter a translation for every item in the list. Any item that you don't enter a translation for will appear in the system language. For example, your system language is Danish, and you send a document to a customer in Spain. If you haven't entered Spanish (ESP) translations for the address information, that information will appear in Danish both in the program and on the printed document.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure address books

2/18/2021 • 2 minutes to read • [Edit Online](#)

Use this procedure, and the decisions that you made in the Planning the configuration of the global address book and additional address books topic, to set up additional address books for your organization.

The demo data company used to create this task is USMF. This recording is intended for the Planning and configuration team members.

## Configure address books

1. In the **Navigation pane**, go to **Modules > Organization administration > Global address book > Address books**.
2. Click **New**.
3. In the **Name** field, type a value.
4. In the **Description** field, type a value.
5. Click **Save**.
6. In the list, find and select the desired record.
7. Click the arrow to add the selected available teams to the address book.
8. Click **Save**.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Workflow system overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the workflow system.

## What is workflow?

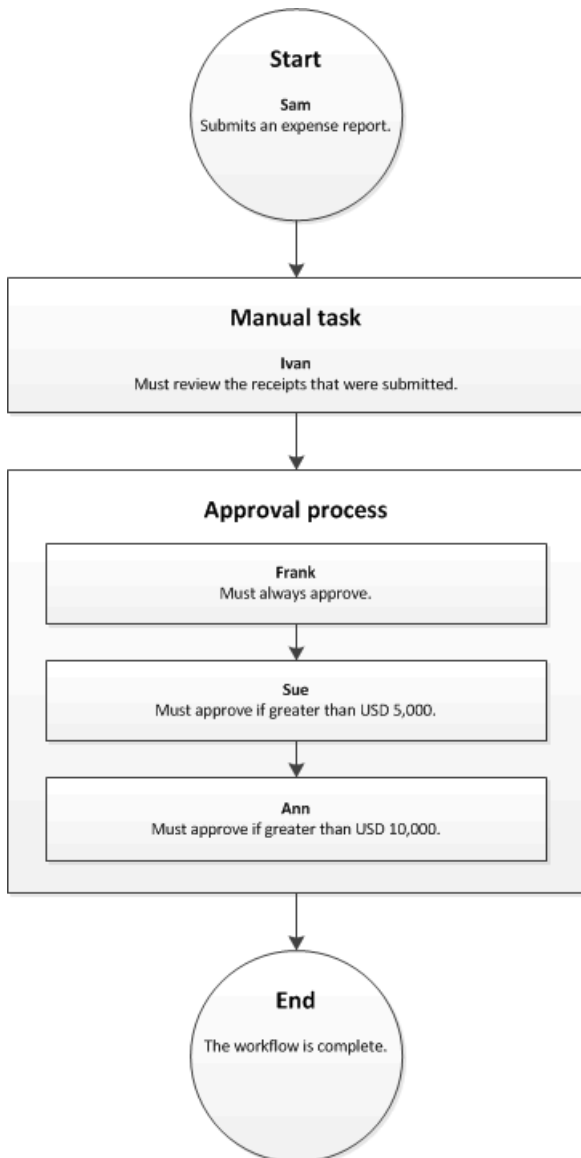
The term *workflow* can be defined in two ways: as a system and as a business process.

### Workflow is a system

Workflow is a system that runs on the Application Object Server (AOS). The workflow system provides functionality that you can use to create individual workflows, or business processes.

### Workflow is a business process

A workflow represents a business process. It defines how a document flows, or moves, through the system by showing who must complete a task, make a decision, or approve a document. For example, the following illustration shows a workflow for expense reports.



To better understand this workflow, suppose that Sam submits an expense report for USD 7,000. In this scenario, Ivan must review the receipts that Sam routes to him. Then Frank and Sue must approve the expense report.

Now suppose that Sam submits an expense report for USD 11,000. In this scenario, Ivan must review the receipts, and Frank, Sue, and Ann must approve the expense report.

## Benefits of using the workflow system

There are several benefits of using the workflow system in your organization:

- **Consistent processes** – You can define how specific documents, such as purchase requisitions and expense reports, are processed. By using the workflow system, you ensure that documents are processed and approved in a consistent and efficient manner.
- **Process visibility** – You can track the status, history, and performance metrics of workflow instances. This helps you determine whether changes should be made to the workflow to improve efficiency.
- **Centralized work list** – Users can view a centralized work list that displays the workflow tasks and approvals that are assigned to them.

## Workflow content

- [Workflow system architecture](#)
- [Workflow elements](#)
- [Actions in workflow approval processes](#)
- [Create workflows overview](#)
- [Configure workflow properties](#)
- [Configure manual tasks in a workflow](#)
- [Configure automated tasks in a workflow](#)
- [Configure approval processes in a workflow](#)
- [Configure approval steps in a workflow](#)
- [Configure manual decisions in a workflow](#)
- [Configure conditional decisions in a workflow](#)
- [Configure parallel activities in a workflow](#)
- [Configure parallel branches in a workflow](#)
- [Configure line-item workflows](#)
- [Workflow FAQ](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Workflow system architecture

2/18/2021 • 3 minutes to read • [Edit Online](#)

This article describes the architecture of the workflow system.

The workflow infrastructure consists of two components that are hosted on Application Object Server (AOS): the X++ workflow runtime and the managed workflow runtime.

The X++ workflow runtime consists of the following components:

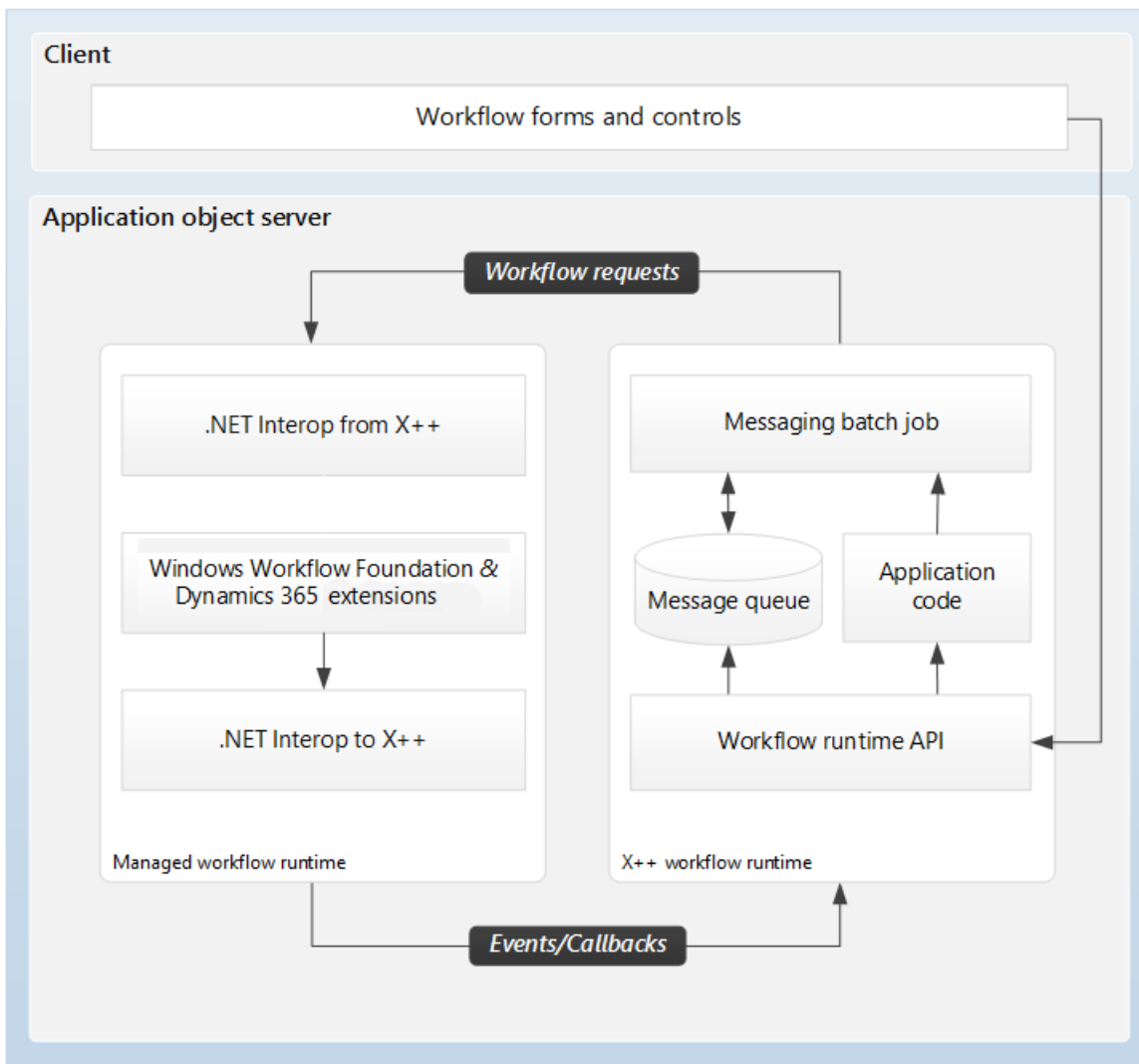
- Workflow runtime application programming interface (API)
- A messaging batch job
- A message queue

Either the messaging batch job or the workflow runtime API can invoke the application code, if it's required. The X++ workflow runtime is compiled into the Common Intermediate Language (CIL) of the Microsoft .NET Framework.

The managed workflow runtime consists of the Windows Workflow Foundation and Finance and Operations apps extensions.

Logically, the workflow infrastructure is an extension and is transparent to users. Physically, both the X++ workflow and the managed workflow runtimes are hosted on AOS. The workflow infrastructure uses batch processing on AOS and .NET Interop to integrate both subsystems, and to pass messages from one subsystem to the other. The X++ code that is run in the batch processor is compiled to .NET CIL. The batch processing runs in the .NET common language runtime (CLR).

The following figure shows the high-level architecture of the workflow infrastructure.



Users can use workflow pages and controls to participate in business processes.

Developers can create workflows for the objects that they have added. The following table describes the workflow steps that occur when a user submits an expense report to the workflow system for approval.

STEP	RUNTIME	ACTIVITY
1	X++ workflow runtime	<p>A user submits an expense report by clicking the <b>Submit</b> button on one of the workflow controls. This action causes X++ code to activate a workflow instance by calling the workflow runtime API. The workflow runtime API posts a message to the message queue. The messaging batch job reads the message and sends a workflow activation request to the managed workflow runtime.</p> <p>[!NOTE] The messaging batch job processes the message queue at one-minute intervals.</p>

STEP	RUNTIME	ACTIVITY
2	Managed workflow runtime	<p>.NET Interop from X++ receives the message and starts a new workflow instance via Windows Workflow Foundation. This workflow instance performs a callback to the X++ workflow runtime API via .NET Interop to X++ CIL and posts a message that the workflow has started.</p> <p>After the message is posted, the managed workflow runtime saves the idle workflow instance to the database. The runtime then removes the workflow instance from memory. When the managed workflow runtime receives another message from the X++ workflow runtime for this workflow instance, it restores the workflow instance to memory and resumes it.</p> <p>Each workflow instance is unique. If two users submit their expense reports for approval, two workflow instances are started.</p>
3	X++ workflow runtime	<p>The messaging batch job reads the <b>workflow started</b> message from the message queue and invokes the application event handler to process a <b>workflow started</b> event. The batch job then posts an acknowledgment message that the event was processed.</p>
4	Both	<p>This same messaging pattern is repeated, as required, throughout the lifecycle of the workflow instance.</p>

The workflow architecture helps provide a reliable and durable messaging system, and also helps guarantee that the state of the workflow is always synchronized with the state of the application. If an unexpected hardware or software failure occurs, the workflow instance state is returned to its last known saved point, and the message stays in the queue. Therefore, from an architecture perspective, the recovery model is to fix the problem and resume the workflow.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Workflow elements

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the various elements that make up a workflow.

A workflow consists of elements. The sections that follow describe each type of element.

## Tasks

A *task* is a unit of work that must be performed. Two types of tasks can be added to a workflow: manual tasks and automated tasks.

### Manual task

A *manual task* is a unit of work that must be performed by a user. For example, an expense report workflow can have manual tasks that require the assigned users to complete the following actions:

- Review the receipts that are submitted together with an expense report.
- Call an employee's manager.

### Automated task

An *automated task* is a unit of work that must be performed by the system. No human interaction is required. For example, a sales order workflow can have automated tasks that require the system to complete the following actions:

- Perform a credit check.
- Create a customer record for the customer, if a record doesn't already exist.

## Approval processes

An *approval process* is a process that consists of separate steps. At each approval step, the user can perform the following actions:

- Approve the document.
- Reject the document.
- Request a change to the document.
- Assign the document to another user for approval.

## Line-item workflow elements

A workflow can be created to process either documents or the line items on a document. For example, you've created an approval workflow for timesheets. (We will refer to this workflow as the *document workflow*.) You can add a *line-item workflow* element to that document workflow. When the line-item element is run, each line item on the document is submitted for processing. You might want all the line items to be processed by the same line-item workflow, or you might want each line item to be processed by a different line-item workflow. Imagine that an employee has submitted a timesheet that resembles the following figure.

Timesheet: 1234  
Employee: 5678  
Submitted by: Mike

**Your Organization's Name**

Date	Project ID	Project Manager	Description of Work	Start Time	Stop Time
DD/MM/YY	1111	Mary	Work at site A	9:00	17:00
DD/MM/YY	1111	Mary	Work at site A	9:00	17:00
DD/MM/YY	2222	Hans	Work at site B	9:00	17:00
DD/MM/YY	3333	Jen	Work at site C	9:00	17:00
DD/MM/YY	3333	Jen	Work at site C	9:00	17:00

**Document workflow**  
You can configure a workflow to process the whole document.

**Line-item workflow**  
You can configure a workflow to process each line item on the document.

In this scenario, you might want to create the following line-item workflows:

- **Line-item workflow 1** – This workflow is used to process line items where the project ID is 1111.
- **Line-item workflow 2** – This workflow is used to process line items where the project ID is 2222.
- **Line-item workflow 3** – This workflow is used to process line items where the project ID is 3333.

## Flow-control elements

The following elements let you design workflows that have alternate branches or branches that run at the same time.

### Manual decision

A *manual decision* is a point where a workflow divides into two branches. A user must make a decision, and this decision determines which branch is used to process the document that was submitted.

### Conditional decision

A *conditional decision* is also a point where a workflow divides into two branches. However, the system decides which branch is used to process the document that was submitted. To make this decision, the system evaluates the document to determine whether it meets specified conditions.

### Parallel activity

A *parallel activity* is a workflow element that includes two or more workflow branches that run at the same time.

### Subworkflow

A *subworkflow* is a workflow that runs in the context of another workflow.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

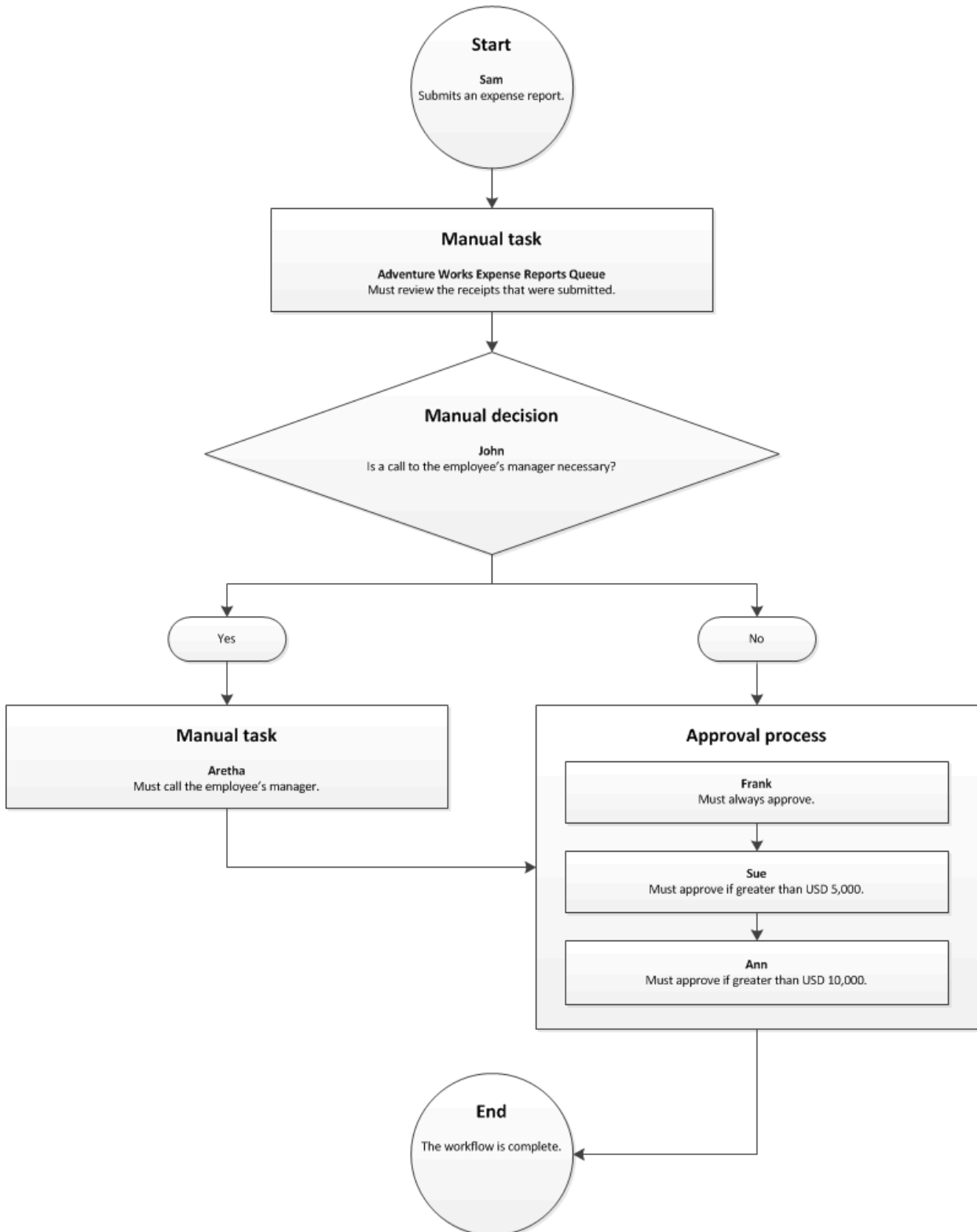
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Actions in workflow approval processes

2/18/2021 • 6 minutes to read • [Edit Online](#)

This article explains the actions that each participant in a workflow approval process can take.

A workflow can involve several groups of people: the originator, task assignees, decision makers, and approvers. For example, in the following expense report workflow, Sam is the originator, the members of the queue are task assignees, John is a decision maker, and Frank, Sue, and Ann are approvers.



The following sections explain the workflow actions that each group can perform.

## Actions that an originator can perform

The originator starts a workflow instance by submitting a document for processing. For example, Sam must click the **Submit** button on the **Expense report** page to submit his expense report.

## Actions that a task assignee can perform

A task can be assigned to multiple people or to a work item queue that is monitored by several people. However, only one person can complete a task. For example, Sam has submitted an expense report and has routed his receipts to his organization's Expense Reports department for review.

The members of the Adventure Works Expense Reports department monitor the queue. Julie, a member of that department, has accepted the task of reviewing Sam's expense report and receipts. She can now perform one of the following actions: complete, reject, delegate, request change, reassign, or release.

### NOTE

The actions that are available vary, depending on how the software developer designed the task.

### Complete

When a user completes a task, the document that was submitted for processing is assigned to the next user in the workflow, if there is a next user. If no additional processing is required, the workflow process ends.

For example, Julie, a member of the Adventure Works Expense Reports department, has accepted the task of reviewing Sam's expense report and receipts. After Julie completes her review, the document is assigned to John.

### Reject

When a user rejects a document, the workflow process ends.

For example, Julie, a member of the Adventure Works Expense Reports department, has accepted the task of reviewing Sam's expense report and receipts. If Julie rejects the expense report, the workflow process ends.

Sam can then resubmit the expense report. He can make changes first, or he can resubmit the original version. If Sam resubmits the expense report, the workflow process starts at the manual review task.

### Delegate

When a user delegates a task, the task is assigned to another user.

For example, Julie, a member of the Adventure Works Expense Reports department, has accepted the task of reviewing Sam's expense report and receipts. Julie delegates this task to Tim, who is her assistant.

Tim then acts on behalf of Julie. Therefore, when Tim completes his review, the expense report is assigned to John, just as if Julie had completed the task.

### Request change

When a user requests a change to a document that was submitted, the document is sent back to the originator.

For example, Julie, a member of the Adventure Works Expense Reports department, has accepted the task of reviewing Sam's expense report and receipts. Julie notices some errors on the expense report and requests changes. The expense report is sent back to Sam.

Sam can resubmit the expense report. He can make the requested changes first, or he can resubmit the original version. If Sam resubmits the expense report, a member of the work item queue must review the expense report

and the receipts again.

### **Reassign**

The members of a work item queue can reassign documents that are in that queue to another queue.

For example, Julie, a member of the Adventure Works Expense Reports department, is monitoring the queue. To help balance the workload, she can reassign the expense report, and the receipts that are included with it, to another queue.

### **Release**

Occasionally, a member of a work item queue might accept a task, but then decide that he or she can't complete the task. In this case, the person who accepted the task can release the document back to the work item queue.

For example, Julie, a member of the Adventure Works Expense Reports department, has accepted the task of reviewing Sam's expense report and receipts. If Julie decides that she can't complete the task, she can release the document. The expense report is returned to the queue, so that other members of the Adventure Works Expense Reports department can complete the task.

## **Actions that a decision maker can perform**

Typically, a document is assigned to a decision maker, because there is a question that the decision maker must answer. The answer to the question is typically **Yes** or **No**, or **True** or **False**. If the decision maker doesn't select one of those choices, he or she can delegate the decision.

### **[Choice 1] or [Choice 2]**

A decision maker must answer a question that is related to the document. The answer to the question is typically **Yes** or **No**, or **True** or **False**. The answer that the decision maker selects determines the workflow branch that is used to process the document.

For example, Sam's expense report is assigned to John. John must decide whether the information in the document requires a call to Sam's manager. If John decides that a call is required, the expense report is assigned to Aretha, who must then call Sam's manager. If John decides that a call isn't required, the expense report is assigned to Frank for approval.

### **Delegate**

When a decision maker delegates a decision, the document is assigned to another user who must make the decision.

For example, Sam's expense report is assigned to John. John delegates the decision to Maria, who is his assistant.

Maria then acts on behalf of John. If Maria decides that a call to Sam's manager is required, the expense report is assigned to Aretha, who must then call Sam's manager. If Maria decides that a call isn't required, the expense report is assigned to Frank for approval.

## **Actions that an approver can perform**

When a document is assigned to an approver, the approver can perform one of the following actions: approve, reject, delegate, or request change.

### **Approve**

When an approver approves a document, the document is assigned to the next user in the workflow, if there is a next user. If no additional processing is required, the workflow process ends.

For example, Sam has submitted an expense report for USD 6,000, and this document is assigned to Frank. When Frank approves the document, it's assigned to Sue for approval. When Sue approves the expense report,

the workflow process ends.

### **Reject**

When an approver rejects a document, the workflow process ends.

For example, Sam has submitted an expense report for USD 12,000, and this document is assigned to Sue. If Sue rejects the expense report, the workflow process ends.

Sam can resubmit the expense report. He can make changes first, or he can resubmit the original version of the expense report. If Sam resubmits the expense report, the workflow process starts at the approval process.

### **Delegate**

When an approver delegates a document, the document is assigned to another user for approval.

For example, Sam has submitted an expense report for USD 12,000, and this document is assigned to Frank. Frank delegates the expense report to Ann.

Ann then acts on behalf of Frank. Therefore, when Ann approves the document, it's assigned to Sue for approval, just as if Frank had approved it. After Sue approves the document, it's sent to Ann for approval.

### **Request change**

When an approver requests a change to a document, the document is sent back to the originator.

For example, Sam has submitted an expense report for USD 12,000, and this document is assigned to Sue. If Sue requests a change, the expense report is sent back to Sam.

Sam can resubmit the expense report. He can make the requested changes first, or he can resubmit the original version of the expense report. If Sam resubmits the expense report, it's sent to Frank for approval, because Frank is the first approver in the approval process.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create workflows overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to create a workflow.

## Open the workflow editor

The module that you're working in determines the types of workflow that you can create. Follow these steps to select the type of workflow to create and open the workflow editor.

1. Open the module that you want to create a new workflow for. For example, to create a workflow for purchase requisitions, click **Procurement and sourcing**.
2. Click **Setup > [Module name] workflows**.
3. On the list page that appears, on the Action Pane, click **New**.
4. On the **Create workflow** page, select the type of workflow to create, and then click **Create workflow**. The workflow editor appears. You can now use the following procedures to design the workflow.

## Drag workflow elements onto the canvas

The **Workflow elements** area of the workflow editor contains the elements that you can add to your workflow. To add elements to the workflow, drag them onto the canvas.

## Connect the elements

To connect one workflow element to another, hold the pointer over an element until connection points appear. Click a connection point, and drag it to another element. Be sure to connect all the elements.

## Configure the properties of the workflow

Follow these steps to configure the properties of the workflow.

1. Click the canvas to make sure that no workflow element is selected.
2. Click **Properties** to open the **Properties** page for the workflow.
3. Follow the procedures in the [Configure workflow properties](#) topic.

## Configure the elements of the workflow

Configure each element that you dragged onto the canvas. For information about how to configure each workflow element, see the following topics:

- [Configure manual tasks in a workflow](#)
- [Configure automated tasks in a workflow](#)
- [Configure approval processes in a workflow](#)
- [Configure approval steps in a workflow](#)
- [Configure manual decisions in a workflow](#)
- [Configure conditional decisions in a workflow](#)
- [Configure parallel branches in a workflow](#)
- [Configure a parallel branch](#)
- [Configure line-item workflows](#)

## Resolve any errors or warnings

The **Errors and warnings** pane at the bottom of the workflow editor shows messages that have been generated for the workflow. To find the element where an error or warning occurred, double-click the error or warning message. You must resolve all errors and warnings before you can make the workflow active.

## Save and activate the workflow

When you're ready to save and activate the workflow, follow these steps.

1. Click **Save and close** to close the workflow editor and open the **Save workflow** page.
2. Enter comments about the changes that you've made to the workflow, and then click **OK**.
3. If all errors and warnings have been resolved, the **Activate workflow** page appears. Select one of the following options:
  - To activate this version of the workflow, click **Activate the new version**. When a workflow is active, users can submit documents to it for processing.
  - If you don't want to activate this version, click **Do not activate the new version**. You can activate the workflow later.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure workflow properties

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how to configure the various properties of a workflow.

To configure the properties of a workflow, open the workflow in the workflow editor. Click the canvas of the workflow editor, and then click **Properties** to open the **Properties** page. You can then use the following procedures to configure the various properties of the workflow.

## Name the workflow

Follow these steps to enter a name for the workflow.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the workflow. For example, if you create a purchase requisition workflow for each country/region that you operate in, you can name the purchase requisition workflow **Purchase Requisitions Denmark** or **Purchase Requisitions Spain**.

## Specify the workflow owner

The workflow owner is the person who manages and maintains the workflow. Follow these steps to specify the workflow owner.

1. In the left pane, click **Basic Settings**.
2. In the **Owner** list, select the name of the person who will manage the workflow.

## Select an email template

Follow these steps to select the email template that is used to generate notification messages about the workflow.

1. In the left pane, click **Basic Settings**.
2. In the **Email template for workflow notifications** list, select the template.

## Enter instructions for users

You can provide instructions to users who submit documents for processing and approval. These users are also referred to as *originators*. For example, you're creating a purchase requisition workflow, and you enter instructions. Those instructions can then be viewed by users who enter purchase requisitions on the **Purchase requisitions** page. To view instructions, the originator clicks the icon in the workflow message bar. Follow these steps to enter instructions for users.

1. In the left pane, click **Basic Settings**.
2. In the **Submission instructions** field, enter the instructions.
3. To personalize the instructions, you can insert placeholders. Placeholders are replaced with the appropriate data when the instructions are shown to users. To insert a placeholder, follow these steps:
  - a. Click in the **Submission instructions** field to specify where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.

- d. Click **Insert**.
4. To add translations of the instructions, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you will enter the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders. For instructions about how to enter a placeholder, see step 3.
  - f. Click **Close**.

#### NOTE

Placeholders cannot be added using copy and paste because the target information is not pasted in correctly. Use the interface to add placeholders.

## Specify when this workflow is used through activation conditions

You can create multiple workflows that are based on the same workflow type. When you have multiple workflows that are based on the same type, you must specify when each workflow is used using activation conditions. If activation conditions are not met, then the default workflow is used. Similarly, if there is only one workflow configuration defined for a workflow type, then that workflow configuration will be used regardless of the activation conditions.

For example, you can create a purchase requisition workflow for each country/region that you operate in, such as Purchase Requisitions Denmark and Purchase Requisitions Spain, with the following conditions:

- Purchase Requisitions Denmark is used when: country/region = DK
- Purchase Requisitions Spain is used when: country/region = ES

Follow these steps to specify when the workflow that you're configuring is used.

1. In the left pane, click **Activation**.
2. Select the **Set the conditions for running this workflow** check box.
3. Click **Add condition**.
4. Enter a condition.
5. Enter any additional conditions that are required.
6. Run through the workflow with some target records to verify that the condition correctly includes and excludes records.

## Specify when notifications are sent

When a document is submitted for processing, a workflow instance is created. You can send notifications to users when workflow instances that are based on the workflow are started, completed, terminated, or stopped because of an error. Follow these steps to specify when notifications are sent.

1. In the left pane, click **Notifications**.
2. Select the check box for each event that should trigger notifications:
  - **Started** – Send notifications when a workflow instance is started.
  - **Stopped** – Send notifications when a workflow instance is stopped because of an error.
  - **Completed** – Send notifications when a workflow instance is completed.

- **Unrecoverable** – Send notifications when a workflow instance is stopped because of an unrecoverable error.
  - **Terminated** – Send notifications when a workflow instance is terminated.
3. Select the row for an event that you selected in step 2.
  4. On the **Notification text** tab, enter the text of the notification.
  5. To personalize the text, you can insert placeholders. Placeholders are replaced with the appropriate data when the text is shown to users. To insert a placeholder, follow these steps:
    - a. Click in the field to specify where the placeholder should appear.
    - b. Click **Insert placeholder**.
    - c. In the list that appears, select the placeholder to insert.
    - d. Click **Insert**.
    - e. A common **Notification text** placeholder to include is "Last Notes: %Workflow.Last note%", which displays any comments from the previous step.
  6. To add translations of the text, follow these steps:
    - a. Click **Translations**.
    - b. On the page that appears, Click **Add**.
    - c. In the list that appears, select the language that you will enter the text in.
    - d. In the **Translated text** field, enter the text.
    - e. To personalize the text, you can insert placeholders. For instructions about how to enter a placeholder, see step 5.
    - f. Click **Close**.
  7. On the **Recipient** tab, use the following options to specify who should receive the notifications.

OPTION	NOTIFICATIONS ARE SENT TO THESE USERS	TO SEND NOTIFICATIONS, FOLLOW THESE STEPS
Participant	Users who are assigned to a specific group or role	<ol style="list-style-type: none"> <li>1. On the <b>Recipient</b> tab, click <b>Participant</b>.</li> <li>2. On the <b>Role based</b> tab, in the <b>Type of participant</b> list, select the type of group or role to send notifications to.</li> <li>3. In the <b>Participant</b> list, select the group or role to send notifications to.</li> </ol>
Workflow user	Users who are participants in this workflow	<ol style="list-style-type: none"> <li>1. On the <b>Recipient</b> tab, click <b>Workflow user</b>.</li> <li>2. On the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a participant in this workflow.</li> </ol>
User	Specific users	<ol style="list-style-type: none"> <li>1. On the <b>Recipient</b> tab, click <b>User</b>.</li> <li>2. On the <b>User</b> tab, the <b>Available users</b> list includes all users. Select the users to send notifications to, and move those users into the <b>Selected users</b> list.</li> </ol>

8. Repeat steps 3 through 7 for each event that you selected in step 2.

## Enter comments about the changes that you made to the workflow

To enter comments about the changes that you made to the workflow, follow these steps.

1. In the left pane, click **Notes**.
2. In the **Enter comments about the workflow** field, enter your comments.
3. Review your comments. After you add comments, you can't modify them.
4. Click **Add** to add your comments to the **Comment history** area.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure manual tasks in a workflow

2/18/2021 • 14 minutes to read • [Edit Online](#)

This topic explains how to configure the properties for a manual task.

To configure a manual task in the workflow editor, right-click the task, and then click **Properties** to open the **Properties** page. Then use the following procedures to configure the properties for the manual task.

## Name the task

Follow these steps to enter a name for the manual task.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the task.

## Enter a subject line and instructions

You must provide a subject line and instructions to users who are assigned to the task. For example, if you're configuring a task for purchase requisitions, the user who is assigned to the task sees the subject line and instructions on the **Purchase requisitions** page. The subject line appears in a message bar on the page. The user can then click the icon in the message bar to view the instructions. Follow these steps to enter a subject line and instructions.

1. In the left pane, click **Basic Settings**.
2. In the **Work item subject** field, enter the subject line.
3. To personalize the subject line, you can insert placeholders. Placeholders are replaced with appropriate data when the subject line is shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.
4. To add translations of the subject line, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 3.
  - f. Click **Close**.
5. In the **Work item instructions** field, enter the instructions.
6. To personalize the instructions, you can insert placeholders. Placeholders are replaced with appropriate data when the instructions are shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.

7. To add translations of the instructions, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 6.
  - f. Click **Close**.

## Assign the task

Follow these steps to specify who the manual task should be assigned to.

1. In the left pane, click **Assignment**.
2. On the **Assignment type** tab, select one of the options in the following table, and then follow the additional steps for that option before you go to step 3.

OPTION	USERS THAT THE TASK IS ASSIGNED TO	ADDITIONAL STEPS
Participant	Users who are assigned to a specific group or role	<ol style="list-style-type: none"> <li>1. After you select <b>Participant</b>, on the <b>Role based</b> tab, in the <b>Type of participant</b> list, select the type of group or role to assign the task to.</li> <li>2. In the <b>Participant</b> list, select the group or role to assign the task to.</li> </ol>

OPTION	USERS THAT THE TASK IS ASSIGNED TO	ADDITIONAL STEPS
Hierarchy	Users in a specific organizational hierarchy	<ol style="list-style-type: none"> <li>1. After you select <b>Hierarchy</b>, on the <b>Hierarchy selection</b> tab, in the <b>Hierarchy type</b> list, select the type of hierarchy to assign the task to.</li> <li>2. The system must retrieve a range of user names from the hierarchy. These names represent users that the task can be assigned to. Follow these steps to specify the starting point and ending point of the range of user names that the system retrieves: <ol style="list-style-type: none"> <li>a. To specify the starting point, select a person in the <b>Start from</b> list.</li> <li>b. To specify the ending point, click <b>Add condition</b>. Then enter a condition that determines where in the hierarchy the system stops retrieving names.</li> </ol> </li> <li>3. On the <b>Hierarchy options</b> tab, specify which users in the range the task should be assigned to: <ul style="list-style-type: none"> <li>• <b>Assign to all users retrieved</b> – The task is assigned to all users in the range.</li> <li>• <b>Assign only to last user retrieved</b> – The task is assigned to only the last user in the range.</li> <li>• <b>Exclude users with the following condition</b> – The task isn't assigned to users in the range who meet a specific condition. Click <b>Add condition</b> to specify the condition.</li> </ul> </li> </ol>
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>

OPTION	USERS THAT THE TASK IS ASSIGNED TO	ADDITIONAL STEPS
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all users. Select the users to assign the task to, and then move those users to the <b>Selected users</b> list.</li> </ol>
Queue	A work item queue	<ol style="list-style-type: none"> <li>1. After you select <b>Queue</b>, click the <b>Queue based</b> tab.</li> <li>2. To assign the task to a specific queue, follow these steps: <ol style="list-style-type: none"> <li>a. In the <b>Queue type</b> list, select <b>Work item queues</b>.</li> <li>b. In the <b>Queue name</b> list, select the queue.</li> </ol> </li> <li>3. If a specific condition should determine which queue the task is assigned to, follow these steps: <ol style="list-style-type: none"> <li>a. In the <b>Queue type</b> list, select <b>Conditional work item queues</b>.</li> <li>b. In the <b>Queue name</b> list, select <b>Conditional queue</b>.</li> </ol> </li> </ol> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>[!NOTE] This option is used for only a few workflows, such as Case management.</p> </div>

3. On the **Time limit** tab, in the **Duration** field, specify how much time the user has to complete the task. Select one of the following options:

- **Hours** – Enter the number of hours that the user has to complete the task. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Days** – Enter the number of days that the user has to complete the task. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Weeks** – Enter the number of weeks that the user has to complete the task.
- **Months** – Select the day and week that the user must complete the task by. For example, you might want the user to complete the task by Friday of the third week of the month.
- **Years** – Select the day, week, and month that the user must complete the task by. For example, you might want the user to complete the task by Friday of the third week of December.

If the user doesn't complete the task in the allotted time, the task is overdue. A task that is overdue can be escalated, based on the options that you select in the **Escalation** area of the page.

## Specify what happens when the task is overdue

If a user doesn't complete the manual task in the allotted time, the task is overdue. A task that is overdue can be escalated, or automatically assigned to another user. Follow these steps to escalate the task if it's overdue.



1. In the left pane, click **Escalation**.
2. Select the **Use escalation path** check box to create an escalation path. The system automatically assigns the task to the users who are listed in the escalation path. For example, the following table represents an escalation path.

SEQUENCE	ESCALATION PATH
1	Assign to: Donna
2	Assign to: Erin
3	Final action: Reject

In this example, the system assigns the overdue task to Donna. If Donna doesn't complete the task in the allotted time, the system assigns the task to Erin. If Erin doesn't complete the task in the allotted time, the system rejects the document that was submitted for processing.

3. To add a user to the escalation path, click **Add escalation**. On the **Assignment type** tab, select one of the options in the following table, and then follow the additional steps for that option before you go to step 4.

OPTION	USERS THAT THE TASK IS ESCALATED TO	ADDITIONAL STEPS
--------	-------------------------------------	------------------

OPTION	USERS THAT THE TASK IS ESCALATED TO	ADDITIONAL STEPS
Hierarchy	Users in a specific organizational hierarchy	<ol style="list-style-type: none"> <li>1. After you select <b>Hierarchy</b>, on the <b>Hierarchy selection</b> tab, in the <b>Hierarchy type</b> list, select the type of hierarchy to escalate the task to.</li> <li>2. The system must retrieve a range of user names from the hierarchy. These names represent users that the task can be escalated to. Follow these steps to specify the starting point and ending point of the range of user names that the system retrieves: <ol style="list-style-type: none"> <li>a. To specify the starting point, select a person in the <b>Start from</b> list.</li> <li>b. To specify the ending point, click <b>Add condition</b>. Then enter a condition that determines where in the hierarchy the system stops retrieving names.</li> </ol> </li> <li>3. On the <b>Hierarchy options</b> tab, specify which users in the range the task should be escalated to: <ul style="list-style-type: none"> <li>• <b>Assign to all users retrieved</b> – The task is escalated to all users in the range.</li> <li>• <b>Assign only to last user retrieved</b> – The task is escalated to only the last user in the range.</li> <li>• <b>Exclude users with the following condition</b> – This task isn't escalated to users in the range who meet a specific condition. Click <b>Add condition</b> to specify the condition.</li> </ul> </li> </ol>
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>

OPTION	USERS THAT THE TASK IS ESCALATED TO	ADDITIONAL STEPS
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all users. Select the users to escalate the task to, and then move those users to the <b>Selected users</b> list.</li> </ol>

4. On the **Time limit** tab, in the **Duration** field, specify how much time the user has to complete the task. Select one of the following options:

- **Hours** – Enter the number of hours that the user has to complete the task. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Days** – Enter the number of days that the user has to complete the task. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Weeks** – Enter the number of weeks that the user has to complete the task.
- **Months** – Select the day and week that the user must complete the task by. For example, you might want the user to complete the task by Friday of the third week of the month.
- **Years** – Select the day, week, and month that the user must complete the task by. For example, you might want the user to complete the task by Friday of the third week of December.

5. Repeat steps 3 through 4 for each user that should be added to the escalation path. You can change the order of the users.

6. If the users in the escalation path don't complete the task in the allotted time, the system takes action on the task. To specify the action that the system takes, select the **Action** row, and then, on the **End action** tab, select an action.

## Specify when the system automatically acts on the task

You can configure the system to take action on the manual task if specific conditions are met. For example, a task requires that a member of the Expense reports department review the receipts that are submitted together with an expense report. According to company policy, this task must be performed if the total amount of the expense report is more than USD 100. In this scenario, you can configure the system to automatically mark the task as **Complete** when the total amount is less than 100. Follow these steps to specify when the system takes action on the manual task.

1. In the left pane, click **Automatic actions**.
2. Select the **Enable automatic actions** check box.
3. Click **Add condition**.
4. Enter a condition.
5. Enter any additional conditions that are required.
6. To verify that the conditions that you entered are configured correctly, follow these steps:
  - a. Click **Test**.
  - b. On the **Test workflow condition** page, in the **Validate condition** area, select a record.
  - c. Click **Test**. The system evaluates the record to determine whether it meets the conditions that you defined.
  - d. Click **OK** or **Cancel** to return to the **Properties** page.

7. In the **Auto complete action** list, select the action that the system should take on the task.

## Specify when notifications are sent

You can send notifications to people when a manual task has been delegated, escalated, completed, or rejected, or when a change has been requested. Follow these steps to specify when notifications are sent, and who the notifications are sent to.

1. In the left pane, click **Notifications**.
2. Select the check box next to the events that notifications should be sent for:
  - **Delegate** – The task has been assigned to another user.
  - **Escalate** – The assigned user hasn't completed the task in the allotted time.
  - **Complete** – The assigned user has completed the task.
  - **Reject** – The assigned user has rejected the document that was submitted.
  - **Request change** – The assigned user has requested a change to the document that was submitted.
3. Select the row for an event that you selected in step 2.
4. On the **Notification text** tab, in the text box, enter the text of the notification.
5. To personalize the notification, you can insert placeholders. Placeholders are replaced with appropriate information when the notification is shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.
6. To add translations of the notification, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 5.
  - f. Click **Close**.
7. On the **Recipient** tab, specify who the notifications are sent to. Select one of the options in the following table, and then follow the additional steps for that option before you go to step 8.

OPTION	NOTIFICATION RECIPIENTS	ADDITIONAL STEPS
Participant	Users who are assigned to a specific group or role	<ol style="list-style-type: none"><li>1. After you select <b>Participant</b>, on the <b>Role based</b> tab, in the <b>Type of participant</b> list, select the type of group or role to send notifications to.</li><li>2. In the <b>Participant</b> list, select the group or role to send notifications to.</li></ol>

OPTION	NOTIFICATION RECIPIENTS	ADDITIONAL STEPS
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>
User	Specific users	<ol style="list-style-type: none"> <li>After you select <b>User</b>, click the <b>User</b> tab.</li> <li>The <b>Available users</b> list includes all users. Select the users to send notifications to, and then move those users to the <b>Selected users</b> list.</li> </ol>

8. Repeat steps 3 through 7 for each event that you selected in step 2.

## Set a time limit

Follow these steps if the manual task must be completed in a specific time.

### NOTE

The options that you select in this procedure override the options that you selected in the **Assignment** and **Escalation** areas of the page.

- In the left pane, click **Advanced settings**.
- Select the **Set a time limit for the workflow element** check box.
- In the **Duration** field, specify when the task must be completed. Select one of the following options:
  - Hours** – Enter the number of hours that the task must be completed in. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - Days** – Enter the number of days that the task must be completed in. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - Weeks** – Enter the number of weeks that the task must be completed in.
  - Months** – Select the day and week that the task must be completed by. For example, you might want the task to be completed by Friday of the third week of the month.
  - Years** – Select the day, week, and month that the task must be completed by. For example, you might want the task to be completed by Friday of the third week of December.
- If the time limit is exceeded, the system takes action on the task. In the **Action** list, select the action that the system should take.

## Specify which actions are available to the user

When the manual task is assigned to a user, the user must take action on the task. Follow these steps to specify which actions the user can take on the task.

### NOTE

The actions that are available vary, depending on the design of the task.

1. In the left pane, click **Advanced settings**.
2. Select the **Complete** check box if the user should be able to mark the task as **Complete**.
3. Select the **Reject** check box if the user should be able to reject the document that was submitted.
4. Select the **Request change** check box if the user should be able to request changes to the document that was submitted.
5. Select the **Delegate** check box if the user should be able to assign the task to another user.
6. Select the **Reassign** check box if the user should be able to reassign the task to another user in the work item queue.
7. Select the **Release** check box if the user should be able to reassign the task to the work item queue. Another user can then complete the task.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure automated tasks in a workflow

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to configure the properties for an automated task.

To configure an automated task in the workflow editor, right-click the task, and then click **Properties** to open the **Properties** page. Then use the following procedures to configure the properties for the automated task.

## Name the task

Follow these steps to enter a name for the automated task.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the task.

## Specify when notifications are sent

You can send notifications to people when an automated task has been run or canceled. Follow these steps to specify when notifications are sent, and who they are sent to.

1. In the left pane, click **Notifications**.
2. Select the check box next to the events to send notifications for:
  - **Execution** – Notifications are sent when the task has been run.
  - **Canceled** – Notifications are sent when the task has been canceled.
3. Select the row for an event that you selected in step 2.
4. On the **Notification text** tab, in the text box, enter the text of the notification.
5. To personalize the notification, you can insert placeholders. Placeholders are replaced with appropriate data when the notification is shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.
6. To add translations of the notification, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 5.
  - f. Click **Close**.
7. On the **Recipient** tab, specify who the notifications are sent to. Select one of the options in the following table, and then follow the additional steps for that option before you go to step 8.

OPTION	NOTIFICATION RECIPIENTS	ADDITIONAL STEPS
--------	-------------------------	------------------

OPTION	NOTIFICATION RECIPIENTS	ADDITIONAL STEPS
Participant	Users who are assigned to a specific group or role	<ol style="list-style-type: none"> <li>1. After you select <b>Participant</b>, on the <b>Role based</b> tab, in the <b>Type of participant</b> list, select the type of group or role to send notifications to.</li> <li>2. In the <b>Participant</b> list, select the group or role to send notifications to.</li> </ol>
Workflow user	Users who participate in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all users. Select the users to send notifications to, and then move those users to the <b>Selected users</b> list.</li> </ol>

8. Repeat steps 3 through 7 for each event that you selected in step 2.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure approval processes in a workflow

2/18/2021 • 6 minutes to read • [Edit Online](#)

Use the following procedure to configure the properties of the approval process.

To configure an approval process, in the workflow editor, right-click the approval element, and then click **Properties** to open the **Properties** form.

## Name the approval process

Follow these steps to enter a name for the approval process.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the approval process.

## Specify when the system automatically acts on the document

You can configure the system to automatically act on the document if specific conditions are met. For example, the system can approve expense reports that have total amounts that are less than USD 100. Follow these steps to specify when the system acts on the document.

1. In the left pane, click **Automatic actions**.
2. Select the **Enable automatic actions** check box.
3. Click **Add condition**.
4. Enter a condition.
5. Enter additional conditions, if necessary.
6. To verify that the conditions that you entered are configured correctly, complete the following steps:
  - a. Click **Test** to open the **Test workflow condition** form.
  - b. Select a record in the **Validate condition** area of the form.
  - c. Click **Test**. The system evaluates the record to determine whether it meets the conditions that you defined.
  - d. Click **OK** or **Cancel** to return to the **Properties** form.
7. In the **Auto complete action** list, select the action that the system should take on the document.

## Specify when notifications are sent

You can send notifications to people when a document has been approved, rejected, delegated, or escalated, or when a change has been requested. Follow these steps to specify when notifications are sent, and who the notifications are sent to.

1. In the left pane, click **Notifications**.
2. Select the check box next to the events to send notifications for:
  - **Delegate** – When a document has been assigned to another user for approval.
  - **Escalate** – When the assigned user has not acted on a document in the allotted time.
  - **Approve** – When a document has been approved.

- **Reject** – When a document has been rejected.
  - **Request change** – When the assigned user has requested a change to a document that was submitted.
3. Select the row for an event that you selected in step 2.
  4. Click the **Notification text** tab.
  5. In the text box, enter the text for the notification.
  6. To personalize the text, you can insert placeholders, which are replaced with the appropriate data when they are displayed to users. To insert a placeholder, follow these steps:
    - a. Click in the text box at the location where the placeholder should appear.
    - b. Click **Insert placeholder**.
    - c. In the list that is displayed, select the placeholder to insert.
    - d. Click **Insert**.
  7. To add translations of the notification, click **Translations**. In the form that is displayed, follow these steps:
    - a. Click **Add**.
    - b. In the list that is displayed, select the language in which you will enter the text.
    - c. In the **Translated text** text box, enter the text.
    - d. To personalize the text, insert placeholders.
    - e. Click **Close**.
  8. Click the **Recipient** tab.
  9. Specify who the notifications are sent to. Select one of the options in the following table, and then follow the additional steps for the option before you go to step 10.

OPTION	NOTIFICATION RECIPIENTS	ADDITIONAL STEPS
<b>Participant</b>	Users who are assigned to a specific group or role	<ol style="list-style-type: none"> <li>1. After you select <b>Participant</b>, click the <b>Role based</b> tab.</li> <li>2. In the <b>Type of participant</b> list, select the type of group or role to send notifications to.</li> <li>3. In the <b>Participant</b> list, select the group or role to send notifications to.</li> </ol>
<b>Workflow user</b>	Users who participate in the current workflow	<ol style="list-style-type: none"> <li>1. After you select <b>Workflow user</b>, click the <b>Workflow user</b> tab.</li> <li>2. In the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ol>
<b>User</b>	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. Select the users to send notifications to, and then move these users to the <b>Selected users</b> list.</li> </ol>

10. Repeat steps 3 through 9 for each event that you selected in step 2.

## Specify a final approver

You can designate a final approver for scenarios where the approver is the person who submitted the document for approval and the "disallow approval by submitter" is being used. Follow these steps to specify a final approver.

1. In the workflow editor, right-click the approval element, and then select **Properties** to open the **Properties** form.
2. In the left pane, click **Advanced settings**.
3. Select the **Use final approver** check box.
4. In the list, select a user to be the final approver.

## Set a time limit

Follow these steps if the approval process must be completed in a specific time.

### NOTE

The options that you select in these steps override the options that you selected in the **Assignment** and **Escalation** areas of each approval step.

1. In the left pane, click **Advanced settings**.
2. Select the **Set a time limit for the workflow element** check box.
3. In the **Duration** field, specify when the approval process must be completed. Select one of the following options:
  - **Hours** – Enter the number of hours in which the approval process must be completed. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Days** – Enter the number of days in which the approval process must be completed. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Weeks** – Enter the number of weeks in which the approval process must be completed.
  - **Months** – Select the day and week by which the approval process must be completed. For example, you may want the approval process to be completed by Friday of the third week of the month.
  - **Years** – Select the day, week, and month by which the approval process must be completed. For example, you may want the approval process to be completed by Friday of the third week of December.
4. If the time limit is exceeded, the system acts on the document. In the **Action** list, select the action that the system should take.

## Specify which actions are available to the user

When a document is assigned to a user for approval, the user must act on the document. Follows these steps to specify which actions the user can take on the document that was submitted.

1. In the left pane, click **Advanced settings**.
2. Select the **Approve** check box if the user can approve the document.
3. Select the **Reject** check box the user can reject the document.
4. Select the **Request change** check box the user can request changes to the document.
5. Select the **Delegate** check box if the user can assign the document to another user for approval.

**NOTE**

The **Enable actions from the work list in Enterprise Portal** check box has been deprecated.

## Configure the approval steps

An approval process consists of approval steps. Complete the following procedure to add steps the approval process and configure the steps.

1. In the workflow editor, double-click the approval process. The workflow editor displays the steps of the approval process.
2. To add an approval step, drag the step from the **Workflow elements** area to the canvas.
3. To configure an approval step, see [Configure approval steps in a workflow](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure approval steps in a workflow

2/18/2021 • 11 minutes to read • [Edit Online](#)

This topic explains how to configure the properties of an approval step.

To configure an approval step in the workflow editor, right-click the approval step, and then click **Properties** to open the **Properties** page. Then use the following procedures to configure the properties of the approval step.

## Name the step

Follow these steps to enter a name for the approval step.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the approval step.

## Enter a subject line and instructions

You must provide a subject line and instructions to users who are assigned to the approval step. For example, if you're configuring an approval step for purchase requisitions, the user who is assigned to the step sees the subject line and instructions on the **Purchase requisitions** page. The subject line appears in a message bar on the page. The user can then click the icon in the message bar to see the instructions. Follow these steps to enter a subject line and instructions.

1. In the left pane, click **Basic Settings**.
2. In the **Work item subject** field, enter the subject line.
3. To personalize the subject line, you can insert placeholders. Placeholders are replaced with appropriate data when the subject line is shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.
4. To add translations of the subject line, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 3.
  - f. Click **Close**.
5. In the **Work item instructions** field, enter the instructions.
6. To personalize the instructions, you can insert placeholders. Placeholders are replaced with appropriate data when the instructions are shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.

7. To add translations of the instructions, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 6.
  - f. Click **Close**.

## Assign the approval step

Follow these steps to specify who the approval step should be assigned to.

1. In the left pane, click **Assignment**.
2. On the **Assignment type** tab, select one of the options in the following table, and then follow the additional steps for that option before you go to step 3.

OPTION	USERS THAT THE APPROVAL STEP IS ASSIGNED TO	ADDITIONAL STEPS
Participant	Users who are assigned to a specific group or role	<ol style="list-style-type: none"> <li>1. After you select <b>Participant</b>, on the <b>Role based</b> tab, in the <b>Type of participant</b> list, select the type of group or role to assign the step to.</li> <li>2. In the <b>Participant</b> list, select the group or role to assign the step to.</li> </ol>

OPTION	USERS THAT THE APPROVAL STEP IS ASSIGNED TO	ADDITIONAL STEPS
Hierarchy	Users in a specific organizational hierarchy	<ol style="list-style-type: none"> <li>1. After you select <b>Hierarchy</b>, on the <b>Hierarchy selection</b> tab, in the <b>Hierarchy type</b> list, select the type of hierarchy to assign the step to.</li> <li>2. The system must retrieve a range of user names from the hierarchy. These names represent users that the step can be assigned to. Follow these steps to specify the starting point and ending point of the range of user names that the system retrieves: <ol style="list-style-type: none"> <li>a. To specify the starting point, select a person in the <b>Start from</b> list.</li> <li>b. To specify the ending point, click <b>Add condition</b>. Then enter a condition that determines where in the hierarchy the system stops retrieving names.</li> </ol> </li> <li>3. On the <b>Hierarchy options</b> tab, specify which users in the range the step should be assigned to: <ul style="list-style-type: none"> <li>• <b>Assign to all users retrieved</b> – The step is assigned to all users in the range.</li> <li>• <b>Assign only to last user retrieved</b> – The step is assigned to only the last user in the range.</li> <li>• <b>Exclude users with the following condition</b> – The step isn't assigned to any users in the range who meet a specific condition. Click <b>Add condition</b> to specify the condition.</li> </ul> </li> </ol>
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>

OPTION	USERS THAT THE APPROVAL STEP IS ASSIGNED TO	ADDITIONAL STEPS
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all system users. Select the users to assign the step to, and then move those users to the <b>Selected users</b> list.</li> </ol>

3. On the **Time limit** tab, in the **Duration** field, specify how much time the user has to take action on, or respond to, documents that reach the approval step. Select one of the following options:

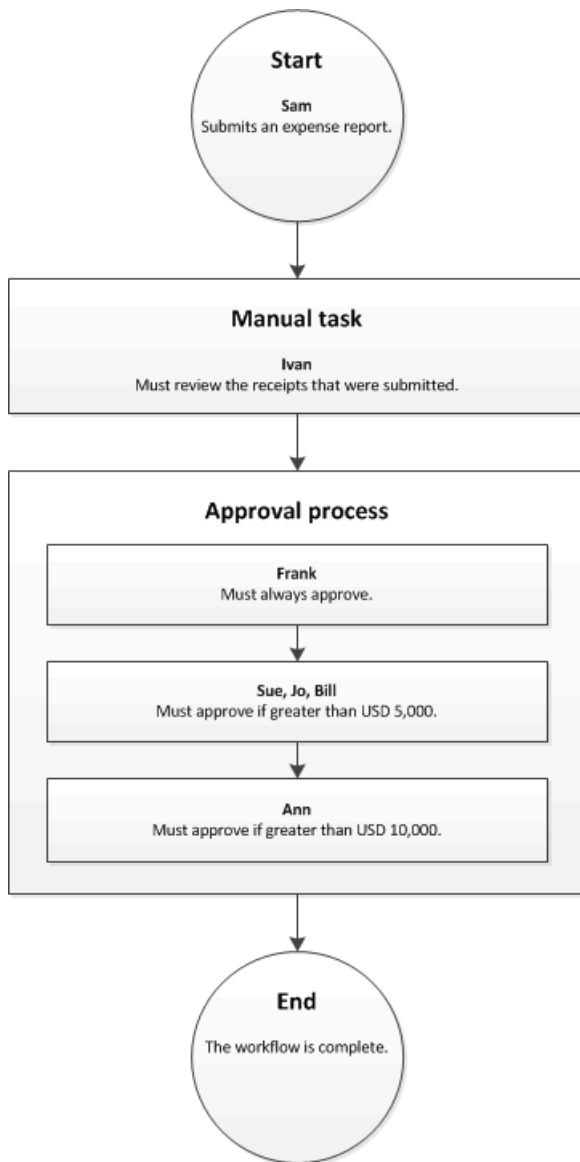
- **Hours** – Enter the number of hours that the user has to respond. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Days** – Enter the number of days that the user has to respond. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Weeks** – Enter the number of weeks that the user has to respond.
- **Months** – Select the day and week that the user must respond by. For example, you might want the user to respond by Friday of the third week of the month.
- **Years** – Select the day, week, and month that the user must respond by. For example, you might want the user to respond by Friday of the third week of December.

If the user doesn't take action on the document in the allotted time, the document is overdue. A document that is overdue is escalated, based on the options that you select in the **Escalation** area of the page.

4. If you assigned the approval step to multiple users or a group of users, on the **Completion policy** tab, select one of the following options:

- **Single approver** – The action that is applied to the document is determined by the first person who responds. For example, Sam has submitted an expense report for USD 15,000. The expense report is currently assigned to Sue, Jo, and Bill. If Sue is the first person who responds to the document, the action that she takes is applied to the document. If Sue rejects the document, it's rejected and sent back to Sam. If Sue approves the document, it's sent to Ann for approval.





- **Majority of approvers** – The action that is applied to the document is determined when most of the approvers respond. For example, Sam has submitted an expense report for USD 15,000. The expense report is currently assigned to Sue, Jo, and Bill. If Sue and Jo are the first two approvers who respond, the action that they take is applied to the document.
  - If Sue approves the document, but Jo rejects it, the document is rejected and sent back to Sam.
  - If both Sue and Jo approve the document, it's sent to Ann for approval.
- **Percentage of approvers** – The action that is applied to the document is determined when a specific percentage of the approvers respond. For example, Sam has submitted an expense report for USD 15,000. The expense report is currently assigned to Sue, Jo, and Bill, and you entered 50 as the percentage. If Sue and Jo are the first two approvers who respond, the action that they take is applied to the document, because they meet the requirement for 50 percent of approvers.
  - If Sue approves the document, but Jo rejects it, the document is rejected and sent back to Sam.
  - If both Sue and Jo approve the document, it's sent to Ann for approval.
- **All approvers** – All the approvers must approve the document. Otherwise, the workflow can't continue. For example, Sam has submitted an expense report for USD 15,000. The expense report is currently assigned to Sue, Jo, and Bill. If Sue and Joe approve the document, but Bill rejects it, the document is rejected and sent back to Sam. If Sue, Jo, and Bill all approve the document, it's sent to Ann for approval.

Specify when the approval step is required

You can specify when the approval step is required. The approval step can always be required, or it can be required only if specific conditions are met.

### The approval step is always required

Follow these steps if the approval step is always required.

1. In the left pane, click **Condition**.
2. Select the **Always run this step** option.

### The approval step is required in specific conditions

The approval step that you're configuring might be required only if specific conditions are met. For example, if you're configuring an approval step for a purchase requisition workflow, you might want the approval step to occur only if the amount of the purchase requisition is more than USD 10,000. Follow these steps to specify when the approval step is required.

1. In the left pane, click **Condition**.
2. Select the **Run this step only when the following condition is met** option.
3. Enter a condition.
4. Enter any additional conditions that are required.
5. To verify that the conditions that you entered are configured correctly, follow these steps:
  - a. Click **Test**.
  - b. On the **Test workflow condition** page, in the **Validate condition** area, select a record.
  - c. Click **Test**. The system evaluates the record to determine whether it meets the conditions that you defined.
  - d. Click **OK** or **Cancel** to return to the **Properties** page.

## Specify what happens when the document is overdue

If a user doesn't take action on a document in the allotted time, the document is overdue. A document that is overdue can be escalated, or automatically assigned to another user for approval. Follow these steps to escalate the document if it's overdue.

1. In the left pane, click **Escalation**.
2. Select the **Use escalation path** check box to create an escalation path. The system automatically assigns the document to the users who are listed in the escalation path. For example, the following table represents an escalation path.

SEQUENCE	ESCALATION PATH
1	Assign to: Donna
2	Assign to: Erin
3	Final action: Reject

In this example, the system assigns the overdue document to Donna. If Donna doesn't respond in the allotted time, the system assigns the document to Erin. If Erin doesn't respond in the allotted time, the system rejects the document.

3. To add a user to the escalation path, click **Add escalation**. On the **Assignment type** tab, select one of the options in the following table, and then follow the additional steps for that option before you go to

step 4.

OPTION	USERS THAT THE DOCUMENT IS ESCALATED TO	ADDITIONAL STEPS
Hierarchy	Users in a specific organizational hierarchy	<ol style="list-style-type: none"><li>1. After you select <b>Hierarchy</b>, on the <b>Hierarchy selection</b> tab, in the <b>Hierarchy type</b> list, select the type of hierarchy to escalate the document to.</li><li>2. The system must retrieve a range of user names from the hierarchy. These names represent users that the document can be escalated to. Follow these steps to specify the starting point and ending point of the range of user names that the system retrieves:<ol style="list-style-type: none"><li>a. To specify the starting point, select a person in the <b>Start from</b> list.</li><li>b. To specify the ending point, click <b>Add condition</b>. Then enter a condition that determines where in the hierarchy the system stops retrieving names.</li></ol></li><li>3. On the <b>Hierarchy options</b> tab, specify which users in the range the document should be escalated to:<ul style="list-style-type: none"><li>• <b>Assign to all users retrieved</b> – The document is escalated to all users in the range.</li><li>• <b>Assign only to last user retrieved</b> – The document is escalated to only the last user in the range.</li><li>• <b>Exclude users with the following condition</b> – The document isn't escalated to any users in the range who meet a specific condition. Click <b>Add condition</b> to specify the condition.</li></ul></li></ol>

OPTION	USERS THAT THE DOCUMENT IS ESCALATED TO	ADDITIONAL STEPS
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all users. Select the users to escalate the document to, and then move those users to the <b>Selected users</b> list.</li> </ol>

4. On the **Time limit** tab, in the **Duration** field, specify how much time the user has to take action on, or respond to, documents. Select one of the following options:
  - **Hours** – Enter the number of hours that the user has to respond. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Days** – Enter the number of days that the user has to respond. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Weeks** – Enter the number of weeks that the user has to respond.
  - **Months** – Select the day and week that the user must respond by. For example, you might want the user to respond by Friday of the third week of the month.
  - **Years** – Select the day, week, and month that the user must respond by. For example, you might want the user to respond by Friday of the third week of December.
5. Repeat steps 3 through 4 for each user that should be added to the escalation path. You can change the order of the users.
6. If the users in the escalation path don't respond in the allotted time, the system automatically take action on the document. To specify the action that the system takes, select the **Action** row, and then, on the **End action** tab, select an action.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure manual decisions in a workflow

2/18/2021 • 12 minutes to read • [Edit Online](#)

This topic explains how to configure the properties of a manual decision.

To configure a manual decision in the workflow editor, right-click the manual decision, and then click **Properties** to open the **Properties** page. Then use the following procedures to configure the properties of the manual decision.

## Name the decision

Follow these steps to enter a name for the manual decision.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the manual decision.

## Enter a subject line and instructions

You must provide a subject line and instructions to users who are assigned to the manual decision. For example, if you're configuring a decision for purchase requisitions, the user who is assigned to the decision sees the subject line and instructions on the **Purchase requisitions** page. The subject line appears in a message bar on the page. The user can then click the icon in the message bar to view the instructions. Follow these steps to enter a subject line and instructions.

1. In the left pane, click **Basic Settings**.
2. On the **Instructions** tab, in the **Work item subject** field, enter the subject line.
3. To personalize the subject line, you can insert placeholders. Placeholders are replaced with appropriate data when the subject line is shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.
4. To add translations of the subject line, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 3.
  - f. Click **Close**.
5. In the **Work item instructions** field, enter the instructions.
6. To personalize the instructions, you can insert placeholders. Placeholders are replaced with appropriate data when the instructions are shown to users. Follow these steps to insert a placeholder:
  - a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.

- d. Click **Insert**.
7. To add translations of the instructions, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 6.
  - f. Click **Close**.

## Specify the possible outcomes of a decision

Typically, when a document is assigned to a decision maker, the decision maker is asked a question. The answer to this question is usually **Yes** or **No**, or **True** or **False**. Follow these steps to specify the possible outcomes of the manual decision.

1. In the left pane, click **Basic Settings**.
2. On the **Outcomes** tab, in the **Outcome 1** field, enter the name of the outcome, or the option.
3. To add translations of the outcome, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. Click **Close**.
4. In the **Outcome 2** field, enter the name of the outcome, or the option.
5. To add translations of the outcome, follow these steps:
  - a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. Click **Close**.

## Specify when notifications are sent

You can send notifications to people when a decision has been made, delegated, or escalated. Follow these steps to specify when notifications are sent, and who the notifications are sent to.

1. In the left pane, click **Notifications**.
2. Select the check box next to the events that notifications should be sent for:
  - **[Choice 1]** – The assigned user has selected **[Choice 1]**.
  - **[Choice 2]** – The assigned user has selected **[Choice 2]**.
  - **Delegate** – The assigned user has assigned the decision to another user.
  - **Escalate** – The assigned user hasn't made the decision in the allotted time.
3. Select the row for an event that you selected in step 2.
4. On the **Notification text** tab, in the text box, enter the text of the notification.
5. To personalize the notification, you can insert placeholders. Placeholders are replaced with appropriate

data when the notification is show to users. Follow these steps to insert a placeholder:

- a. In the text box, click where the placeholder should appear.
  - b. Click **Insert placeholder**.
  - c. In the list that appears, select the placeholder to insert.
  - d. Click **Insert**.
6. To add translations of the notification, follow these steps:
- a. Click **Translations**.
  - b. On the page that appears, click **Add**.
  - c. In the list that appears, select the language that you're entering the text in.
  - d. In the **Translated text** field, enter the text.
  - e. To personalize the text, you can insert placeholders as described in step 5.
  - f. Click **Close**.
7. On the **Recipient** tab, specify who the notifications are sent to. Select one of the options in the following table, and then follow the additional steps for that option before you go to step 8.

OPTION	NOTIFICATION RECIPIENTS	ADDITIONAL STEPS
Participant	Users who are assigned to a specific group or role	<ol style="list-style-type: none"> <li>1. After you select <b>Participant</b>, on the <b>Role based</b> tab, in the <b>Type of participant</b> list, select the type of group or role to send notifications to.</li> <li>2. In the <b>Participant</b> list, select the group or to send notifications to.</li> </ol>
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all users. Select the users to send notifications to, and then move those users to the <b>Selected users</b> list.</li> </ol>

8. Repeat steps 3 through 7 for each event that you selected in step 2.

## Assign a decision

Follow these steps to specify who a manual decision should be assigned to.

1. In the left pane, click **Assignment**.
2. On the **Assignment type** tab, select one of the options in the following table, and then follow the additional steps for that option before you go to step 3.

OPTION	USERS THAT THE DECISION IS ASSIGNED TO	ADDITIONAL STEPS
Participant	Users who are assigned to a specific group or role	<ol style="list-style-type: none"><li data-bbox="1023 203 1342 387">1. After you select <b>Participant</b>, on the <b>Role based</b> tab, in the <b>Type of participant</b> list, select the type of group or role to assign the decision to.</li><li data-bbox="1023 398 1342 488">2. In the <b>Participant</b> list, select the group or role to assign the decision to.</li></ol>



OPTION	USERS THAT THE DECISION IS ASSIGNED TO	ADDITIONAL STEPS
Hierarchy	Users in a specific organizational hierarchy	<ol style="list-style-type: none"> <li>1. After you select <b>Hierarchy</b>, on the <b>Hierarchy selection</b> tab, in the <b>Hierarchy type</b> list, select the type of hierarchy to assign the decision to.</li> <li>2. The system must retrieve a range of user names from the hierarchy. These names represent users that the decision can be assigned to. Follow these steps to specify the starting point and ending point of the range of user names that the system retrieves: <ol style="list-style-type: none"> <li>a. To specify the starting point, select a person in the <b>Start from</b> list.</li> <li>b. To specify the ending point, click <b>Add condition</b>. Then enter a condition that determines where in the hierarchy the system stops retrieving names.</li> </ol> </li> <li>3. On the <b>Hierarchy options</b> tab, specify which users in the range the decision should be assigned to: <ul style="list-style-type: none"> <li>• <b>Assign to all users retrieved</b> – The decision is assigned to all users in the range.</li> <li>• <b>Assign only to last user retrieved</b> – The decision is assigned to only the last user in the range.</li> <li>• <b>Exclude users with the following condition</b> – The decision isn't assigned to any users in the range who meet a specific condition. Click <b>Add condition</b> to specify the condition.</li> </ul> </li> </ol>
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>

OPTION	USERS THAT THE DECISION IS ASSIGNED TO	ADDITIONAL STEPS
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all users. Select the users to assign the decision to, and then move those users to the <b>Selected users</b> list.</li> </ol>

3. On the **Time limit** tab, in the **Duration** field, specify how much time the user has to make the decision. Select one of the following options:

- **Hours** – Enter the number of hours that the user has to make the decision. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Days** – Enter the number of days that the user has to make the decision. Then select the calendar that your organization uses, and enter information about your organization's work week.
- **Weeks** – Enter the number of weeks that the user has to make the decision.
- **Months** – Select the day and week that the user must make the decision by. For example, you might want the user to make the decision by Friday of the third week of the month.
- **Years** – Select the day, week, and month that the user must make the decision by. For example, you might want the user to make the decision by Friday of the third week of December.

If the user doesn't make the decision in the allotted time, the decision is overdue. A decision that is overdue is escalated, based on the options that you select in the **Escalation** area of the page.

## Specify what happens when a decision is overdue

If a user doesn't make the decision in the allotted time, the decision is overdue. A decision that is overdue can be escalated, or automatically assigned to another user. Follow these steps to escalate the decision if it's overdue.

1. In the left pane, click **Escalation**.
2. Select the **Use escalation path** check box to create an escalation path. The system automatically assigns the decision to the users who are listed in the escalation path. For example, the following table represents an escalation path.

SEQUENCE	ESCALATION PATH
1	Assign to: Donna
2	Assign to: Erin
3	Final action: [Choice 1]

In this example, the system assigns the overdue decision to Donna. If Donna doesn't make the decision in the allotted time, the system assigns the decision to Erin. If Erin doesn't make the decision in the allotted time, the system selects **[Choice 1]** as the decision.

3. To add a user to the escalation path, click **Add escalation**. Select one of the options in the following table, and then follow the additional steps for that option before you go to step 4.

OPTION	USERS THAT THE DECISION IS ESCALATED TO	ADDITIONAL STEPS
Hierarchy	Users in a specific organizational hierarchy	<ol style="list-style-type: none"> <li>1. After you select <b>Hierarchy</b>, on the <b>Hierarchy selection</b> tab, in the <b>Hierarchy type</b> list, select the type of hierarchy to escalate the decision to.</li> <li>2. The system must retrieve a range of user names from the hierarchy. These names represent users that the decision can be escalated to. Follow these steps to specify the starting point and ending point of the range of user names that the system retrieves: <ol style="list-style-type: none"> <li>a. To specify the starting point, select a person in the <b>Start from</b> list.</li> <li>b. To specify the ending point, click <b>Add condition</b>. Then enter a condition that determines where in the hierarchy the system stops retrieving names.</li> </ol> </li> <li>3. On the <b>Hierarchy options</b> tab, specify which users in the range the decision should be escalated to: <ul style="list-style-type: none"> <li>• <b>Assign to all users retrieved</b> – The decision is escalated to all users in the range.</li> <li>• <b>Assign only to last user retrieved</b> – The decision is escalated to only the last user in the range.</li> <li>• <b>Exclude users with the following condition:</b> – The decision isn't escalated to any users in the range who meet a specific condition. Click <b>Add condition</b> to specify the condition.</li> </ul> </li> </ol>
Workflow user	Users in the current workflow	<ul style="list-style-type: none"> <li>• After you select <b>Workflow user</b>, on the <b>Workflow user</b> tab, in the <b>Workflow user</b> list, select a user who participates in the workflow.</li> </ul>

OPTION	USERS THAT THE DECISION IS ESCALATED TO	ADDITIONAL STEPS
User	Specific users	<ol style="list-style-type: none"> <li>1. After you select <b>User</b>, click the <b>User</b> tab.</li> <li>2. The <b>Available users</b> list includes all users. Select the users to escalate the decision to, and then move those users to the <b>Selected users</b> list.</li> </ol>

4. On the **Time limit** tab, in the **Duration** field, specify how much time the user has to make the decision. Select one of the following options:
  - **Hours** – Enter the number of hours that the user has to make the decision. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Days** – Enter the number of days that the user has to make the decision. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Weeks** – Enter the number of weeks that the user has to make the decision.
  - **Months** – Select the day and week that the user must make the decision by. For example, you might want the user to make the decision by Friday of the third week of the month.
  - **Years** – Select the day, week, and month that the user must make the decision by. For example, you might want the user to make the decision by Friday of the third week of December.
5. Repeat steps 3 through 4 for each user that should be added to the escalation path. You can change the order of the users.
6. If the users in the escalation path don't make the decision in the allotted time, the system makes the decision. To specify the option that the system selects, select the **Action** row, and then, on the **End action** tab, select an option.

## Set a time limit

Follow these steps if the decision must be made in a specific time.

### NOTE

The options that you select in this procedure override the options that you selected in the **Assignment** and **Escalation** areas of the page.

1. In the left pane, click **Advanced settings**.
2. Select the **Set a time limit for the workflow element** check box.
3. In the **Duration** field, specify when the decision must be made. Select one of the following options:
  - **Hours** – Enter the number of hours. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Days** – Enter the number of days. Then select the calendar that your organization uses, and enter information about your organization's work week.
  - **Weeks** – Enter the number of weeks.
  - **Months** – Select the day and week that the decision must be made by. For example, you might want the decision to be made by Friday of the third week of the month.
  - **Years** – Select the day, week, and month that the decision must be made by. For example, you might

want the decision to be made by Friday of the third week of December.

4. If the time limit is exceeded, the system makes the decision. In the **Action** list, select the option that the system should select.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure conditional decisions in a workflow

2/18/2021 • 2 minutes to read • [Edit Online](#)

Use the following procedure to configure the properties of a conditional decision.

A conditional decision is a point at which a workflow divides into two branches. To configure a conditional decision, in the workflow editor, right-click the conditional decision, and then click **Properties** to open the **Properties** form.

## Name a decision

Follow these steps to enter a name for a conditional decision.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the conditional decision.

## Set conditions

The system determines which branch is used by evaluating the submitted document to determine whether it meets specific conditions.

1. In the left pane, click **Basic Settings**.
2. Click **Add condition**.
3. Enter a condition.
4. Enter additional conditions, if they are required.
5. To verify that the conditions that you entered are configured correctly, complete the following steps:
  - a. Click **Test** to open the **Test workflow condition** form.
  - b. Select a record in the **Validate condition** area of the form.
  - c. Click **Test**. The system evaluates the record to determine whether it meets the conditions that you defined.
  - d. Click **OK** or **Cancel** to return to the **Properties** form.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure parallel activities in a workflow

2/18/2021 • 2 minutes to read • [Edit Online](#)

To configure a parallel activity, complete the following procedures in the workflow editor.

A parallel activity consists of workflow branches that run at the same time.

## Name a parallel activity

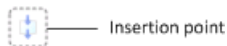
Follow these steps to enter a name for a parallel activity.

1. Right-click the parallel activity, and then click **Properties** to open the **Properties** form.
2. In the left pane, click **Basic Settings**.
3. In the **Name** field, enter a unique name for the parallel activity.
4. Click **Close**.

## Configure the branches of a parallel activity

Follow these steps to add and configure the branches of this parallel activity.

1. Double-click the parallel activity to display the branches of the parallel activity.
2. To add a branch, drag the **Branch** element from the **Workflow elements** area to an insertion point on the canvas. The following figure shows an insertion point.



### NOTE

The order of the branches is not important because all the branches of a parallel activity run at the same time.

3. To configure each branch, see [Configure parallel branches in a workflow](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure parallel branches in a workflow

2/18/2021 • 2 minutes to read • [Edit Online](#)

To configure a parallel branch, complete the following procedures in the workflow editor.

A parallel branch is essentially a workflow that runs in the context of a parent workflow.

## Name a branch

Follow these steps to enter a name for a parallel branch.

1. Right-click the parallel branch, and then click **Properties**. The **Properties** form is displayed.
2. In the left pane, click **Basic Settings**.
3. In the **Name** field, enter a unique name for the parallel branch.
4. Click **Close**.

## Design and configure the elements of a branch

Follow these steps to design and configure the elements of a parallel branch.

1. Double-click the parallel branch.
2. Drag workflow elements onto the canvas, and then configure the elements, just as you would to create any other workflow. For more information, see [Create workflows overview](#).

## Additional resources

[Create workflows overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure line-item workflows

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains how to configure a line-item workflow element.

To configure a line-item workflow element, in the workflow editor, right-click the element, and then click **Properties** to open the **Properties** page. Then use the following procedures to configure the properties of the line-item workflow element.

## Name the line-item workflow element

Follow these steps to enter a name for the line-item workflow element.

1. In the left pane, click **Basic Settings**.
2. In the **Name** field, enter a unique name for the line-item workflow element.

## Specify whether the same workflow is used to process all line items

Follow these steps to specify whether the same workflow is used to process all the line items on a document.

1. In the left pane, click **Basic Settings**.
2. If the same workflow should process all the line items on a document, click **Invoke a single workflow for all line-items**. Then select the workflow to use to process the line items.
3. If a specific workflow should process line items that meet a specific set of conditions, click **Invoke a workflow for each line-item**. Then follow these steps to define the set of conditions:
  - a. Click **Add**.
  - b. Select the condition in the table.
  - c. On the **Condition name** tab, enter a name for the set of conditions that you're defining.
  - d. Click **Add condition** to enter a condition.
  - e. Enter any additional conditions that are required.
  - f. To verify that the set of conditions that you entered is configured correctly, click **Test**. On the **Test workflow condition** page, in the **Validate condition** area, select a record, and then click **Test**. The system evaluates the record to determine whether it meets the conditions that you defined. Click **OK** or **Cancel** to return to the **Properties** page.

On the **Workflow** tab, select the workflow select the workflow to use to process line items that meet the set of conditions that you defined.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure the Workflow message processing batch job as critical

2/18/2021 • 2 minutes to read • [Edit Online](#)

The workflow system uses various batch jobs. **Workflow message processing** is an important batch job used to process workflow messages. If workflow is a key component of your organization, you should consider configuring the **Workflow message processing** batch job as critical.

Configuring the **Workflow message processing** batch job as critical ensures that the system actively tracks its status. When a critical batch job fails, the support team can better monitor failures and take action to resolve any issues that may have caused the failure.

Follow these steps to configure the **Workflow message processing** batch job as critical.

1. Navigate to the **Batch jobs** page.
2. Search for **Workflow message processing** using the quick filter.
3. Select the **Workflow message processing** batch job.
4. Click **Edit** in the action pane.
5. Select the **Critical Job** check box.
6. Click **Save** in the action pane.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Delegate work items in a workflow

2/18/2021 • 2 minutes to read • [Edit Online](#)

## Manually delegate a work item

To delegate an individual work item, select the **Delegate** option in the **Workflow** menu and then enter the user to be delegated to along with a comment. This will reassign the work item to that user so they can complete it.

## Manually delegate multiple work items

Multiple work items can be delegated together from the **Work items assigned to me** page. The following workflow types are eligible for mass delegation: Purchase agreement approval workflow, Purchase order workflow, Purchase requisition review, and Vendor invoice workflow. The **Delegate multiple work items** feature is disabled by default and can be enabled in **Workspaces > Feature management**. Contact your system administrator for help with enabling this feature.

1. Go to **Common > Common > Work items > Work items assigned to me**.
2. Select the work items that will be delegated.
3. Click the **Delegate work items** menu.
4. In the **User** field, select the user to delegate the work items to.
5. In the **Comment** field, enter a comment that explains why you're delegating the work items.
6. Click the **Delegate work items** button to complete the work item delegation.

## Automatically delegate work items

If you plan to be out of the office or otherwise unavailable to act on work items for a period of time, you can automatically delegate new work items to other users using the **User options** page.

### Set up automatic delegation

1. Go to **Common > Setup > User options**.
2. Click the **Workflow** tab. Make sure the Delegation section is expanded. To configure the system to automatically delegate your work items to other users, you must create delegation rules, which specify when certain types of work items are delegated. Follow these steps to create a delegation rule.
3. Click **Add**.
4. In the **Scope** field, select an option:
  - All – Delegate all work items that are assigned to you.
  - Module – Delegate only the work items that are related to a specific type of workflow. If you select this option, you must select the type of workflow in the **Name** field.
  - Workflow – Delegate only the work items that are related to a specific workflow. If you select this option, you must select the workflow in the **Name** field.
5. In the **Name** field:
  - For **Module** scope, select the target module.
  - For **Workflow** scope, select the target workflow.
6. In the **Delegate** field, select the user to delegate the work items to. Use the **Start date/time** and **End date/time** fields to specify when you want the work items to be automatically delegated.
7. In the **Start date/time** field, enter a date and time.
8. In the **End date/time** field, enter a date and time.

9. Select the **Enabled** check box to activate the delegation rule.
10. In the **Comment** field, enter a comment that explains why you're delegating the work items.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# View workflow history

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the steps to view the status of a document that was submitted to the workflow system for processing and approval. The demo data company used to create this procedure is USMF.

1. Go to **Navigation pane > Modules > Common > Inquiries > Workflow > Workflow history**.

- Use this form to view the status of a document that was submitted to the workflow system for processing and approval.
- The **Instance ID** is the identification code of the workflow instance that is processing, or has processed the document.
- The **Status** is the workflow status of the document.
- The **Workflow ID** is the identification code of the workflow that is processing, or has processed the document.
- The **Document** is the identification code of the document.
- The **Document type** is the type of document that was submitted for processing.
- The **Workflow** is the name of the workflow that is processing, or has processed the document.
- The **Version** is the version number of the workflow that is processing, or has processed the document.
- The **Created date and time** is the date and time that the document was submitted.
- The **Elapsed time** is the time that has passed since the document was submitted.
- The **Resume** button allows you to resume the workflow process for the selected document.
- The **Recall** button allows you to recall the selected document so that it is not processed.

2. In the list, select the link in the desired row.

- Make sure the **Work items** section is expanded. In this section you can view the work items that are associated with the selected document. For example, a task may have to be completed, or the document may have to be approved.
- The **Reassign** button will open a dialog box where you can reassign a work item to another user.
- Make sure the **Tracking details** section is expanded. In this section you can view the workflow history of the selected document.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable users to receive workflow-related email messages

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can configure the system to send email messages to users when workflow-related events occur. For example, email messages can be sent to users when documents are assigned to them for approval. The demo data company used to create this procedure is USMF.

1. Go to **Navigation pane > Modules > System administration > Users > Users**.
2. In the list, find and select the desired record.
3. On the **Action pane**, click **User options**.
4. Click the **Workflow** tab. Make sure that the **Notifications** section is expanded. In the **Notifications** section, you can specify how you want the user to be notified about workflow-related events.
5. In the **Line-item workflow notification type** field, select an option.
  - Grouped – Notifications for line items are grouped into a single email message.
  - Individual – An email message is sent for each line item.
  - If you want the user to receive notifications in the client, select the **Send notifications in email** check box.
6. Click **Save**.
7. Close the page.

## NOTE

The workflow email templates will be sourced from either system email templates or organization email templates depending on whether the workflow is a system-level (not company specific) or organization-level (company specific) workflow.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a workflow type

2/18/2021 • 2 minutes to read • [Edit Online](#)

To add workflow support for a document, you must create a workflow type. After you create a workflow type, it can be used to create workflow configurations for the document. This topic provides links to the procedures for creating a workflow type.

You create workflow types to define the following elements:

- The workflow document to assign the workflow to
- A category that defines the module that the workflow type is available in
- Tasks, automated tasks, and approvals that are supported for the workflow
- The workflow started, completed, configuration data change, and canceled event handlers
- A `SubmitToWorkflowMenuItem` menu item

## In this section

- [Workflow type checklist](#)
- [Create a workflow category](#)
- [Create a query for a workflow type](#)
- [Create a new workflow type](#)
- [Create a workflow document class](#)
- [Create a `SubmitToWorkflow` class](#)
- [Associate a workflow document class with a workflow type](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Workflow type checklist

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes the steps that are required to create a new workflow type. Workflow types are used to create configurations for a workflow.

## Workflow type checklist

1. If an existing category isn't appropriate, create a new workflow category to use for the workflow type. For more information, see [Create a workflow category](#).
2. Create a query that accesses the document that the new workflow type is being created for. For more information, see [Create a query for a workflow type](#).
3. Create a workflow type in Application Explorer. Typically, you complete this step by using the **Workflow** wizard. For more information, see [Create a new workflow type](#).

We recommend that you use the **Workflow** wizard to create a workflow type. The wizard performs the following tasks. In some cases, you must add more code for your workflow type. Links are provided so that you can see the details of the actions that the wizard performs and the additional steps that you must perform to complete your workflow type.

1. Create a workflow document class, and then bind the query that is used for the workflow to the class. For more information, see [Create a workflow document class](#).
2. Bind the workflow document class to the workflow type. For more information, see [Associate a workflow document class with a workflow type](#).
3. Create workflow event handlers for started, completed, configuration change, and canceled events, and then bind event handlers to the workflow type. For more information, see [Create a workflow event handler](#).
4. Create a class for **SubmitToWorkflowMenuItem**. Optionally create a class for **CancelMenuItem**. Bind the classes to the action menu items that you will create in the next step. For more information, see [Classes and methods](#).
5. Create an action menu item for the **SubmitToWorkflowMenuItem** workflow property. Optionally create an action menu item for the **CancelMenuItem** property. Bind the actions to the workflow type. For more information, see [Create a new workflow type](#).

## Next steps

After you create a workflow type, you can add tasks, automated tasks, and approvals.

## See also

[Configure workflow properties](#)

[Configure manual tasks in a workflow](#)

[Configure automated tasks in a workflow](#)

[Configure approval processes in a workflow](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a workflow category

2/18/2021 • 2 minutes to read • [Edit Online](#)

When you create a workflow type, you must assign it to a workflow category. The workflow category determines whether the workflow type is available in a specific module. If an appropriate workflow category doesn't already exist, you must create it.

For example, a workflow type for a customer invoice should not be available in the **Master planning** module. To make the workflow type available only in the **Customer** module, select the **Customer** module when you create a workflow category.

This topic describes how to create a workflow category.

## Create a workflow category

1. In Application Explorer, expand the **Business Process and Workflow** node.
2. Follow one of these steps to create the workflow category:
  - Right-click the **Workflow Categories** node, and then select **New Workflow Category**. A new workflow category group appears under the **Workflow Categories** node. Right-click the new workflow category, and then select **Properties**.
  - On the **Project** menu, select **Add new item**. In the **Add new item** dialog box, select **Workflow category**. Enter a name for the category, and then select **Create**.
3. On the **Properties** sheet, set the following properties, as required.

PROPERTY	VALUE
Name	Specify the name that is used to reference the workflow category.
Label	Specify the label that is used for the workflow category in the user interface (UI).
Help Text	Specify the description that is shown for the workflow category in the workflow configuration UI.
Module	Specify the module that the workflow will be available in. The default module is <b>Ledger</b> .

After you create a workflow category, a workflow type can be bound to it. Typically, the **Workflow** wizard completes this step. However, the following procedure explains how to manually bind a workflow type to a workflow category.

## Bind a workflow type to a workflow category

1. In Application Explorer, expand the **Business Process and Workflow** node, and then expand the **Workflow Types** node.
2. Right-click the workflow type that you want to bind to a workflow category, and then select **Properties**.
3. On the **Properties** sheet, set the **Category** property to the workflow category that you created in the previous procedure.

## See also

[Create a workflow type](#)

[Create a new workflow type](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a query for a workflow type

2/18/2021 • 2 minutes to read • [Edit Online](#)

Before you create a workflow type, you must create a query that will access the table fields for the workflow document. This topic describes how to create a query for a workflow type.

## Create a query for a workflow type

1. Follow one of these steps to create a new query:
  - In Application Explorer, right-click the **Queries** node, and then select **New Query**. A query group appears under the **Queries** node. Right-click the new query, and then select **Rename**. Enter a name for the query.
  - On the **Project** menu, select **Add new item**. In the **Add new item** dialog box, select **Query**. Enter a name for the query, and then select **Create**.
2. Expand the new query, right-click the **Data Sources** node, and then select **New Data Source**. A data source group appears under the **Data Sources** node.
3. Right-click the new data source group, and then select **Properties**.
4. On the **Properties** sheet, set the **Table** property to the main table for the document type that you're defining a workflow for.
5. In the Application Object Tree (AOT), right-click the new query, and then select **Save**.

For more information about how to create queries, see [Create queries by using the AOT](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a new workflow type

2/18/2021 • 2 minutes to read • [Edit Online](#)

To make the workflow process available for a workflow document, you must create the workflow types that are used in the workflow configuration user interface (UI). This topic describes how to create a workflow type in Application Explorer.

A workflow type defines the following information:

- The document that the workflow is used for
- Workflow categories, which are used to assign a workflow type to a specific module
- Tasks, automated tasks, approvals, and line item workflows that the user can configure
- Menu items and event handlers

## Create a new workflow type

1. Follow one of these steps to open the **Workflow** wizard. The wizard will help you create a new workflow type.
  - In Application Explorer, right-click the **Business Process and Workflow** node, and then select **Add-Ins > Workflow type wizard**.
  - On the **Project** menu, select **Add new item**. In the **Add new item** dialog box, select **Workflow type**. Enter a name for the query, and then select **Create**.
2. Set the following values for the wizard.

VALUE	DESCRIPTION
Name	Specify the name that will be used for the workflow type.
Category	Select the workflow category that will be used for the workflow type. The category determines the module that the workflow type is available in. You can use an existing category or a new category that you create. For more information, see <a href="#">Create a workflow category</a> .
Query	Select the query that will access the table fields for the workflow document. For more information, see <a href="#">Create a query for a workflow type</a> .
Document menu item	Select the menu item that points to the main page that shows the document that you're creating the workflow type for.
Document web menu item	Select the web menu item that points to the Enterprise Portal page that shows the document that you're creating the workflow type for.

3. Specify the types of menu items that you want to create. You can create menu items, web menu items, or both.
4. Select **Next**. A list of all the resources that will be created for the workflow type is shown.
5. Select **Finish** to create the resources. The wizard creates classes, menu items, web menu items, the

workflow type, and a project that contains all the items.

6. When dialog box appears that indicates the status, select **OK**. The project that contains the workflow type resources is shown.

After you create a workflow type, the next step is to create a workflow document class to expose document data for conditions. For more information, see [Create a workflow document class](#).

## See also

[Create a workflow type](#)

[Create a workflow category](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a workflow document class

2/18/2021 • 3 minutes to read • [Edit Online](#)

You define table fields in a query to create workflow conditions. In a typical scenario, calculated fields are used to determine the behavior of a workflow. For example, a dynamic sales total of all records in a table can be used as a workflow condition to determine whether the step should be used. However, a limitation of queries is that you can't define calculated fields in the queries themselves. To overcome this query limitation, you must use a workflow document class. This topic describes how to create a workflow document class.

The workflow document class that you create defines table fields for conditions in two ways: the Application Explorer query and parameter methods. You must override the `getQueryName` method of the [WorkflowDocument class](#) to return the name of the query. You can optionally add calculated fields by adding parameter methods that have a specific signature on the class. For more information about workflow conditions, see [Configure workflow properties](#) and [Configure conditional decisions in a workflow](#).

The following procedures explain how to create a workflow document class that includes a parameter method for a calculated field. Before you begin, you must create a query that specifies the data that will be accessed. For more information about workflow queries, see [Create a query for a workflow type](#).

## NOTE

If you used the **Workflow** wizard to create the workflow type, the wizard has already created the workflow document class.

## Create a workflow document class

1. Follow one of these steps to create a new query:
  - In Application Explorer, expand the **Classes** node. Right-click the **Classes** node, and then select **New Class**. A class group appears under the **Classes** node. Right-click the new class, select **Rename**, and then enter a name for the workflow document class.
  - On the **Project** menu, select **Add new item**. In the **Add new item** dialog box, select **Class**. Enter a name for the class, and then select **Create**.
2. Expand the new class, select **classDeclaration**, right-click the class declaration, and then select **Edit**.
3. Enter the following code in the class declaration.

```
class <MyWorkflowDocumentClassName> extends WorkflowDocument
{
}
```

4. Close the **Editor** window, and select **Yes** to save your changes.
5. Right-click the new class, point to **Override Method**, and then select **getQueryName**. A method node that is named `getQueryName` appears under the workflow document class node, and the **Editor** window appears.

#### NOTE

Be sure to select `getQueryName` as the method to override. The `WorkflowDocument.getQuery` method is used only internally to retrieve the actual query, based on the string that is returned by the `WorkflowDocument.getQueryName` method.

6. In the **Editor** window, enter the following code for the query method.

```
QueryName getQueryName()
{
    return querystr(<MyWorkflowDocumentQueryName>);
}
```

After you create the workflow document class, you can bind it to the workflow type. For more information, see [Associate a workflow document class with a workflow type](#).

You can add a calculated field to the workflow document class only if it meets these conditions:

- It must be named `parm <name>`.
- It must define the `CompanyId`, `TableId`, and `RecId` parameters.
- It must return an extended data type (EDT) that will be included in the list of fields that define conditions or notification messages. The label for the EDT must uniquely identify the value.

## Add a calculated field to the workflow document class

1. In the workflow document class that you want to add a calculated field to, right-click the class, and then select **New Method**. A new method node appears under the **Classes** node.
2. Right-click the new method node, and then select **Edit**.
3. Enter code in the format that is shown in the following example. This example shows how to add a calculated field to determine the total credit amount for a journal.

```
public TotalJournalCreditAmount parmTotalJournalCreditAmount(CompanyId _companyId, TableId _tableId,
RecId _recId)
{
    // The calculateAmounts method contains business and validation logic
    this.calculateAmounts(_companyId, _tableId, _recId);
    return totalJournalCreditAmount;
}
```

## See also

[Associate a workflow document class with a workflow type](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Create a SubmitToWorkflow class

2/18/2021 • 2 minutes to read • [Edit Online](#)

A workflow is started when the user selects the **Submit** button on the workflow toolbar. The **Submit** button is bound to an action menu item that calls the **main** method of a class that you create to activate a workflow. This topic describes how to create a **SubmitToWorkflow** class and use the name of the workflow type to activate the workflow.

You can also activate a workflow by using the workflow configuration ID or the workflow sequence number. The basic procedure is the same. For more information, see [Activating a workflow](#).

## NOTE

If you used the **Workflow** wizard to create the workflow type, the wizard has already created a workflow submission manager class. You just have to add code to it.

## Create a SubmitToWorkflow class

1. In Application Explorer, expand the **Classes** node.
2. Right-click the **Classes** node, and then select **New Class**. A class group appears under the **Classes** node.
3. Right-click the new class, and then select **New Method**. A new method node appears under the **Classes** node.
4. Right-click the new method, and then select **Edit**.
5. Enter the following code for the **main** method to use the name of the workflow type to activate the workflow.

## NOTE

This example applies to workflow submissions. For an example that also works with Enterprise Portal, see [Adding enterprise portal support for workflow submission](#).

```

public static void main(Args args)
{
    // Variable declaration.
    recId _recId = args.record().RecId;
    WorkflowCorrelationId _workflowCorrelationId;
    // Hardcoded workflow type name.
    WorkflowTypeName _workflowTypeName = workflowtypestr("MyWorkflowType");
    // Initial note is the information that users enter when they
    // submit the document for workflow.
    WorkflowComment _initialNote = "";
    WorkflowSubmitDialog workflowSubmitDialog;
    // Opens the submit to workflow dialog box for user comments.
    workflowSubmitDialog =
WorkflowSubmitDialog::construct(args.caller().getActiveWorkflowConfiguration());
    workflowSubmitDialog.run();
    if (workflowSubmitDialog.parmIsClosedOK())
    {
        _recId = args.record().RecId;
        // Get user comments from the submit to workflow dialog box.
        _initialNote = workflowSubmitDialog.parmWorkflowComment();
        try
        {
            ttsbegin;
            // Activate the workflow from a template.
            _workflowCorrelationId = Workflow::activateFromWorkflowType(_workflowTypeName, _recId,
_initialNote, NoYes::No);
            ttscommit;
            // Updates the workflow button to display Actions instead of Submit.
            args.caller().updateWorkflowControls();
        }
        catch(exception::Error)
        {
            // ToDo Insert your error code here.
        }
    }
}
}

```

6. Close the Editor window, and select **Yes** to save your changes.

#### NOTE

When you save this code, you will receive an "Empty compound statement" warning message in the **Compiler Output** window unless you add valid code in the `catch(exception::Error)` block.

## See also

[Activating a workflow](#)

[Create a new workflow type](#)

[Workflow::activateFromWorkflowType method](#)

[Workflow::activateFromWorkflowSequenceNumber method](#)

[Workflow::activateFromWorkflowConfigurationId method](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Associate a workflow document class with a workflow type

2/18/2021 • 2 minutes to read • [Edit Online](#)

To create a workflow, you must bind a workflow document class to the workflow type. The workflow document class contains references to the table data fields that the workflow uses. This topic describes how to bind a workflow document class to a workflow type.

## NOTE

If you used the **Workflow** wizard to create the workflow type, the wizard has already bound the workflow document class to the workflow type.

Before you begin the following procedure, you must create a workflow document class to expose the table fields that are used for conditions in the configuration user interface (UI). For more information, see [Create a workflow document class](#).

## Bind a workflow document class to a workflow type

1. In Application Explorer, expand the **Workflow** node.
2. In the **Workflow Types** node, right-click the workflow type that you want to bind a workflow document class to, and then select **Properties**.
3. On the **Properties** sheet, set the **Document** property to the workflow document class that defines the workflow document.

## See also

[Create a new workflow type](#)

[Create a workflow document class](#)

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Workflow types report

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic points you to a report that lists the available workflow types.

To create a workflow, you must first select the *type* of workflow that you want to create. For example, you may want to create a **Purchase requisition line review** workflow to define who must approve line items on purchase requisitions. For more information about creating a workflow, see [Create workflows overview](#).

## View the report

The **Workflow types report**, included with the [Technical reference reports](#), lists each type of workflow that is available. The report also describes what each type of workflow is used for and indicates whether the workflows of each type are associated with a specific company in the organization or with the whole organization.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Workflow FAQ

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic answers frequently asked questions about the workflow system.

## Why are multiple notifications received when a work item is rejected?

When a work item is rejected, that work item is completed as rejected. Another work item is created and assigned to the originator. This means that there is a notification to the originator for the rejected work item, and a separate notification to the user assigned to the new "change requested" work item.

Each notification is for a different work item, but the similarity can cause confusion. We are looking at ways to improve this in a future release.

## Why are my workflow exports failing?

There is currently a limitation in the workflow export feature that prevents workflow names from exceeding 48 characters. Using a name that is longer than 48 characters can result in a "Server failed to authenticate the request" error and/or prevent a file to be exported without a file type. The following blog post provides more details, [Workflow export troubleshooting](#).

## Can the submitter of a workflow also approve the workflow?

Yes, a submitter of a workflow can also approve the workflow if it is configured that way. To prevent this behavior, set **System administration > Workflow > Workflow parameters > General > Approver > Disallow approval by submitter** to Yes.

## Can I add alerts to workflows to provide notifications to users?

Here are a few key areas to note about adding alerts to workflows to provide notifications:

- Alerts versus workflow notification mechanisms
  - Alerts can be set up for record changes. Workflows change records, so it's possible to set up an alert related to a record change caused by a workflow. However, because workflows have different built-in notification options, using alerts isn't ideal.
- Standard notifications from workflows
  - Workflows have built in email notifications. A customer can configure the notifications so that the users are sent emails. Those notifications don't result in Action Center messages for users.
  - In a future update we will be adding an Action Center message so a user is assigned a workflow work item.
- Adding notifications to workflows
  - Action Center messages can be created for specific users, such as a message created from a workflow in X++.
  - [Workflows have business events](#) that the customer could use to trigger Flows have the notifications that they are looking for.

In summary, if a user does not get the proper notification from the Action Center when they are assigned a workflow work item, then leverage [Workflow business events](#) with Microsoft Power Automate to provide additional or different notifications.

## Why is workflow editor not able to start under AD FS?

When running under Active Directory Federation Services (AD FS) in an upgraded environment, the workflow editor may have trouble starting. If it does, make sure that the URL "https://dynamicsaxworkflweditor/" is added to the property **Microsoft Dynamics 365 for Operations On-premises - Workflow - Native application** in the ADFS settings.

## Why am I getting SQL deadlocks on workflow processing?

The default field value for the **Number of workflow items per batch** on the **Workflow parameters** page is 0. A value of 0 causes the default to change to 20 items per batch. Be careful when adjusting this value because a high number of items per batch (> 40) can cause SQL deadlocks.

## What is the Workflow Enhanced Error feature?

The Workflow Enhanced Error feature in version 10.0.13 adds error codes to differentiate different classes of workflow errors. The error messages reported will be mostly similar with minor differences to make them clearer.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Alerts overview

2/18/2021 • 4 minutes to read • [Edit Online](#)

## About alerts

Alerts form a notification system for critical events in the system. You can use alerts to stay informed about events that you want to track during the workday. You can easily create your own set of alert rules so that you're alerted about deliveries that are overdue, orders that are deleted, prices that change, or other events that you must respond to.

In enterprise resource planning (ERP), there are several typical scenarios where the alerts feature can be used. Here are some examples.

### Scenario 1: Create an alert rule for new sales orders

1. Open the **All sales orders** page.
2. On the Action Pane, on the **Options** tab, in the **Share** group, select **Create a custom alert**.
3. In the **Create alert rule** dialog box, on the **Alert me when** FastTab, in the **Event** field, select **Record has been created**.

### Scenario 2: Create an alert rule for postponement of a delivery date

1. Open the **All purchase orders** page.
2. Select a purchase order ID to access the purchase order details.
3. Expand the **Purchase order header** FastTab.
4. On the Action Pane, on the **Options** tab, in the **Share** group, select **Create a custom alert**.
5. In the **Create alert rule** dialog box, on the **Alert me when** FastTab, in the **Field** field, select **Delivery date**.
6. In the **Event** field, select **has been postponed**.

After you close the **Create alert rule** dialog box, your rule appears on the **Manage alert rules** page. You can use the **Manage alert rules** page to update your existing alert rules. For example, you can modify event triggers, update event notifications, and update expiration dates. To open the **Manage alert rules** page, use the **Alert me** button on the **Options** tab of the Action Pane.

## What occurs when an alert rule is created?

When you create alert rules, you can associate a predefined event with a specific field. For example, the date that is specified in the field arrives, or the contents of the field change. Alternatively, you can associate an event with the records on a specific page. For example, a record is created, or a record is deleted.

When the selected event occurs for the field or for a record on the page, an alert is sent to you. For example, you create a rule where you associate the **Delivery date** field on a specific purchase order line with the **was due this amount of time ago** event. You set the time frame to five days. In this case, an alert is sent five days after the delivery date of that purchase order line.

Additionally, you can refine alert rules by setting conditions. For example, you can be alerted about new purchase orders that are created for specific vendor accounts.

## Preparing for an alert

Before you set up an alert rule, decide when or in what situations you want to receive alerts. When you know which event you want to be notified about, find the page where the data that causes that event appears. The

event can be a date that arrives or a specific change that occurs. Therefore, you must find the page where the date is specified, or where the field that changes or the new record that is created appears. After you have this information, you can create the alert rule.

## Components of an alert rule

An alert rule has five components:

- **Event** – The event that triggers an alert rule can be a date that arrives or a specific change that occurs. You define events on the **Send email alerts for job status changes** FastTab of the **Create alert rule** dialog box.
- **Condition** – On the **Alert me for** FastTab of the **Create alert rule** dialog box, you can select the scope of the condition, to control when you're alerted about events. You can apply the rule either to the current record only or to all visible records on the page. If the rule applies across legal entities, you can set the **Organization-wide** option to **Yes**.
- **Expiry of rule** – On the **Alert me until** FastTab of the **Create alert rule** dialog box, you can specify how long the alert rule should be active.
- **Contents** – On the **Alert me with** FastTab of the **Create alert rule** dialog box, you can specify the subject text and message text that the alert messages should use.
- **User** – On the **Alert who** FastTab of the **Create alert rule** dialog box, you can specify which user should receive the alert messages. By default, your user ID is selected.

### NOTE

This option is restricted to organization administrators.

## Videos

### How to use alerts to monitor filtered data

The [How to use alerts to monitor filtered data](#) video (shown above) is included in the [Finance and Operations playlist](#) available on YouTube.

### Alert rule options

The [Alert rule options](#) video (shown above) is included in the [Finance and Operations playlist](#) available on YouTube.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Create alert rules

2/18/2021 • 5 minutes to read • [Edit Online](#)

## Getting started

Before you set up an alert rule, decide when or in what situations you want to receive alerts. When you know which event you want to be notified about, find the page where the data that causes that event appears. The event can be a date that arrives or a specific change that occurs. Therefore, you must find the page where the date is specified, or where the field that changes or the new record that is created appears. After you have this information, you can create the alert rule.

When you create an alert rule, you define the criteria that must be met before an alert is triggered. Criteria is basically the match between the occurrence of an event and the fulfillment of specific conditions. When an event occurs, the system starts to perform a check according to the conditions that are set up.

## Ensure the alert batch jobs are running

The batch jobs for data change and due date alerts need to be running for the alert conditions to be processed and the notifications to be sent. To run batch jobs, go to **System administration > Periodic tasks > Alerts** and add a new batch job for **Change based alerts** and/or **Due date alerts**. If a long and frequently running batch job is needed, select **Recurrence** and set **No end date** with a **Recurrence pattern** of **Minutes** and a **Count** of 1.

## Events

The event that triggers an alert rule can be a date that arrives or a specific change that occurs. Triggers for events are defined on the **Alert me when** FastTab of the **Create alert rule** dialog box. The events that are available for a particular field depend on the trigger that is selected.

For example, if you're setting up an alert rule for the **Start date** field, due date events are appropriate. Therefore, the **is due in** event type is available for that field. However, for a field such as **Cost center**, a due date event isn't appropriate. Therefore, the **is due in** event type isn't available. Instead, the **has changed** event type is available.

## Event types

Three types of events can occur:

- **Create-type and delete-type events** – These events trigger an alert when a record is created or deleted.
- **Update-type events** – These events trigger an alert when the data in a specific field changes.
- **Due date-type events** – These events trigger an alert when a date arrives.

Changes that occur can be initiated by a user. For example, a user changes the delivery date of a purchase order. Alternatively, changes can occur as part of a process. For example, the **Status** field on a page changes to reflect the life cycle of various processes in the system.

## Conditions

On the **Alert me for** FastTab in the **Create alert rule** dialog box, you can use conditions to control when you're alerted about events.

For example, you can specify that the system should alert you when the status of purchase orders changes, but only if the status matches a specific set of conditions. Specifically, you want to be alerted when the status of a purchase order is set to **Received**. This change in status is the event that triggers the alert.

Next, you must decide which purchase orders you want to be alerted about. For example, you can select one of the following options. These options define the conditions for the alert rule.

- **Current selected record** – You receive an alert when the status of a specific purchase order changes to **Received**.
- **All records** – You receive an alert when the status of a purchase order is changed for an item in the active page view. You can use the advanced filtering that is available on the page to create rules for a specific set of records. For example, you can create an alert that is triggered for all purchase orders for the customers in a specific customer group.

## Expiry of rule

On the **Alert me until** FastTab of the **Create alert rule** dialog box, you can specify how long the alert rule should be active.

## Alert contents

On the **Alert me with** FastTab of the **Create alert rule** dialog box, you can specify the subject text and message text that the alert messages should use.

## User ID

On the **Alert me with** FastTab of the **Create alert rule** dialog box, you can specify which user should receive the alert messages. By default, your user ID is selected. The ability to change the user receiving the alert is restricted to organization administrators.

## Alerts as business events

You can send alerts externally using the business events framework. When creating an alert, set **Organization-wide** to **No** and set **Send externally** to **Yes**. After you have the alert triggering the business event, you can trigger a flow built in Power Automate using the **When a business event occurs** trigger on the Finance and Operations connector, or explicitly send the event to a business events endpoint via the **Business events catalog**.

## Create an alert rule

0. Ensure the alert batch jobs are running (see above).
1. Open the page that contains the data to monitor.
2. On the Action Pane, on the **Options** tab, in the **Share** group, select **Create alert rule**.
3. In the **Create alert rule** dialog box, in the **Field** field, select the field to monitor.
4. In the **Event** field, select the type of event.
5. On the **Alert me for** FastTab, select the desired option. If you want to send the alert as a business event, set the **Organization-wide** value to **No**.
6. If the alert rule should become inactive on a specific date, on the **Alert me until** FastTab, select an end date.
7. On the **Alert me with** FastTab, in the **Subject** field, accept the default subject heading for the email message, or enter a new subject. The text becomes the subject heading for the email message that you receive when an alert is triggered. If you want to send the alert as a business event, set **Send externally** to **Yes**.
8. In the **Message** field, enter an optional message. The text becomes the message that you receive when an

alert is triggered.

9. Select **OK** to save the settings and create the alert rule.

## Limitations and workarounds

### Workaround for creating alerts for the secondary data sources of a form

You can't create alerts for some secondary data sources on forms. For example, when creating alerts on the customer or vendor posting profiles form, only the fields on the header (CustLedger or VendLedger) are available and not the dimension accounts. The workaround for this limitation is to use **SysTableBrowser** to open that table as primary data source.

1. Open the table in the **SysTableBrowser** form.

```
https://<EnvironmentURL>/?cmp=USMF&mi=SysTableBrowser&TableName=<TableName>
```

2. Create an alert from the SysTableBrowser form.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Batch processing of alerts

2/18/2021 • 4 minutes to read • [Edit Online](#)

Alerts are processed by the batch processing functionality. You must set up batch processing before the process and deliver alerts.

Batch processing functionality supports two types of events:

- Events triggered by change-based events. These events are also referred to as create/delete and update events.
- Events triggered by due dates.

You can set up batch processes for each type of event.

## Batch processing for change-based events

The system reads all change-based events that have occurred since batch processing was last run. Change-based events include updates to fields, the deletion of records, and the creation of records. These events are compared with the conditions that you set up in alert rules. When an event matches the conditions in a rule, the batch process generates an alert.

### Frequency for change-based events

For change-based events, you can set up a batch job that triggers the processing of an event soon after the system logs the event. If you set up the batch job to recur more often, users receive their alerts sooner after changes occur. However, a high frequency for batch processing might adversely affect system performance.

On the other hand, a batch job that recurs less often, and that you schedule for times when the system load is low, might help improve system performance. However, a low frequency for batch processing might not meet the users' demands for timely alerts.

Therefore, when you set up the frequency of batch processing for change-based events, consider the balance between the timeliness of alerts and the performance of the whole system. These considerations become more relevant as the number of users who create alert rules increases. The frequency doesn't affect the number of events that the system processes. However, if more users create rules, the process runs more checks. This type of data exchange might affect system performance.

### The risks of low batch frequency

If you set up a low frequency for batch processing for change-based events, data that is relevant to the conditions in alert rules might change before processing. Therefore, you might lose alerts.

For example, you create an alert to trigger when the event is **customer contact changes** and the condition is **customer = BB**. In other words, when the customer contact for customer BB changes, the process logs the event. However, the batch processing system is set up so that batch processing occurs less often than data entry. If the customer name changes from **BB** to **AA** before the event is processed, the data in the database no longer matches the condition in the rule, **customer = BB**. Therefore, when the event is finally processed, no alert is generated.

### Set up processing for change-based alerts

1. Go to **System administration > Periodic tasks > Alerts > Change based alerts**.
2. In the **Change based alerts** dialog box, enter the appropriate information.

# Batch processing for due-date events

The system detects all events that are caused by due dates, and these events are compared with the conditions that are set up in alert rules. The batch process generates an alert when an event matches the conditions in a rule.

## Frequency for due-date events

For due-date events, you might want to set up batch jobs that are run during the night or at specific times of the day, to balance the system load. We recommend that you set up the batch job so that it's run at least one time per day. If alerts should be sent as early as possible, set up the batch processing to occur immediately after the system date changes. If you want to generate alerts for due-date events that occur after a batch job has already processed alerts, you can run the batch job again on the same day.

For example, a batch job has been run on a particular day. You then create a purchase order that has a due date that should trigger an alert on that same day. To receive the alert on that day, you must run the batch job again after the purchase order is created. However, if you don't run the batch job again on that day, the next day's batch job detects any due-date events that weren't processed on previous days.

### NOTE

Even when the batch process is run more than one time per day, alerts aren't duplicated for the same due-date event and conditions. Alerts are generated only for dates that have become due because of changes that occurred in the system after the last batch job was run.

## Batch processing window

The processing of alert rules in a company can be stopped for several reasons. These reasons include vacations, system errors, or other issues that prevent the batch jobs from being run for some time.

To prevent due-date alerts from becoming obsolete because the batch job hasn't been run for several days, you can set up a batch processing window. A batch processing window can be used to prevent a batch job from being run for a specified number of days.

If you set up a batch processing window, an alert is sent when the alert rule is processed, even if the alert exceeds the time limit that is defined in the due-date criteria. An alert continues to be sent for as long as the period that is defined by this time limit plus the batch processing window isn't exceeded. However, when the period exceeds the value defined by the time limit plus the batch processing window, an alert is no longer sent.

## Set up processing for due-date alerts

1. Go to **System administration > Periodic tasks > Alerts > Due date alerts**.
2. In the **Due date alerts** dialog box, enter the appropriate information.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Client alert notifications by email

2/18/2021 • 2 minutes to read • [Edit Online](#)

You can define custom alert rules that monitor filtered views of data and automatically send email notifications when predefined events occur. The option to send email notifications is available for all supported alert types and you can also turn them on for existing alert rules.

You can use built-in controls to create alert rules that monitor the filtered views of system batch jobs. By monitoring the value of the **Status** field, you can also configure alert rules that send email when a batch job fails. After you create these alert rules, you no longer have to check reports for changes to business data. Instead, you can let the intelligent change detection service do the monitoring for you.

Client alerts depend on the email subsystem that is provided through integration with Microsoft Office. We recommend that you use the Simple Mail Transfer Protocol (SMTP) provider, so that email distribution doesn't have to rely on a local mail client.

To send notifications by email, customers must configure integrated email services. Email notifications are sent to recipients on behalf of alert owners.

For more information about how to configure email, see [Configure and send email](#).

The following image shows the **Create alert rule** dialog box, which now includes a **Send email** option.

The screenshot shows the Dynamics 365 interface with the 'Create alert rule' dialog box open. The dialog is titled 'Create alert rule' and is for a rule named 'Customers'. It shows fields for Rule ID (000311) and Table name (Customers). The 'Alert me for' field is set to 'Record has been created in Custo'. The 'Alert me until' field is empty. The 'Alert me with' section includes a Subject field (Record has been created in Custo) and a User ID dropdown (Admin). The Message field is empty. The 'Send email' option is set to 'Yes' with a toggle switch. The 'Email recipients' field contains 'karif@contoso.com'. The dialog has 'OK' and 'Cancel' buttons at the bottom right.




## NOTE

When the **Send email** option is set to **Yes**, alert notifications will continue to be delivered from the Action Center.

## Alert notification email templates

The service sends email notifications by using predefined email templates that deliver the basic details of the alert notification.

The following image shows the structure of the alert notifications when they are received by email.

<p>From: <a href="mailto:owner@domain.com">owner@domain.com</a> Sent: Thursday, November 29, 2018 To: <a href="mailto:target@domain.com">target@domain.com</a> Subject: Alert Notification – Record has been created in Customers</p> <p> Microsoft Dynamics 365</p> <p>Message: Custom alert messages entered by the Alert owner goes here</p> <p>Details: <a href="#">View customer account: Test Cust 02, Test Account 02</a></p>	<p>From: <a href="mailto:owner@domain.com">owner@domain.com</a> Sent: Thursday, November 29, 2018 To: <a href="mailto:target@domain.com">target@domain.com</a> Subject: Alert Notification – Field Customer group in table Customers has changed</p> <p> Microsoft Dynamics 365</p> <p>Message: Custom alert messages entered by the Alert owner goes here</p> <p>Details: <a href="#">View Customer account: DE-001, Contoso Europe</a></p>	<p>From: <a href="mailto:owner@domain.com">owner@domain.com</a> Sent: Thursday, November 29, 2018 To: <a href="mailto:target@domain.com">target@domain.com</a> Subject: Alert Notification – Record has been deleted in Customers</p> <p> Microsoft Dynamics 365</p> <p>Message: Custom alert messages entered by the Alert owner goes here</p> <p>Details: <a href="#">Go to Customers</a></p>
RECORD CREATED	FIELD CHANGED	RECORD DELETED

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Electronic signatures overview

2/18/2021 • 4 minutes to read • [Edit Online](#)

This article provides an overview of electronic signatures and describes how they can be used.

## What is an electronic signature?

An electronic signature confirms the identity of a person who is about to start or approve a computing process. In some industries, an electronic signature is as legally binding as a handwritten signature.

Electronic signatures are a regulations compliance requirement for several regulated industries, such as pharmaceuticals, food and beverage, and aerospace and defense. They are also required for compliance with regulations in 21 CFR Part 11 that was issued by the Food and Drug Administration (FDA) in the United States.

### NOTE

An electronic signature by itself isn't the same as a digital signature. An electronic signature is just a substitute for a handwritten signature, whereas a digital signature provides additional security measures. A digital signature can help identify whether another user or process has tampered with the data. A digital signature can also be verified, and this verification can't be refuted by the owner of the certificate that was used to sign the data. As described below, electronic signatures have built-in digital signature functionality.

## Electronic signatures

You can use electronic signatures for critical business processes. Some processes have built-in electronic signature capabilities. You can also create custom signature requirements for any database table and field.

Electronic signatures have built-in digital signature functionality. Every user who signs documents must obtain a valid cryptographic certificate. When a document is signed, the private key that is associated with that certificate is validated. Electronic signature information is recorded in a log to provide an audit trail. To set up electronic signatures, see [Set up electronic signatures](#).

## Users who require access to electronic signatures

Three kinds of users typically require security access to electronic signatures: electronic signature administrators, signers, and electronic signature auditors.

### Electronic signature administrator

The electronic signature administrator sets up signature requirements, general parameters, and approvers, and receives alerts when signatures can't be verified. By default, a user who belongs to the **Information technology manager** security role has permission to administer electronic signatures.

### Signer

A signer provides electronic signatures for documents and processes that require signatures. By default, a user who belongs to the **System user** security role has permission to sign documents electronically.



#### NOTE

The signer might require additional permissions before access is granted to data that is related to the document or process that is being signed. A user who changes data and must then sign for those changes must have permission to change the data. A user who signs on behalf of another user might not require access to the data. An example of this kind of user is a supervisor who signs for an employee's changes.

### Electronic signature auditor

The electronic signature auditor reviews the database log and the signature review log that is available from the database log. By default, a user who belongs to the **Information technology manager** security role has permission to audit electronic signatures.

If you use a role other than **Information technology manager**, make sure that the role is assigned the following privileges:

- View electronic signature failures
- View database log

## Signing documents electronically

### Get a certificate

Before you sign documents electronically, you must request a certificate.

#### NOTE

Microsoft SQL Server features are used to create certificates and enable electronic signing. No additional certificate or public key infrastructure (PKI) is required.

When you request a certificate, a public key and a private key are created for you. The private key is encrypted by using a password that is known only to you. When you sign a document electronically, your identity is verified when you enter the password.

To request a certificate, on the **Options** page, on the **Accounts** tab, click **Get certificate**.

You must enter and confirm the password that you will use for signing. The password is used to protect your private key and authorize the use of your certificate. This password isn't stored in the database, and it isn't available to anyone else, not even to the administrator.

If you forget the password that is connected with your certificate, that certificate must be reset. If you reset the certificate, you don't affect documents that you signed by using the previous certificate. To reset the certificate, on the **Options** page, click **Reset certificate**.

### Sign a document electronically

The **Sign document** page is displayed when you make a change that requires an electronic signature.

1. On the **Sign document** page, click the **Document** tab to review the changes to the document.
2. On the **Signature** tab, select a reason code.
3. Enter a comment, if a comment is required.
4. If your user ID doesn't appear in the **Signer** field, select it in the list.
5. Enter your location, if this information is required.
6. Click **OK**.

### Sign for another user's changes

Occasionally, you might want a user to sign for another user's changes. For example, a supervisor might be

required to sign for changes that an employee makes to a bill of materials (BOM). Use this procedure to designate a user as a signer for another user.

**NOTE**

When one user signs for another user's change, the signature must be provided at the workstation of the user who made the change. The user can't save the change until the signature has been provided.

To designate approvers, follow these steps.

1. On the **Options** page, on the **Accounts** tab, click **Designate approver**.
2. In the **Approver user ID** field, select the ID of the user who must sign for another user's changes.
3. In the **Sign for user ID** field, select the ID of the user whose changes must be signed for.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up electronic signatures

2/18/2021 • 2 minutes to read • [Edit Online](#)

Use this procedure to set up electronic signatures. An electronic signature confirms the identity of a person who is about to start or approve a computing process. The demo data company used to create this procedure is DAT.

## Enable the Electronic signature configuration key

1. Go to System administration > Setup > License configuration.
2. In the tree, expand 'Administration'.
  - Verify that the Electronic signature check box is selected.
  - If the Electronic signature check box is not selected, you must enable maintenance mode. Maintenance mode can be enabled in this environment by running a maintenance job from Lifecycle Services, or by using the Deployment.Setup tool locally.
3. Close the page.

## Set up electronic signature parameters

1. Go to Organization administration > Setup > Electronic signature > Electronic signature parameters.
2. Click Edit.
3. In the Notice field, type a value.
  - Enter the notice that signers will receive when a signature is requested. You can enter any text. Typically, this text tells the user what it means to sign a document electronically.
  - If you want to enter the Notice text in additional languages, click the Translations button.
4. Click Save.
5. Close the page.

## Set up reason codes for electronic signatures

1. Go to Organization administration > Setup > Electronic signature > Electronic signature reason codes.
2. Click New.
  - You must set up reason codes before using electronic signatures. A valid reason code is required when signing a document. A signer selects a reason code to indicate the purpose of an electronic signature. For example, a reason code could be used to indicate legal approval.
3. In the Reason code field, type a value.
4. In the Description field, type a value.
  - Enter additional reason codes, if needed.
5. Click Save.
6. Close the page.

## Require electronic signatures for existing processes

1. Go to Organization administration > Setup > Electronic signature > Electronic signature requirements.
2. In the list, find and select the desired record.
  - Select a process that requires electronic signatures.
3. Select or clear the Signature required check box.
  - Repeat these steps for each process that requires electronic signatures.

4. Click Save.

## Create a custom requirement for electronic signatures

1. Click New.

2. Select or clear the Signature required check box.

3. In the Name field, enter a name for the process that requires electronic signatures.

4. In the Table name field, click the drop-down button to open the lookup.

5. In the list, find and select the table where the data that must be signed is stored.

6. In the list, click the link in the selected row.

7. In the Field name field, click the drop-down button to open the lookup.

8. In the list, find and select the field in the table that you want to monitor.

9. In the list, click the link in the selected row.

- Specify when a signature is required. Select Always if a signature is required when the data in the field changes. Select Only if a signature is required only under certain conditions. If you select Only, you must also select one of the following options: When a record is inserted, When a record is updated, or When a record is deleted.

10. Click Save.

11. Close the page.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Case management overview

2/18/2021 • 6 minutes to read • [Edit Online](#)

By planning, tracking, and analyzing cases, you can develop efficient resolutions that can be used for similar issues. For example, when customer service representatives or Human Resources generalists create cases, they can find information in knowledge articles to help them work more efficiently. The following examples show how cases can be used for different situations in an organization.

## Example: How Fabrikam uses cases for customers in the private sector

Lisa, a customer service representative at Fabrikam, receives a telephone call from Lionel, a Fabrikam customer. Lionel is having trouble setting the correct volume level on the new sound system that Fabrikam installed in his music store. Lisa creates a case for Lionel and assigns the **Volume** category to the case. Lisa then elevates the priority and assigns a one-day service level agreement (SLA) to the case.

Lisa also enters the case details in the case log. Lisa notices that several knowledge articles are associated with the **Volume** category, and that three of them are marked as helpful for resolving cases. Lisa opens each article and discusses the resolution steps with Lionel, but none of the solutions solve the issue that Lionel is having with the new sound system. Lisa tells Lionel that an audio technician will call him within 24 hours and work to solve the issue.

Lisa activates the case, and a set of activities is created. She assigns the activities to Terrence, a member of the audio engineering team. Terrence sees that new activities are assigned to him. He opens the case and reads the case log to learn more about the case. Terrence encountered the same issue the day before, and he developed a solution. Terrence contacts Lionel and offers this solution for the issue. Terrence also enters it in the case details.

Because his solution is successful, Terrence decides to document it, so that other people can use it if they encounter the same issue. Terrence adds the document to the **Knowledge article** page, assigns the document to the **Volume** category, and manually elevates the ranking, so that other Fabrikam employees will know that it's a successful solution. Terrence then elevates the case to the next level. By elevating the case, he creates a new activity for Marie, who is a quality assurance representative in the customer service department.

Marie sees that a new activity is assigned to her, and opens the case that is associated with the activity. Marie reviews the case and the case details to make sure that the correct process was followed for the case. Marie verifies that the actual case time did not exceed the time frame that was estimated in the SLA. Marie notes that Terrence contacted the customer, and that the issue was resolved. Marie is satisfied with the treatment that the customer received and the results of the case. She resolves the case as closed. When Marie closes the case, the open activity that is assigned to her is also closed.

## Example: How City Power & Light uses cases for customers in the public sector

Annie, a customer service representative with City Power & Light, receives a telephone call from a resident of the city that City Power & Light serves. Annie records the call as an activity and takes notes of the conversation. The resident tells Annie that his house has no power. Annie informs the resident that City Power & Light will investigate, find, and resolve the issue as quickly as possible.

Annie then creates a case, associates the telephone call with the case, and creates a service order. Annie knows that other residents are likely to call to report a power outage. Therefore, to avoid overwhelming the customer service center, and to save time, Annie sends a group instant message (IM) to inform the other representatives about the issue, and to tell them that a case and service order have been created. Annie includes the case

number and service order number in her IM. Then, if City Power & Light receives more telephone calls about the power outage, the customer service representatives can create an activity for each telephone call and assign it to the existing case.

## Example: How Fabrikam uses cases for employees

The following scenarios show how Fabrikam Human Resources generalists in different locations can use case management when they address issues for employees.

### In Great Britain

Christine, the Human Resources generalist for the Great Britain division of Fabrikam, receives a telephone call from Peter, a Fabrikam employee. Peter informs Christine that nine weeks ago, immediately after the birth of his son, he changed the number of dependents on his tax withholdings. Peter wants to know why the changes haven't become effective. Christine creates a case for Peter. She reviews Peter's tax information and learns that, although Peter entered new dependent information, he didn't select a start date for the new tax withholdings. Christine sends an email message to inform Peter that he must select a start date and resubmit his changes. Peter replies to Christine's message to tell her that he has now selected a start date and resubmitted his changes. Christine attaches the email message from Peter to the case record, verifies that the correct changes were made and submitted, and closes the case.

### In the United States

Luke, the Human Resources generalist for the United States division of Fabrikam, receives an email message from Shannon, a Fabrikam employee. Shannon is a machine operator who was injured on the job six months ago. Since then, she has been working with Humongous Insurance to have her medical expenses paid. Because Shannon contacted Luke about this issue four weeks ago, a case has already been created. Shannon's new email message explains that Humongous Insurance is still not returning her telephone calls. Luke opens the existing case, adds Shannon's email message as a document, and reviews the case log. When Luke created this case, he assigned the **Insurance** category to it. He now sees that there is a new knowledge article that is associated with the **Insurance** category. Luke reads the knowledge article and learns that all phones at Humongous Insurance are down while the company's telephone system is being updated. The article states that the insurance company sent an email message to all its customers, but that several customers did not receive the message because of a problem with the company's email system. All customers who have active insurance claims are asked to send their inquiries to Humongous Insurance by email or paper mail. Luke sends Shannon an email message that explains what she must do to have her insurance claim settled. He also ranks the knowledge article that he read as a helpful source of information. Luke creates another activity for himself, so that he can follow up with both Shannon and Humongous Insurance in four weeks to make sure that the claim has been resolved. After four weeks, Luke contacts Shannon. He learns that Humongous Insurance has paid her claims, and that she is happy with the resolution. Luke changes the status of the case to **Closed**.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Plan case category security, case processes, and case categories

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article describes the considerations and decisions that you must make during the planning process, before you begin to configure cases.

You can use the case functionality to resolve both issues that are external to your organization and internal issues.

## Case category security by role

Only appropriate employees in an organization should have access to cases and related information. To control which employees have access to view, create, and update different types of cases, you can assign security roles to case category types. You must determine which security roles should have access to the various case category types.

**Decision:** Determine which security roles should have access to the following case category types. Your organization might not use all these category types, so make decisions only for those categories that are appropriate.

- General
- Sales
- Purchase
- Service
- Project
- Production
- Collections
- Audit
- Web
- Human Resources
- Product change
- FMLA

## Case processes

You should set up processes that employees must follow for the cases that are opened in your organization. Processes help guarantee consistency for the people who are involved in cases, and also help employees resolve cases faster and more efficiently. You can set up a process for each case category that cases are assigned to. Although planning a separate process for each case type takes time, case resolution will go much more smoothly if the processes are planned out.

**Decisions:** For each case process, you must make the following decisions:

- What are the name and description of the process?
- Is the process active, and should employees use it when they handle a case that the process is assigned to?
- Who in the organization will be responsible for applying the process to a case? For example, Cost accounting or Human resources might be responsible for some case processes. Note that multiple areas can be responsible for completing one process.

# Case categories

The Case category hierarchy provides a list of categories that you can assign cases to (see the "Case category security by role" section). Each top-level category includes subcategories, so that you can create more specific categories for the cases that your organization works with. Review the list of existing categories and subcategories to determine whether you must create more. If you must create more categories and subcategories, you must make the following decisions for each addition.

## Decisions:

- Are you creating a new top-level category?
  - What is the name of the category
- Are you creating a new subcategory?
  - What is the parent category of the subcategory?
  - What is the name of the subcategory?
  - Which worker will own the subcategory?
  - What department is the subcategory assigned to?
  - What case process will be assigned to this subcategory?
  - What is the default service level agreement (SLA) that is assigned to this subcategory?
  - Should an activity be created when a case that this subcategory is assigned to is opened?
  - If so, what are the activity category, type, purpose, and phase?
  - Should a follow-up activity for the case be created?
  - If so, what are the follow-up activity category, type, purpose, and phase?

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Record templates overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article introduces the concept of record templates and explains how they can be used to create records that share information.

Record templates can help you to create records more quickly, however you can only create record templates for some record types.

For example, imagine you are entering rental information for a car rental business that is located in San Francisco. Since most of the customers are likely to be from the San Francisco area, it would be nice if you could automatically fill in the values for the **State**, **Country**, and **City** fields on the rental form.

## NOTE

You can apply templates only in areas that you have access to. However all template titles are visible to you when you create a new record, and to other users as well, if you are creating templates that will be available for all users. Be sure to consider this when naming templates. For example, avoid using names that include words, such as "commission," if is confidential that some employees in the company have commission-based salaries.

When one or more templates that you have access to exist for a specific form and you attempt to create a new record in the form, the **Select a template for** page is displayed. When you select a template from the list, the new record is created and contains default information that is based on the template that you selected. If you do not want to use templates when you create new records, select the **Do not ask again** check box in the **Select a template for** page. To display the template selection dialog box again, right-click any record, click **Record info**, and then click **Show template selection**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a record template to facilitate data entry

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic demonstrates how to create a record template so that field values that are used often do not have to be entered explicitly for each new record. In this procedure, you'll create a new record for new laptops that should be added to your fixed assets. This procedure uses the USMF sample company.

1. In the navigation pane, go to **Modules > Fixed assets > Fixed assets > Fixed assets**.
2. Select **New**.
3. In the **Fixed asset group** field, enter or select a value.
4. In the **Name** field, type a value. For example, enter **Corporate lead laptop**.
5. In the **Search name** field, type a value. For example, enter **laptop**.
6. Expand the **Technical information** section.
7. In the **Make**, **Model**, and **Model year** fields, type values.
8. On the Action Pane, select **Options**.
9. Select **Record info**.
10. Select **User template**.
11. In the **Name** field, type a value.
12. In the **Description** field, type a value.
13. Select **OK**.
14. Select **Close**.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Use record template to create a new record

2/18/2021 • 2 minutes to read • [Edit Online](#)

This procedure shows how to use a previously defined record template to create a new record. To complete this procedure, you must first complete the "Create a record template to facilitate data entry" procedure.

This procedure uses the USMF company.

1. In the **Navigation pane**, go to **Fixed assets > Fixed assets > Fixed assets**.
2. Click **New**. You will be prompted to select a template. Select the one that corresponds to your business need.
3. In the list, find and select the desired record.
4. Click **OK**.

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure document management

2/18/2021 • 15 minutes to read • [Edit Online](#)

This topic explains how to configure document management (document handling) so that it stores file attachments and notes for records. It includes information about the concepts and features that are involved in this functionality.

To learn more about document management, watch the short [Document Management](#) video.

## Configure document types

Document types are used to categorize the documents that you attach to records or the templates that you create. Each document type can be stored in a unique location.

A default set of document types is provided. You can use these document types to categorize an attachment as a file, image, note, or URL. The **File** and **Image** default document types are configured to use **Azure storage** as the location.

To create a new document type, follow these steps.

1. Go to the **Document types** page.
2. Click **New**.
3. In the **Type** field, enter a short name for the new document type, such as **SharePoint** or **HR Docs**.
4. In the **Name** field, enter a longer name, such as **SharePoint files** or **HR Docs**.
5. In the **Class** field, specify a class to define the behavior for the document type:
  - **Attach file** – The user is prompted for a file.
  - **Attach URL** – The user can enter a URL in the **Notes** field, such as `https://www.microsoft.com`. The **Open** button on the **Attachments** page will open the URL on a browser tab.
  - **Simple note** – The user can add a simple note in the **Notes** field.
6. If you specified **Attach file** in the **Class** field, in the **Location** field, specify the storage mechanism to use.
7. If you specified **SharePoint** in the **Location** field, specify the Microsoft SharePoint address in the **SharePoint Address** field. To do this, click the **Edit** button (pencil symbol) and use the **Folder selection** dialog box.

## Configure SharePoint storage

Microsoft SharePoint Online is one of the storage locations that are supported natively. Currently, only SharePoint Online is supported. Support for on-premises SharePoint (a local SharePoint server) may be added in the future.

To use SharePoint storage, set the **Location** field for a document type to **SharePoint**. Then, in the **SharePoint Address** field, enter a valid SharePoint address.

To configure SharePoint storage, follow these steps.

1. Go to the **Document management parameters** page.
2. On the **SharePoint** tab, in the **Default SharePoint server** field, review the host name that was

automatically detected for the SharePoint site, such as contosoax7.sharepoint.com. Typically, the SharePoint host name is in the form tenantname.sharepoint.com, and accounts on that tenant are in the form `user1@tenantname.onmicrosoft.com`.

Typically, if no default SharePoint server is specified, either there is no SharePoint site for the tenant, or a valid Microsoft 365 license isn't associated with the current user (the admin).

3. Optional: Click **Test SharePoint connection** to test the specified SharePoint host name. This verifies that the security and license are working correctly.
4. Optional: Click **Open SharePoint** to open the specified SharePoint host name in a browser. Note that this does not verify security, it just opens the SharePoint path in a browser tab for easy exploration.

### Troubleshooting SharePoint communication

SharePoint communication works for the current user only if the following conditions are met:

- A Microsoft 365 license is associated with the user's account.
- The user is a typical user on the tenant, not an external user (for example, a user from another tenant).
- There is a SharePoint site for the tenant (for example, Contoso.SharePoint.com).
- The user has access to the folder that the document is stored in.

If documents stored in SharePoint don't open or don't display in preview, follow these steps to troubleshoot the issue:

1. Verify the Admin account has an associated email account (verify or change this in the **User** page). If this isn't set up, you need to add the email and provider via the OData Excel add-in. By default, the email address isn't present in the Excel design. The user needs to edit the Excel design, add all fields, apply, and refresh. Once complete, you can update the Admin account.
2. After the Admin account has an associated email account, sign in to Dynamics as the admin.
3. Open an attachment that is stored in SharePoint.
4. Sign in with another user account that has read access to the attachments page and the configured SharePoint folder. Verify that they can also open and preview the attachment.

## Configure file types

By modifying the list of file extensions that are allowed, you can control the types of files that users can attach to records.

To specify file types, follow these steps.

1. Go to the **Document management parameters** page.
2. On the **File types** tab, review the default file types.
3. Remove any file types that users should not be able to attach to records, and add any file types that users should be able to attach to records.

## Configure document preview

The attachments preview uses the Web app Open Platform Interface (WOPI) that is provided by Microsoft Office Online Server. On the **Document management parameters** page, on the **General** tab, in the **Office Web Apps Server** field, specify the Office Online Server instance to use for attachment previews. The default value is `https://onenote.officeapps.live.com`. This value points to the cloud-based WOPI server.

### For a Microsoft Dynamics 365 Finance + Operations (on-premises) environment

The default cloud-based WOPI server in Finance + Operations can't read the attachment file to provide a

preview. If previews are required, you must [install an on-premises Office Online Server instance](#) and configure it inside the environment. Set the **Office Web Apps Server** field to the host name of the installed Office Online Server instance, and then click **Save**.

If previews aren't required, set the **Office Web Apps Server** field to `https://localhost`. The preview will then show the message "No preview available" instead of an error message.

### **Document preview (WOPI) will not work in environments with an IP safe list enabled**

Document preview (WOPI) will not work in environments with an IP safe list enabled, because the WOPI service that provides the preview will not be able to connect back to the file service to retrieve the file for rendering.

## Other configuration

Here are some other configuration options to consider, although these options are rarely used:

- On the **Document management parameters** page, on the **General** tab, you can use the **Use Document Tables** option to enable the **Active document tables** allow list. If you set this option to **Yes**, you disable attachments on all other tables. Therefore, turn on this option only when it's required.
- On the **Document management parameters** page, on the **General** tab, you can use the **Maximum file size in megabytes** field to set the maximum file size for attachments. Note that the ability of users to provide files is also constrained by the file size limit that is set for the environment in configuration files. These configuration files can't be changed via a client page.
- On the **Options** page (**Settings > User options**), on the **Preferences** tab, you can use the **Enable document handling** option to disable document handling (document management).

## Accessing document management attachments

Document management appears to users as the **Attach** button at the top of most pages that contain data. When you select the **Attach** button (or when you use the corresponding keyboard shortcut, **Ctrl+Shift+A**), the **Attachments** page is opened in the context of the data source of the control that is currently selected on the page. This page shows all the attachments that are related to the corresponding data source.

The **Attach** button also shows a count of the attachments for the currently selected record. Therefore, you can determine whether there are attachments for the current record without having to open the **Attachments** page. The button shows exact counts for zero through nine attachments. If there are more than nine attachments, the button shows **9+** as the count. In this way, the performance impact and visual noise that exact larger counts might cause are reduced.

In version 10.0.12, the **Show related document attachments** feature changes the document attachment experience in two ways. First, when the feature is enabled, the **Attachments** page doesn't show only attachments that are related to a single data source. Instead, it shows attachments from all data sources on the page that are related to the active record. The count of attachments on the **Attach** button also reflects this change. Second, users can move and copy attachments between the related data sources on the **Attachments** page.

## Document attachment history

Starting in version 10.0.16/Platform update 40, a history mechanism has been made available for record attachments. This allows your organization to maintain an audit of actions related to individual attachments. In particular, you can see when an attachment was created, marked for pending deletion, restored, deleted, or moved and who performed that action. Note that attachment history is not maintained until 10.0.16/Platform update 40, so any actions on attachments prior to that version will not be available.

### **Configuration of document attachment history**

Document attachment history can be enabled (or disabled) by going to **Document management parameters**

> **General** > **History** > **Enable document history**. The default history retention period is 180 days, but this value can be modified as needed using the **Number of days to retain history** field.

### Viewing an attachment's history

There are two entry points for viewing the history of a record attachment:

- When you are looking at the attachments for a specific record (see the [Accessing document management attachments](#) section for more details), you can view the history for the current set of attachments on the **Attachments** page by selecting the **Document history** button in the Action pane.
- Administrators can select the **Document history** button in the **History** section of the **Document management parameters** page. This action opens the **Document history** page, which shows a list of all attachments in the system. You can then drill into any record to see the detailed history for the selected attachment.

## Attachment recovery

In Platform update 29, an attachment recovery feature has been added that provides a recycle bin for record attachments to be recovered within a configured period of time.

### Configuration of attachment recovery

Attachment recovery can be enabled by going to **Document management parameters** > **General** > **Deferred deletion** > **Deferred deletion enabled**. The default for **Number of days to defer deletion** is 30 days but can be changed as needed. If the **Number of days to defer deletion** value is zero, this means that the deleted attachments will be recoverable for an indefinite period.

After attachment recovery is enabled, a batch job with this name will be created, "Scans for deleted references which have reached the end of their retention period". This batch job will use the **Number of days to defer deletion** to determine how long to retain a deleted attachment based on the **Deleted data and time**.

### Deleting attachments when attachment recovery is active

When a user deletes an attachment, a notification will be added to the Message Center to provide confirmation of the deletion and an option to undo to the action if the deletion was unintended.

Table extension support has been built-in, so that any extension or custom field values on the **DocuRef** or **DocuValue** tables will be retained to enable their recovery.

### Recovering attachments

When attachment recovery is enabled, attachments can be recovered in one of three ways:

1. Immediately after deletion, the user can use the undo link in the **Attachment deleted** notification.
2. On the **Attachments** page, a **Deleted attachments** button provides access to the list of deleted attachments that can be recovered for a particular record. The deleted attachments can be opened for review, permanently deleted, or restored.
3. In **System administration** > **Inquiries**, the **Deleted attachments** page provides access to the list of deleted attachments that can be recovered for any record. The deleted attachments can be opened for review, permanently deleted, or restored.

## Scanning attachments for viruses and malicious code

When you work with attachments, you might want to scan the files for viruses and malicious code. Therefore, in version 10.0.12 and later, extension points are available so that customers can integrate with the file scanning software of their choice when they work with attachments. A similar extension point is also available for file upload. For more information, see [File upload control](#).

### IMPORTANT

Out of the box, Finance and Operations apps don't scan files for viruses and malicious code, and we don't recommend specific software for file scanning. Instead, customers are responsible for choosing their own file scanning software, and for adding the appropriate code to the delegate handlers so that they can use the software or service of their choice to scan files.

The **Docu** class exposes the following two delegates. Handlers can be implemented for these delegates for document scanning purposes:

- **Docu.delegateScanDocument()** – This delegate applies the file scanning logic when a new document attachment is uploaded, or when a user tries to preview or download an existing attachment. The corresponding action will fail if the scanning service determines that the file is malicious.
- **Docu.delegateScanDeletedDocument()** – This delegate applies the file scanning logic to documents in the attachments recycle bin when a user tries to preview or download a file. The corresponding action will fail if the scanning service determines that the file is malicious.

### Implementation details

The following example of the **ScanDocuments** class shows boilerplate code for the two handlers. For general information about how to implement handlers for delegates, see [EventHandlerResult classes in request or response scenarios](#).



```

public final class ScanDocuments
{
    [SubscribesTo(classStr(Docu), staticDelegateStr(Docu, delegateScanDocument))]
    public static void Docu_delegateScanDocument(DocuRef _docuRef, EventHandlerRejectResult
_validationResult)
    {
        if (!ScanDocuments::scanDocument(_docuRef))
        {
            _validationResult.reject();
        }
    }

    [SubscribesTo(classStr(Docu), staticDelegateStr(Docu, delegateScanDeletedDocument))]
    public static void Docu_delegateScanDeletedDocument(DocuDeletedRef _docuDeletedRef,
EventHandlerRejectResult _validationResult)
    {
        if (!ScanDocuments::scanDeletedDocument(_docuDeletedRef))
        {
            _validationResult.reject();
        }
    }

    private static boolean scanDocument(DocuRef _docuRef)
    {
        /*
        Custom implementation required for connecting to a scanning service
        If document scanning process found an issue, return false; otherwise, return true;
        */
        return true;
    }

    private static boolean scanDeletedDocument(DocuDeletedRef _docuDeletedRef)
    {
        /*
        Custom implementation required for connecting to a scanning service
        If document scanning process found an issue, return false; otherwise, return true;
        */
        return true;
    }
}

```

## Frequently asked questions

### **What is the difference between document handling and document management?**

There is no difference between document handling and document management. Both terms refer to the same functionality. Different terms are used in different versions of the product.

### **What is the difference between document management and print management?**

Document management lets you add notes, documents, and other files to records.

Print management lets you control print settings for selected reports. Print settings include the number of copies, the printer destination, and the multilanguage text that can be included on the report. For more information, see [Document Reporting Services](#).

### **What is the difference between document types and file types?**

Document types are used to categorize the documents that you attach to records or the templates that you create. Each document type can be stored in a unique location. The table for document types is named DocuType.

File types include Microsoft Word documents and images. A file type is denoted by the extension of the file, such

as .txt, .png, .doc, .xlsx, or .pdf.

### **Does document management integrate with Microsoft 365?**

Yes. SharePoint storage is supported natively and can be selected as the storage location for a document type. In addition, any URL addressable file can be made an attachment via the URL document type.

### **How does the default storage location for Document Management change in Finance + Operations environments?**

For Finance + Operations, the Azure Blob storage provider for attachments is replaced by a file folder storage provider so that attachments are kept on-premises instead of being stored in the cloud. Therefore, the default storage location for attachments is a file folder.

### **If I accidentally delete an attachment stored in Azure Blob storage, can it be restored?**

If an attachment stored in Azure Blob storage is accidentally deleted, it cannot be restored or recovered because it has been permanently deleted and the reference to it has also been deleted.

### **Is the database information about attachments stored separately from the attachments themselves?**

Record attachment information is stored in the DocuRef and DocuView tables. The DocuRef table is the record that represents the attachment. The DocuRef record points to the record being attached to and to a DocuView record. The DocuView record points to the file that is the attachment. Files are stored outside the database, therefore any database operations, like restoring from backup, will only affect the database information about the attachment, not the attachment file itself.

### **Can attachments be stored in the database?**

No. By default, attachments are stored in Azure Blob storage.

### **What are the main differences between Azure Blob storage and database storage?**

Database storage is Azure SQL Database. File storage is Azure Blob storage. Azure Blob storage is simpler and much less expensive.

### **How much storage do we get for Azure Blob storage?**

That information is in the [licensing guide](#). Currently, you get 40 gigabytes (GB) of storage.

### **What is the cost for additional storage?**

The cost for additional storage varies, but it's similar to the [standard Azure costs for storage](#). In other words, the cost is about \$0.05 per GB.

### **How can we learn how much storage we've already used?**

There will be proactive communications when you're approaching your database and file storage limits. However, Microsoft Dynamics Lifecycle Services (LCS) provides some information, and you can log support requests for additional information.

### **Is there an option to export all document attachments from the system?**

Although attachments can be exported, that capability isn't a standard capability, because there isn't a standard attachment entity. Entities that provide attachments for a specific business document or record must be built.

### **How can attachments be extracted from the system?**

To extract attachments, an Attachments entity must be built for a specific business document or record. There isn't a standard attachment entity because the identity for each record type is different. To learn how to build an Attachments entity, you can find examples in the Application explorer by searching for "Attachment" under the AOT > Data Model > Data Entities node.

### **How does the document preview work for attachments stored in SharePoint?**

The files are retrieved from SharePoint using the current user permissions by the WOPI service. Those files are then rendered in HTML to provide a document preview. This means that the current user needs access to the

files to be able to preview them or open them.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and send email

2/18/2021 • 13 minutes to read • [Edit Online](#)

The behavior of the email subsystem is influenced by a combination of administrator configuration, user configuration, and user choices. This topic is divided into sections for administrators and users to make it easy to find relevant information.

Both administrators and users set the behavior of the email subsystem.

## [Administrator] Email parameters page

### Configuration tab

On the **Email parameters** page, note the following settings on the **Configuration** tab.

FIELD	DESCRIPTION
Batch email provider	Specifies which email provider will be used to send emails that are sent by processes in a batch or non-interactive manner. The Exchange provider will use the account associated with the batch process.
Attachment size limit	Specifies the maximum size of a single email that can be sent via the email subsystem.

In Platform update 32, an **Email history** page was added to allow administrators to review all sent emails, including any errors that might have prevented an email from being sent. By default, the last 30 days of email history is retained. This can be configured by changing the **Number of days to retain email history** to a non-zero amount. Zero provides the default amount and behavior.

In version 10.0.16/Platform 40, an **Email throttling** section is visible, if your environment has enabled the **Email throttling** feature in Feature management. This feature allows non-interactive email providers (such as the batch email provider) to adhere to a per-minute sending limit. This can prevent errors from the system attempting to send more emails than the provider allows. The sending limits for Microsoft 365 email providers are set automatically according to [Exchange Online sending limits](#). Manual configuration is required for all other email providers. The per-minute sending limit can be removed from a provider by resetting the **per-minute email sending limit** field to 0.

### SMTP settings tab

On the **Email parameters** page, note the following settings on the **SMTP settings** tab.

FIELD	DESCRIPTION
Outgoing mail server	The host name of the desired SMTP server. <ul style="list-style-type: none"><li>For <a href="#">Microsoft 365 production</a> (including *.onmicrosoft.com accounts) use smtp.office365.com. (You can find this setting at outlook.office.com at <b>Settings &gt; Mail &gt; POP and IMAP</b>.)</li><li>For Outlook/Hotmail use smtp-mail.outlook.com.</li></ul>
SMTP port number	Typically, the port number should be set to 587 for secure transport.

FIELD	DESCRIPTION
User name and Password	Specify, as needed, to send the email via the appropriate mail account. All users need to provide the SMTP account <b>Send As</b> and <b>Send On Behalf Of</b> permissions to enable the ability to send Simple Mail Transfer Protocol (SMTP) mail. You can configure Send As permissions in the Microsoft 365 admin center (portal.office.com/Admin), at <b>Users &gt; Active users &gt; User &gt; Edit mailbox permissions &gt; Send email from this mailbox</b> . For more information, see <a href="#">Enable sending email from another user's mailbox in Microsoft 365</a> .
Specify if SSL is required	Determines whether secure transport is used. Typically, this is <b>Yes</b> , except for internal or troubleshooting scenarios.

## [Administrator] Email distributor batch process

Email that is sent directly from the server, without user interaction, via SMTP is sent by the **Email distributor batch** process. That batch process must be started to process the email queue. To start the process, open the **Email distributor batch** pane (**System administration > Periodic tasks > Email processing > Batch**) and turn on **Batch processing**.

If the Exchange provider is used, then the user account associated with the batch process (usually admin) will be sender.

## [Administrator] User email

The default **send from** address for each user is pulled from the **Email** field on the **Users** page (**System administration > Users > Users**). Administrators can override this **send from** default if needed using the **Sender email** field on the **Options** page.

## [User] Email provider selection section on the Options page

The **Options** page can be opened via **Settings > User options**. The **Email provider selection** section is on the **Account** tab.

FIELD	DESCRIPTION
Email provider ID	Allows the user to select the email provider that should be used when sending an email. Selecting an option here is the equivalent of selecting <b>Do not ask again</b> in the <b>How would you like to send email</b> dialog box. Selecting the blank option <b>Prompt for which email provider to use</b> will cause the <b>How would you like to send email</b> dialog box to display when an email is going to be sent.

FIELD	DESCRIPTION
Sender email	<p>Allows the administrator to provide an email address override for the user in the <b>From</b> field of the email. By default, the email alias that is associated with the user account is used as the <b>From</b> field in new emails, but this user option email address will override that. When sending email via SMTP, the user needs to have appropriate <b>Send As</b> and <b>Send On Behalf Of</b> permissions configured in Exchange or on the SMTP server.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>[!NOTE] You can configure <b>Send As</b> and <b>Send On Behalf Of</b> permissions in the Microsoft 365 admin center (portal.office.com/Admin) at <b>Users &gt; Active users &gt; User &gt; Edit mailbox permissions &gt; Send email from this mailbox</b>. For more information, see <a href="#">Enable sending email from another user's mailbox in Microsoft 365</a>.</p> </div>

## [User] How would you like to send email dialog box (optional)

When an email is going to be sent, the user will see the **How would you like to send email** dialog box that will list the available options for sending email.

FIELD	DESCRIPTION
Use an email app, such as Outlook	Provides the user with a generated email (.eml) file.
Use Exchange email server	Uses the Exchange Online server associated with the tenant. On-premises Exchange servers are currently not supported for the <b>Exchange</b> mail provider.
Use the system email client	Opens the <b>Send email</b> composition dialog box and then sends the resulting email via SMTP.
Do not ask again	If this field is not selected, the next time an email is sent the most recently selected option will be used and the dialog box will not open.

## [User] Send email dialog box (optional)

The **Send email** dialog box is opened to allow the user to edit the contents of the email that will be sent. Some of the following fields will be pre-populated in this window.

FIELD	DESCRIPTION
From	Populated from the <b>Email</b> field on the <b>Options</b> page.
<b>To, Cc, Bcc, Subject, and Body</b>	Populated with values specified by the process that initiated the sending of the email. These fields can be edited as needed by the user.
Attachments list	May be populated with attachments specified by the process that initiated the sending of the email. This list can be edited as needed by the user.

When the email is ready to be sent, the **Send** button will cause the email to be sent via SMTP.

## Usage scenarios to verify if email is configured correctly

### Send mail via a local mail client

Email workflows that are enabled via the SysEmail framework can generate email messages (.eml files) that contain attachments. You can then send these messages via Microsoft Outlook or another email client.

1. In Internet Explorer, navigate to **Accounts receivable > Customers > All customers**.
2. Select **US-008 Sparrow Retail**.
3. Click **Collect > Customer balances > Collections** to open the **Collections** page.
4. Click **Communicate > Email > Statements to contact**.
5. Click **OK** to accept the default values in the dialog box.
6. If you're prompted for the mail option to use, clear the **Do not ask again** check box (you can change this option from the **User options** page), select **Use an email app, such as Outlook**, and then click **OK**.
7. If you're using Internet Explorer on your computer, open the email (.eml) file that is generated. If you're using Internet Explorer on the VM, copy the file to your computer, and open it there.
8. Note the email address in the **To** field and the generated workbook attachment.

### Send mail via SMTP

Email workflows that are enabled via the SysEmail framework can also be created in a simple email dialog box and then sent via Simple Mail Transfer Protocol (SMTP).

1. Go to the **Email parameters** page.
2. Click **SMTP settings**.
3. Set the **Outgoing mail server** to the desired SMTP server:
  - For **Microsoft 365 production** (including \*.onmicrosoft.com accounts) use smtp.office365.com. (Find this setting via outlook.office.com, at **Settings > Mail > POP and IMAP**.)
  - For Outlook/Hotmail use smtp-mail.outlook.com.
4. Set the user name and password to an appropriate email account and password.
5. Leave **SSLRequired** turned on, and leave **SMTP port number** set to **587**.
6. Click **Save**.
7. In Internet Explorer, navigate to **Accounts receivable > Customers > All customers**.
8. Select **US-008 Sparrow Retail**.
9. Click **Collect > Customer balances > Collections** to open the **Collections** page.
10. Click **Communicate > Email > Statements to contact**.
11. Click **OK** to accept the default values in the dialog box.
12. If you're prompted for the mail option to use, select **Use the Microsoft Dynamics 365 for Finance and Operations email client**, and then click **OK**.
13. To receive the test message, change the **To address** to your email address.

Ensure that the account specified in the SMTP settings is able to **Send As** and **Send On Behalf Of** your email account. The easiest way to ensure this is to use your email account in the SMTP settings.

14. Enter a subject and body for the message.

15. Click **Send**. The message should be delivered in one to five minutes.

## [Administrator] Workflow email notifications

Workflow email configuration is a collection of related settings that work in conjunction.

### Workflow email notification setup

1. Verify email settings:
  - a. Go to **System administration > Setup > Email > Email parameters**.
  - b. Verify that SMTP is enabled.
  - c. Set the SMTP mail server settings.
2. Verify that the email batch process is running:
  - a. Go to **System administration > Periodic tasks > Email processing > Email distributor batch**.
  - b. Enable the **Batch processing** option.
  - c. Optionally, adjust the recurrence of the email process:
    - a. Select **No end date** to adjust all recurrences of the email batch process.
    - b. Adjust the count.
    - c. Adjust to run every minute if needed.
3. Verify workflow notification system email templates:
  - a. Go to **System administration > Setup > Email > System email templates** (for system-wide templates).
  - b. Verify that the **Sender email** field is set and valid.
4. Verify workflow notification organization email templates:
  - a. Go to **Organization administration > Setup > Organization email templates** (for organization-specific templates).
  - b. Verify that the **Sender email** field is set and valid.
5. Verify that the user can receive email notifications:
  - a. Go to **Settings > User options**.
  - b. Go to the **Account** tab.
    - a. Set the **Email provider ID** (for example, SMTP).
    - b. Optionally, set a **Sender email** override if the default **send from** address should not be used for the current user.
  - c. Navigate to the **Workflow** tab. Set the option to send notifications in email to **Yes**.
6. Verify that the workflow system will send email notifications:

For each workflow that should have a notification, open the workflow properties in the workflow editor.

  - a. Click **Basic settings**. Adjust the email template for the workflow notifications.
  - b. Click **Notifications**.
    - a. Enable the events for which a user should be notified.
    - b. Set the recipient of the workflow notification for each event notification that is enabled.
  - c. On a workflow approval element for which a user should be notified:



- a. Go to **Properties**.
- b. Enable the events for which a user should be notified.
- c. Set the recipient of the workflow notification for each event notification that is enabled.

### **Workflow email notification testing**

The testing for email notifications is to simply trigger the notification and then check for it.

1. Submit a workflow that has been set up for email notifications.
2. Check the workflow history to make sure that a workflow work item was assigned to the expected user.
3. Check the status of the pending email notification in **System administration > Periodic tasks > Email processing > Batch email sending status**.

If the email fails to send, make sure that the SMTP mail account can be opened.

4. Check for the email notification in the appropriate inbox.

## Troubleshoot email

There are a few standard steps that can help you troubleshoot the configuration of email settings.

1. Verify email settings:
  - a. Go to **System administration > Setup > Email > Email parameters**.
  - b. Verify that SMTP is enabled.
  - c. Verify the settings of the SMTP mail server.
  - d. Sign in to the SMTP account in a separate window to make sure that the account and password are correct.
  - e. Send a test email using **System administration > Setup > Email > Email parameters > Test email**.
2. Verify that the email batch process is running:
  - a. Go to **System administration > Periodic tasks > Email processing > Batch**.
  - b. Make sure that the **Batch processing** option is set to **Yes**.
  - c. Review the recurrence of the email process:
    - a. Select **No end date** to adjust all recurrences of the email batch process.
    - b. Adjust the count as you require.
3. To review the contents and status of batch emails, go to **System administration > Periodic tasks > Email processing > Batch email sending status**.
  - a. If you're using a release that is earlier than Platform update 28, personalize the form to add the email sender for easy review. To do this, right-click the grid header, select **Add columns**, select **Email**, and then click **Insert**. If the **Email** field isn't added into the grid, you can view the sender by selecting **Show message**, and then selecting the **Email** field.
  - b. Verify that emails are being sent from the correct account. If the account is incorrect, you need to adjust settings such as user options, system templates, or organization templates, as needed.
  - c. Verify that all email user accounts have been granted permission to **Send As** for the configured SMTP account (see step 4 for details).
4. In Platform update 32, an **Email history** page was added to allow administrators to review all sent emails, including any errors that might have prevented an email from being sent. The **Email history** page will show interactive as well as non-interactive/batch emails. For any emails that have an **Email status** of **Failed**, review the error message on the **Failure details** tab and determine if corrective

actions should be taken.

5. In the Microsoft 365 admin center, verify that all user mail accounts that will be used to send emails have **Send As** and **Send On Behalf Of** permissions for the configured SMTP account. For more information, see [Enable sending email from another user's mailbox in Microsoft 365](#).
6. Sign in to all user mailboxes to verify that they are valid and can be accessed using sign in.
7. Send a test email using **System administration > Setup > Email > Email parameters > Test email**.
8. If the SMTP settings were migrated from another environment, clear the password field and re-enter the password to ensure that the field encryption hasn't negatively affected the stored value.
9. If you continue to experience issues when email is sent via SMTP, enter the SMTP account information in a tool such as [SMTPer.net](#) to verify that the SMTP server and account are valid and working correctly.

## Troubleshoot the Exchange mail provider

The **Email parameters** page allows an administrator to select Exchange as an interactive email provider and as the Batch email provider. The Exchange mail provider will use the current user's Exchange Online account to send emails. When used as the Batch email provider, the batch account will be used. No additional configuration is needed. If troubleshooting is needed, ensure that the current user's account can be signed into and that emails can be sent from that account to the intended recipients.

### **Exchange mail provider not supported for external users**

Users that are external to the primary tenant will not have exchange accounts on that tenant, so the Exchange mail provider is not supported for external users.

## Other notes

The system communicates with Exchange or an SMTP server like a typical email client, so standard behavior and limits apply. For example, standard [Exchange Online receiving and sending limits](#) apply.

## Troubleshooting

### **Where do workflow email templates come from?**

The email templates will be sourced from either system email templates or organization email templates depending on whether the workflow is a system-level (not company specific) or organization-level (company specific) workflow.

## Additional resources

[Troubleshoot the Office integration](#)

[Office integration tutorial](#)

[Configure email functionality in Microsoft Dynamics AX \[AX 2012\]](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Date/time data and time zones

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides information about date and time fields, and time zones.

## Date and time fields

There are three types of date and time fields that correspond to different data types in the database:

- **Combined date/time fields** – These fields are the preferred method of entering date and time data. The **utcdatetime** data type stores time and date data in a single field in Coordinated Universal Time (UTC). UTC is the primary time standard by which the world regulates clocks and time. It is, within about 1 second, mean solar time at 0° longitude; it does not observe daylight saving time. Time zones around the world are expressed using positive or negative offsets from UTC. For most purposes, UTC is considered interchangeable with Greenwich Mean Time (GMT). The current version of UTC is defined by International Telecommunications Union Recommendation (ITU-R TF.460-6).
- **Date fields** – These fields are used to enter dates only. The **date** data type stores a day, month, and year. However, these values are not stored in UTC and cannot be associated with a time zone.
- **Time fields** – These fields are used to display the number of seconds that have elapsed since midnight on the current date. The **timeofday** data type stores an integer value. Time values are not stored in UTC.

## Time zones

To express UTC times in the local time, you must provide a time zone. The time zone controls the offset from UTC that is the equivalent of the local time. For example, the offset for Moscow is UTC+3. Your preferred time zone is first set according to the Windows locale of your computer, although it might have been changed by an administrator. Your preferred time zone is used only when displaying combined dates and times. To set the preferred time zone for a user, go to **Users** page. The page will show the list of users of the system. Select the user that you want to set the preferred time zone for, and click **User options**. On the **Language and region** tab, select the preferred time zone.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce for IT pros and developers

2/18/2021 • 3 minutes to read • [Edit Online](#)

This page lists the resources that are available for IT pros and developers who support Commerce.

## Online channel

- [Online channel extensibility](#)

## Components and architecture

- [Commerce components](#)
- [Modern POS \(MPOS\) architecture](#)
- [Commerce Scale Unit architecture](#)
- [Online store publishing architecture](#)
- [Retail channel performance PowerBI.com solution](#)

## Deployment

- [Configure, install, and activate Modern POS \(MPOS\)](#)
- [Manage accounts and devices from headquarters](#)
- [Point of sale \(POS\) device activation](#)
- [Install the POS Layout designer](#)
- [Configure and install Retail hardware station](#)
- [Configure and install Commerce Scale Unit \(self-hosted\)](#)
- [Retail component events for diagnostics and troubleshooting](#)
- [Access instances](#)

## Point of Sale (POS)

- [Modern POS configuration and installation](#)
- [Manage activation accounts and validate devices](#)
- [Point of sale \(POS\) device activation](#)
- [Security best practices for Cloud POS in shared environments](#)
- [Support for external gift cards](#)

## Customize POS

- [Screen layouts for the point of sale \(POS\)](#)
- [Install the POS layout designer](#)
- [Modern POS \(MPOS\) triggers and printing](#)
- [Block transactions by using triggers](#)
- [Add custom controls to POS views](#)
- [Point of sale \(POS\) payment extension](#)
- [Support for tipping in the POS payments SDK](#)
- [Run the point of sale \(POS\) samples](#)
- [Extend POS views to add custom columns and app bar buttons](#)

- [Add custom controls to Modern POS \(MPOS\) transaction pages](#)
- [Add custom columns to a point of sale \(POS\) transaction grid](#)
- [Call point of sale \(POS\) APIs or operations from POS extensions](#)
- [POS \(MPOS\) triggers and printing](#)
- [Add custom controls to non-screen designer-based POS views](#)
- [Development in cloud-hosted environments without admin access](#)
- [Test recorder and Regression suite automation tool for Cloud POS](#)
- [POS Cart view events and handlers](#)
- [Add custom buttons to the POS header bar](#)

## Integrating Commerce

- [Create and apply branding to the Retail Experience app](#)

## Commerce Scale Unit

- [Commerce Scale Unit customer and consumer APIs](#)
- [Commerce Scale Unit](#)
- [Configure and install Commerce Scale Unit \(self-hosted\)](#)

## Hardware station

- [Configure and install Retail hardware station](#)
- [Hardware Station extensibility](#)
- [Integrate POS with a new hardware device](#)

## Commerce Runtime

- [Commerce runtime \(CRT\) architecture and configuration](#)
- [Commerce runtime \(CRT\) extensibility and triggers](#)
- [Manage secrets for retail channels](#)
- [Log extension events to Application Insights](#)
- [Create async Commerce \(CRT\) APIs in your business logic](#)
- [Pre-extended columns in the channel database](#)
- [Extend Commerce Data Exchange - Real-time Service](#)

## Retail SDK

- [Retail software development kit \(SDK\) architecture](#)
- [Upcoming changes in the Retail SDK](#)
- [Development and ALM changes from version 10.0.10 to 10.0.13](#)
- [Create deployable packages](#)
- [Merge the build systems for Commerce and Finance](#)
- [Migrate the Retail SDK from Visual Studio 2015 to Visual Studio 2017](#)
- [Retail software development kit \(SDK\) samples](#)
- [Download samples and packages from GitHub and NuGet](#)
- [Channel database extensions](#)
- [Enable custom Commerce Data Exchange synchronization via extension](#)
- [Define and set order attributes](#)

- [Customer attributes](#)
- [Typescript and C# proxies for Retail point of sale \(POS\)](#)
- [Run the Retail point of sale \(POS\) samples](#)
- [Extend POS views to add custom columns and app bar buttons](#)
- [Add custom controls to Modern POS \(MPOS\) transaction pages](#)
- [Add custom columns to a point of sale \(POS\) transaction grid](#)
- [Call point of sale \(POS\) APIs or operations from POS extensions](#)
- [Modern POS \(MPOS\) triggers and printing](#)
- [Add custom controls to POS views that aren't screen layout designer-based](#)
- [Sign MPOS with a code signing certificate](#)
- [Show custom notifications in POS](#)
- [Set up Commerce SDK build pipeline](#)
- [Retail SDK FAQ](#)
- [Create Retail Server extension API \(Retail SDK version 10.0.11 and later\)](#)
- [Create new Retail Server extension API \(Retail SDK version 10.0.11 and earlier\)](#)
- [Consume Retail Server APIs in external applications](#)

## eCommerce

- [e-Commerce platform software development kit \(SDK\)](#)
- [Configure online stores](#)

## Payment connectors

- [Create an end-to-end payment integration for a payment terminal](#)
- [Implementing a payment connector and payment device \(white paper\)](#)
- [Create Windows installers for payment connectors](#)
- [Deploy payment connectors](#)
- [Create Commerce payment packaging for Finance and Operations deployment](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Online channel extensibility

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers online platform extensibility in Microsoft Dynamics 365 Commerce.

## Overview

The Dynamics 365 Commerce platform provides a rich online software development kit (SDK) for developer extensibility. It also provides a module library. The module library provides a set of ready-built modules, data actions, and themes that you can use for your site.

Online pages (for example, the home page, product details pages, and category pages) are made up of component modules (for example, header, carousel, and content block modules). The modules use data actions to fetch data (for example, product data, and ratings and reviews) and to render HTML to show a customer-facing page. Each module contains configuration fields that a page author or site administrator can set in the site builder tool. These fields include fields for layout options such as image placement in the module, fields for links to products or pages, and fields for images or strings that will be shown in the module. Themes contain Cascading Style Sheets (CSS) code that is used for styling. They also contain layout overrides for modules.

## Online SDK

The online SDK lets developers create and customize e-Commerce modules, data actions, and themes.

## Module library

The module library contains production-ready components that work with preconfigured authoring templates and pages. These components include modules, data actions, and themes. A developer can use the online SDK to customize each module and theme as required.

## Command-line interface tools

Command-line interface (CLI) tools are provided as part of the online SDK. These tools help you create new modules, data actions, and themes. There is also a CLI tool that you can use to package all the configurations for your site into a single configuration file. You can then upload this file to your production or test site by using Microsoft Dynamics Lifecycle Services (LCS).

## Additional resources

[Architectural overview](#)

[System requirements](#)

[Set up a development environment](#)

[Module library overview](#)

[e-Commerce components](#)

[CLI command reference](#)

[Package configurations and deploy them to an online environment](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Get started with e-commerce online extensibility development

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic provides an overview that will help you start to develop e-commerce customizations by using the Microsoft Dynamics 365 Commerce online software development kit (SDK).

You can use the Commerce online SDK to develop and debug e-commerce [modules](#), [data actions](#), and [themes](#). It can be installed and used in any Windows 10 environment, and doesn't require a deployed Commerce environment. However, it can also be useful to connect to a live cloud environment (for example, "Dev," "Test," "UAT," or "Prod"). In this way, you can do deeper debugging and test changes against a cloud environment without having to deploy or destabilize the environment.

The typical process when you extend the Commerce platform includes the following actions:

- Install the required development tools.
- Install the latest online SDK.
- Optional: [Set up an Azure DevOps build pipeline for code sharing and build management](#).
- Build and extend modules, data actions, and themes as required.
- Test modules, data actions, and themes by using mock data.
- Optional: Test modifications against a cloud environment.
- Build a Commerce configuration package of all changes.
- Deploy the configuration package to a cloud environment.

## Architectural overview

Before you start Commerce extensibility development, we recommend that you learn about the Commerce architecture. The [Dynamics 365 Commerce architecture overview](#) and [E-commerce architectural overview](#) topics are great places to start. Additionally, the [E-commerce components](#) topic provides an overview of the main extensible e-commerce components, such as modules, data actions, and themes.

## Set up a local online SDK development environment

### Development system requirements

Before you set up a development environment, make sure that your environment meets the minimum requirements that are specified in [System requirements for a Dynamics 365 Commerce online extensibility environment](#).

### Install development tools and the online SDK

Commerce development uses some free open-source tools, such as Node.js for the JavaScript runtime, Yarn for dependency management, and Visual Studio Code for source editing. For detailed information about how to install each of these tools, see [Set up a development environment](#). That topic also explains how to run a Node app to test and preview modules that are under development, and describes the [command-line interface \(CLI\) tools](#) that are used to create and clone a module.

## Develop against a cloud-hosted Commerce environment

The online SDK lets you develop and debug modules, data actions, and themes without having to use a deployed Commerce environment. There might be scenarios where you must debug your live e-commerce

website or test configuration changes against a live [cloud environment](#), such as a "Dev," "Test," "UAT," or "Prod" environment. This section explains how to configure your development environment against a cloud-hosted Commerce environment.

### Configure the SDK environment .env file

The online SDK includes an .env file in the root folder. This file has several environment variables that you can use to configure your development environment against a cloud environment. For detailed information about each environment variable, see [Configure a development \(.env\) file](#).

### Configure against a cloud environment

For detailed information about how to configure the .env file so that it points to a cloud environment, see [Debug against a tier 1 Commerce development environment](#). That topic also provides guidance that will help you set the correct URL for each variable in the .env file.

There are two main scenarios that you can set up. In the first, you point the .env file to a cloud Retail Server, and in the second, you point it to a cloud e-commerce site.

#### Configure against a cloud-hosted Retail Server

By configuring a development environment so that it points to a cloud-hosted Retail Server, you can debug modules and data actions against live data. After the server is configured, all data action calls go directly to it and pass real data back to your development environment. This setup gives you a great way to see how a module will be rendered. Otherwise, you must use mock data to render your modules. For more information, see [Test modules with mock data](#) and [Test data action with mocks](#).

The .env file contains a `MSDyn365Commerce_BASEURL` variable that must point to the URL of a cloud-based Retail Server. Additionally, you must point to a specific channel and channel operating unit number (OUN).

The following example shows an .env file that points to a specific Retail Server, channel, and OUN.

```
...
MSDyn365Commerce_BASEURL=https://e-comdevtestf1d01de665c744a7devret.cloud.retail.dynamics.com/
MSDyn365Commerce_CHANNELID=68719478279
MSDyn365Commerce_CATALOGID=0
MSDyn365Commerce_OUN=128
...
```

#### NOTE

The catalog ID variable, `MSDyn365Commerce_CATALOGID`, must always be set to 0 (zero), because Commerce doesn't support multiple catalogs.

#### Configure against a cloud-hosted e-commerce site

You can also configure a development environment so that it points to a cloud-hosted e-commerce site. In this way, e-commerce pages that are created in Commerce site builder can be rendered in the local environment under the local Node.js JavaScript server. This setup is useful because you can test changes to modules, data actions, and themes against live pages without affecting the live site. For example, you can test module view changes that you've made locally for any e-commerce page, without affecting the live e-commerce site or customer views of it.

Commerce site pages are stored in the Commerce content management system (CMS) as JavaScript Object Notation (JSON) files that can be saved and used as mock files. The JSON files contain the breakdown of modules and their configuration values. Those configuration values are used to render pages. In this scenario, the local development environment renders the local modules. Therefore, you can quickly test and iterate your changes.

For information about how to use page mocks to test modules, see [Test modules by using page mocks](#). That

topic includes information about how to create a page mock from a live page by using the ?  
`item=nodeserviceproxy:true` query string parameter.

The following example of a JSON file shows how to set up the `MSDyn365_HOST` variable so that it points to your Commerce environment.

```
MSDyn365_HOST=www.fabrikam.com
MSDyn365Commerce_BASEURL=https://e-comdevtestf1d01de665c744a7devret.cloud.retail.dynamics.com/
MSDyn365Commerce_CHANNELID=68719478279
MSDyn365Commerce_CATALOGID=0
MSDyn365Commerce_OUN=128
...
```

For more information about how to set up the `MSDyn365_HOST` variable so that it points to your Commerce environment, see [Debug against a tier 1 Commerce development environment](#).

## Change modules, data actions, and themes

After you've set up a Commerce development environment, you're ready to build custom [modules](#), [data actions](#), and [themes](#).

## Package creation and deployment

After you've completed all your changes, you can build a zip file package and upload it to [Microsoft Dynamics Lifecycle Services \(LCS\)](#). You can then see all your changes in the cloud-hosted environment that the package was deployed to. For more information and instructions, see [Package configurations and deploy them to an online environment](#).

## Additional resources

[System requirements for a Dynamics 365 Commerce online extensibility development environment](#)

[Set up a development environment](#)

[Configure a development environment \(.env\) file](#)

[Configure an e-commerce development environment against a Commerce cloud environment](#)

[Set up Azure DevOps code sharing and create a build pipeline](#)

[Dynamics 365 Commerce architecture overview](#)

[E-commerce architectural overview](#)

[E-commerce components](#)

[Modules overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# System requirements for a Dynamics 365 Commerce online extensibility development environment

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic lists the system requirements for setting up a Microsoft Dynamics 365 Commerce online extensibility development environment. Before you install the online software development kit (SDK) and tools, you should verify that the system that you're working with meets or exceeds the minimum hardware and software requirements.

## Operating systems

The online extensibility development environment requires Windows 10 (the latest publicly available version).

## Memory and hard drive requirements

We recommend that you have 8 gigabytes (GB) or more of RAM and at least 20 GB of free hard drive space to run the online extensibility environment.

## Supported web browsers

The online extensibility development environment can preview customizations in any of the following web browsers:

- Google Chrome (up to the current version minus 1)
- Mozilla Firefox (up to the current version minus 1)
- Microsoft Edge (up to the current version minus 1)

## Additional resources

[Get started with e-commerce online extensibility development](#)

[Set up a development environment](#)

[Configure a development environment \(.env\) file](#)

[Configure an e-commerce development environment against a Commerce cloud environment](#)

[Set up Azure DevOps code sharing and create a build pipeline](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up a development environment

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to set up a development environment for Microsoft Dynamics 365 Commerce.

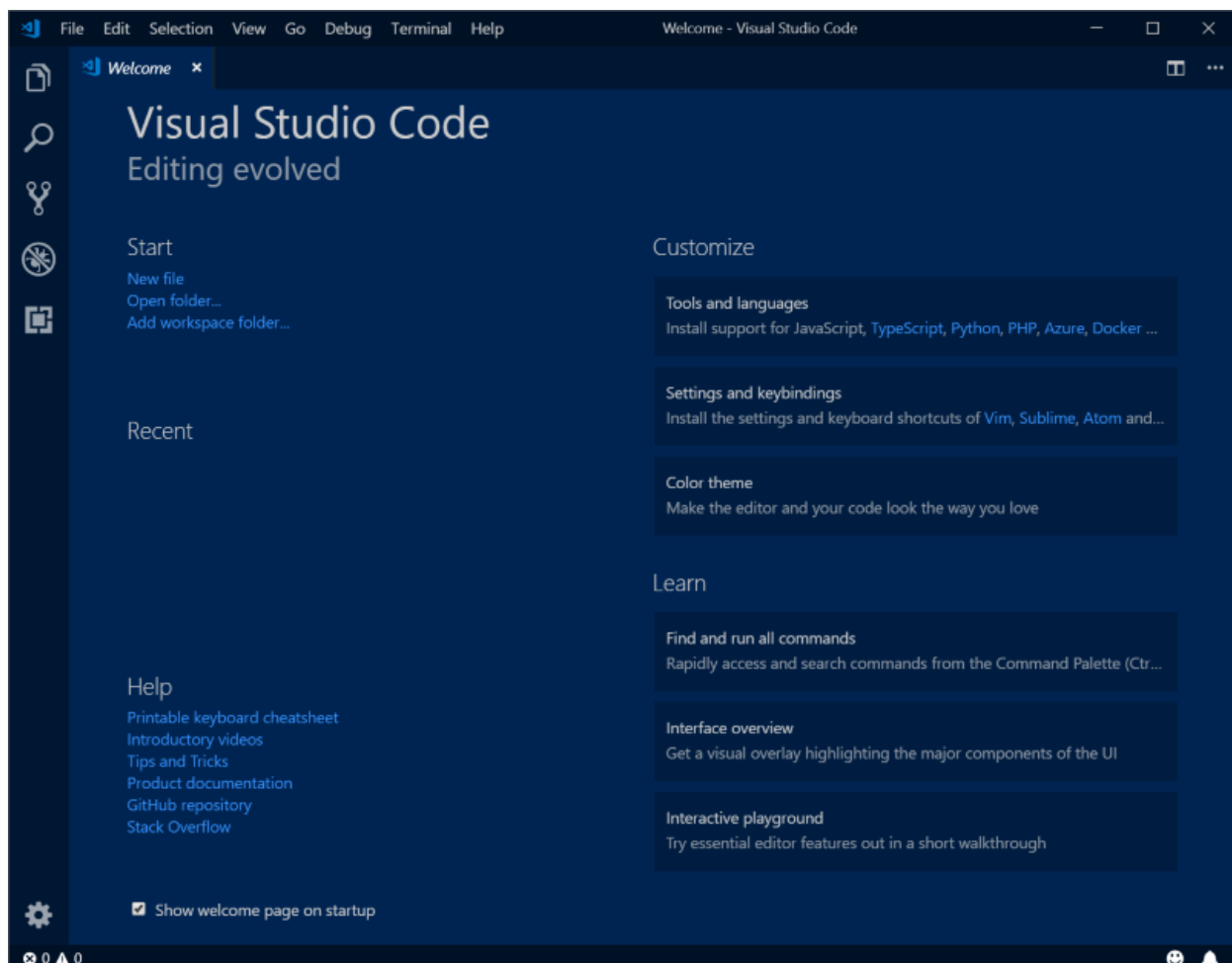
## Overview

To set up a development environment for Dynamics 365 Commerce online extensibility development, you must install three free tools: Microsoft Visual Studio Code, Node.js, and Yarn. You must also install the Dynamics 365 Commerce Online Software Development Kit (SDK). You can install these tools in any order.

## Install Visual Studio Code

We recommend that you use a source code editor such as Visual Studio Code. Visual Studio Code is a lightweight source code editor that runs on your Microsoft Windows desktop. It has built-in support for JavaScript, TypeScript, and Node.js.

Go to the [Visual Studio Code site](#), and download and install the latest build. After installation is completed, Visual Studio Code is automatically opened and should resemble the following screenshot.



## Install Node.js

Node.js is a JavaScript runtime that is built on [Chrome's V8 JavaScript Engine](#).

Version 12.x is the current supported version. You can find the installer on the [Node.js website](#).

If you rely on other versions of Node.js for other projects, we recommend that you use [Node Version Manager \(nvm\)](#) to help guarantee that each version runs in its own isolated environment.

## Install Yarn

Yarn is a dependency management tool that helps guarantee that you have all the latest packages that you require for e-Commerce extensibility.

Currently, version 1.x is the only supported version. You can find the installer on the [Yarn website](#). As newer versions become supported, this document will be updated.

## Install the Online SDK and module library

The Online SDK provides everything that you require to extend your online channel with new modules, data actions, and themes.

The SDK configuration package is available through the [Msdyn365.Commerce.Online GitHub repository \(repo\)](#). Download or clone the repo to a local folder on your development computer. To clone the repo, use the following command. (This command will work only if you have [Git tools](#) installed.)

```
git clone https://github.com/microsoft/Msdyn365.Commerce.Online.git
```

### NOTE

The whole SDK and module library won't be downloaded and installed until you run the **yarn** command. For more information, see the [Download SDK dependencies](#) section later in this topic.

If you cloned the repo, you can remove the .git folder (the hidden directory under the root). You will use Yarn to pull down updated dependencies.

We recommend that you use a source code repository to manage your configuration changes. Many options are available, such as [Git](#).

## Download SDK dependencies

To download the SDK dependency packages, follow these steps.

1. At a command prompt, go to the root folder of the e-Commerce SDK (c:\repos\Msdyn365.Commerce.Online in the following example).
2. To get all the latest dependency packages that are required, run the **yarn** command.

### IMPORTANT

This step should be done after you've completed any update to the packages.json file.

```
c:\repos\Msdyn365.Commerce.Online>yarn
```

This command can take several minutes to run.

## Run your Node app

To run your Node app, follow these steps.

1. Run the **yarn start** command to open the Node app.

```
c:\repos\Msdyn365.Commerce.Online>yarn start
```

This command can take up to a minute to run. When it's completed, you will see output that indicates that the server has been started. The output also shows the allocated port number (4000 by default, but you can change the value in the `.env` file).

2. To test that your Node app is running correctly, open the following URLs in a web browser:

- `https://localhost:4000/version`
- `https://localhost:4000/_sdk/allmodules`

3. To close the Node app, at the command prompt, press **Ctrl+C** two times.

## Create a new module

To add a new module, run the **yarn msdyn365 add-module MODULE\_NAME** command. For example, the following command creates a module that is named **product-feature**.

```
c:\repos\Msdyn365.Commerce.Online>yarn msdyn365 add-module product-feature
```

This command can take several seconds to run. It adds a new module under `\src\modules\product-feature`.

## Clone an existing module library module

Several of the available module library modules can be cloned. These modules include the carousel, content-block, and header modules. A cloned module is a copy of the module and has a new name. Unlike the module library modules, cloned modules don't get regular service updates. Instead of cloning a module to make layout changes, you might want to extend the views on the module.

For example, to modify the content-block module, run the **yarn msdyn365 clone MODULE\_LIBRARY\_MODULE\_NAME NEW\_MODULE\_NAME** command to pull down the source code. Here is an example.

```
c:\repos\Msdyn365.Commerce.Online>yarn msdyn365 clone content-block super-content-block
```

You can find the new module under `\src\modules\super-content-block`.

### NOTE

After a module is cloned, you might have to fix up references in the code. You can run **yarn start** to highlight any errors that must be fixed.

## Preview modules

To preview a specific module (for example, `product-feature`) in a local web browser, follow these steps.

1. At a command prompt, open your Node app by running the **yarn start** command from the root of your SDK.

```
c:\repos\Msdyn365.Commerce.Online>yarn start
```

2. In a web browser, open the following URLs. Notice the module name in the "type=MODULE\_NAME" query string parameter.

- `https://localhost:4000/modules?type=product-feature`
- `https://localhost:4000/modules?type=content-block`
- `https://localhost:4000/modules?type=super-content-block`

## Adding an SSL certificate

The Dynamics 365 online SDK installs a self-signed SSL certificate for developing and testing on a local environment which work against localhost. You can find these files inside the `.ssl` folder under the root SDK folder. Note: `yarn start` must be run at least once for these files to be generated.

To install a new certificate on a developer environment, replace the public key (`cert.pem`) and private key (`key.pem`) files with your own.

## Additional resources

[Get started with e-commerce online extensibility development](#)

[System requirements for a Dynamics 365 Commerce online extensibility development environment](#)

[Configure a development environment \(.env\) file](#)

[Configure an e-commerce development environment against a Commerce cloud environment](#)

[Set up Azure DevOps code sharing and create a build pipeline](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Configure a development environment (.env) file

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to configure the .env file that is used for the development environment in Microsoft Dynamics 365 Commerce.

## Overview

The .env file that is provided as part of the Dynamics 365 Commerce online software development kit (SDK) is a simple configuration text file. It defines a set of variables that is used by a Node app that runs in the development environment.

## Default .env file

The default .env file that is provided with the online SDK should resemble the following example.

```

### Environment File #####
# This is a simple configuration
# Online documentation for this file is available at
# https://docs.microsoft.com/en-us/dynamics365/commerce/e-commerce-extensibility/configure-env-file
#####

# MSDyn365_APP_TYPE variable is required. It can be set only to the value partner.

MSDyn365_APP_TYPE=partner

# PORT variable is required. It defines the port number that is used to preview your Node
# application when the Node server is started by using the yarn start command. The default value is 4000

PORT=4000

# MSDyn365_HOST variable defines the domain name of your customer-facing e-Commerce site.
# When this variable is set, if the URL https://localhost:4000/ is opened in a development environment,
# your e-Commerce site will be rendered locally. If your site is protected through Azure Active Directory
# (Azure AD) credentials, a prompt for a user name and password will appear.

MSDyn365_HOST=

# MSDyn365Commerce_BASEURL variable defines the URL of the Microsoft Dynamics 365 Retail Server.
# When this variable is set, local development and testing can be done against Dynamics 365 Retail Server
# application programming interfaces (APIs). If you set this variable, you must also set the
# MSDyn365Commerce_CHANNELID, MSDyn365Commerce_OUN, and MSDyn365Commerce_CATALOGID variables.
# MSDyn365Commerce_OUN variable defines the operating unit number for the channel.
# MSDyn365Commerce_CATALOGID variable defines the catalog ID for the online store that you're connecting to.
# Currently, only the value 0 (zero) is supported.
# MSDyn365Commerce_CHANNELID variable defines the online channel that you're connecting to.

MSDyn365Commerce_BASEURL=
MSDyn365Commerce_CHANNELID=
MSDyn365Commerce_CATALOGID=
MSDyn365Commerce_OUN=

# MSDyn365Commerce_BASEIMAGEURL variable defines the URL for a website's image assets.
# The URL follows a pattern and must be manually generated. For more information,
# see online product documentation.

MSDyn365Commerce_BASEIMAGEURL=

```

Two of the variables in the .env file, **MSDyn365\_APP\_TYPE** and **PORT**, are required and have preset values. All the other variables are optional. The optional variables enable your development environment to get data from live environments. This data is then rendered in the local development Node server.

The following sections describe the variables in the .env file.

## MSDyn365\_APP\_TYPE

The **MSDyn365\_APP\_TYPE** variable is required. It can be set only to the value **partner**.

## PORT

The **PORT** variable is required. It defines the port number that is used to preview your Node application when the Node server is started by using the **yarn start** command. The default value is **4000**.

The following example shows the syntax for this variable.

```
PORT=4000
```

Here is an example of a development environment URL that includes the port number:

```
https://localhost:4000/version
```

## MSDyn365\_HOST

The `MSDyn365_HOST` variable defines the domain name of your customer-facing e-Commerce site. When this variable is set, if the URL `https://localhost:4000/` is opened in a development environment, your e-Commerce site will be rendered locally. If your site is protected through Azure Active Directory (Azure AD) credentials, a prompt for a user name and password will appear.

The following example shows the syntax for this variable.

```
MSDyn365_HOST=demo.fabrikam.com
```

## MSDyn365Commerce\_BASEURL

The `MSDyn365Commerce_BASEURL` variable defines the URL of the Microsoft Dynamics 365 Commerce Scale Unit. When this variable is set, local development and testing can be done against application programming interfaces (APIs). If you set this variable, you must also set the `MSDyn365Commerce_CHANNELID`, `MSDyn365Commerce_OUN`, and `MSDyn365Commerce_CATALOGID` variables.

The following example shows the syntax for this variable.

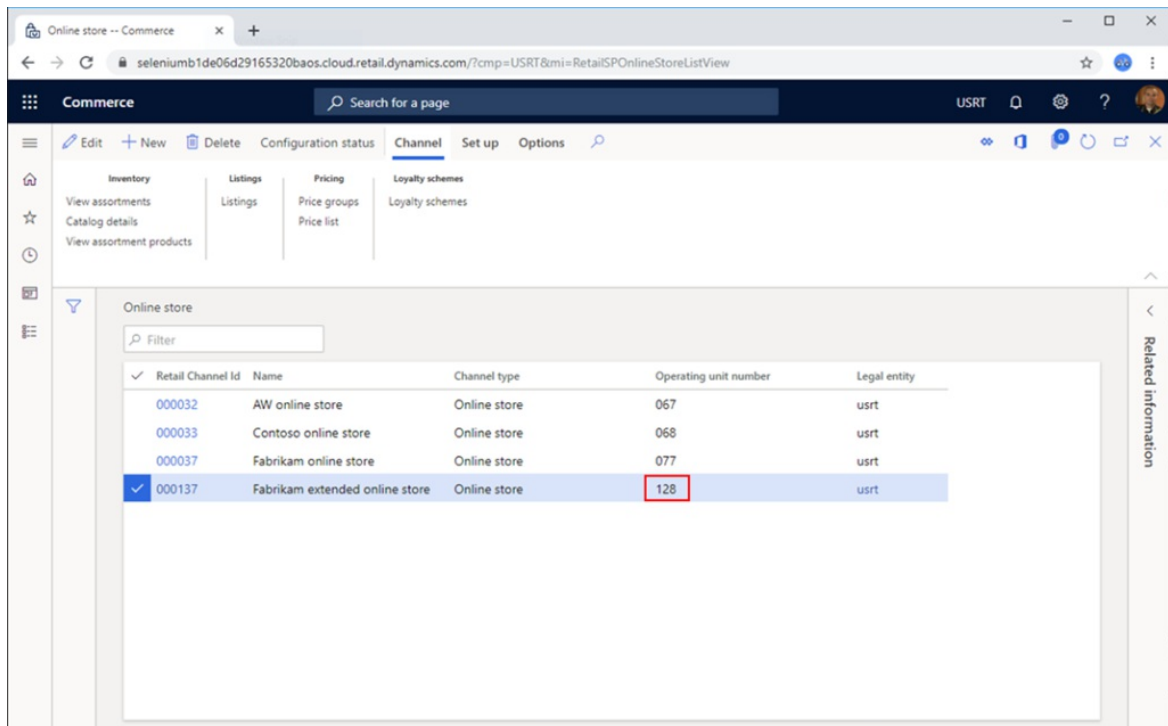
```
MSDyn365Commerce_BASEURL=https://fabrikamb1de06d29165320bret.cloud.retail.dynamics.com
```

## MSDyn365Commerce\_OUN

The `MSDyn365Commerce_OUN` variable defines the operating unit number for the channel.

To find the channel operating unit number, follow these steps.

1. Go to the Commerce website.
2. In the search field at the top of the page, enter **Online channels**, and then select the channel to use. You should see an **Operating unit number** column, as shown in the following illustration.



The following example shows the syntax for this variable.

```
MSDyn365Commerce_OUN=128
```

## MSDyn365Commerce\_CATALOGID

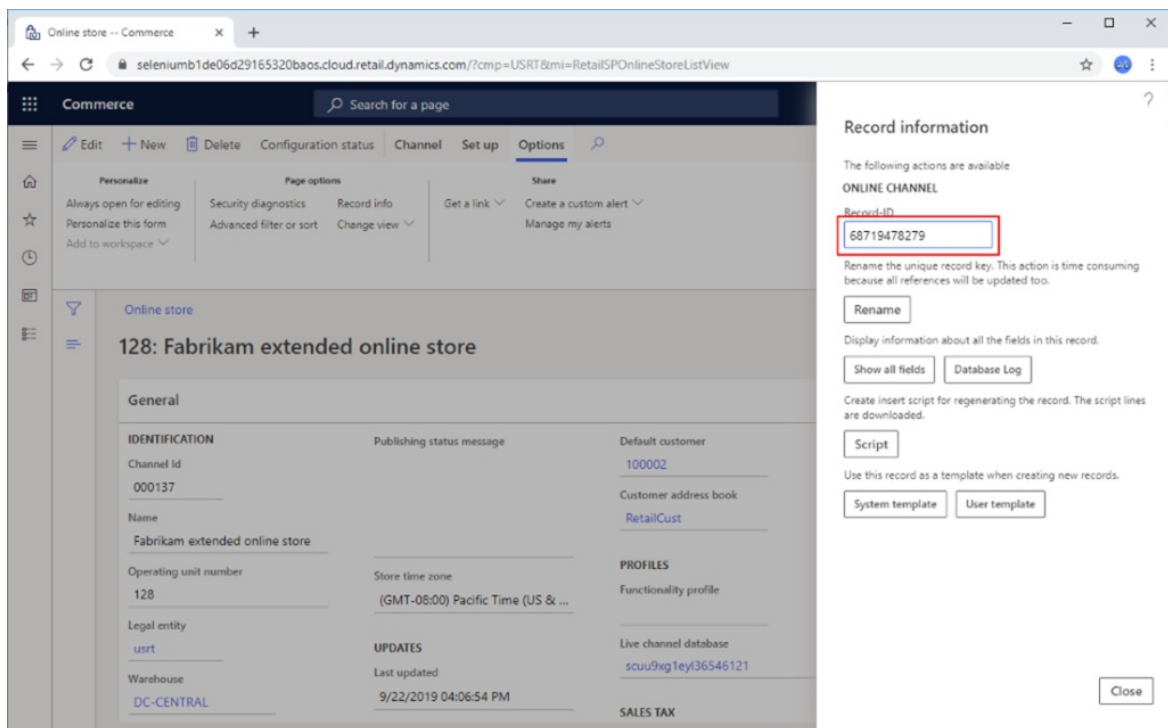
The `MSDyn365Commerce_CATALOGID` variable defines the catalog ID for the online store that you're connecting to. Currently, only the value 0 (zero) is supported.

## MSDyn365Commerce\_CHANNELID

The `MSDyn365Commerce_CHANNELID` variable defines the online channel that you're connecting to.

To find the channel ID, follow these steps.

1. Go to the Dynamics 365 Commerce website.
2. In the search field at the top of the page, enter **Online channels**, and then select the channel to use.
3. On the Action Pane, on the **Options** tab, in the **Page options** group, select **Record Info**.
4. In the **Record information** dialog box, the value of the **Record-ID** field is the channel ID. Copy this value.



The following example shows the syntax for this variable.

```
MSDyn365Commerce_CHANNELID=68719478279
```

## MSDyn365Commerce\_BASEIMAGEURL

The `MSDyn365Commerce_BASEIMAGEURL` variable defines the URL for a website's image assets. The URL follows a pattern and must be manually generated. Follow one of these steps:

- For evaluations sites, the URL has the following format:

```
https://images-us-sb.cms.commerce.dynamics.com/cms/api/{CMS_TENANT_ID}/imageFileData/search?fileName=/
```

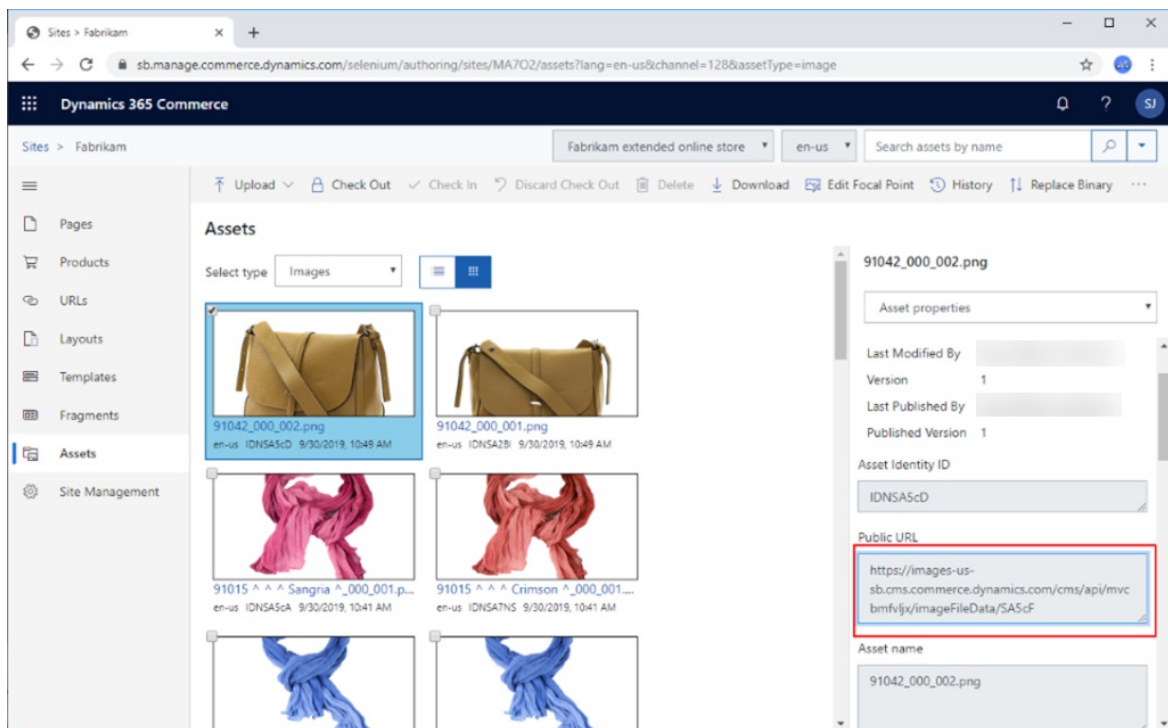
- For production sites, the URL has the following format:

```
https://img-prod-cms-mr-microsoft-com.akamaized.net/cms/api/{CMS_TENANT_ID}/imageFileData/search?fileName=/
```

In both URLs, you must replace `{CMS_TENANT_ID}` with the content management system (CMS) tenant ID that is assigned to your site.

To get the CMS tenant ID in Dynamics 365 Commerce, follow these steps.

1. Under **Sites**, select your site.
2. In the navigation pane on the left, select **Assets**.
3. In the **Select type** field, select **Images**.
4. Select the first image.
5. In the property pane on the right, find the URL in the **Public URL** field. Your CMS tenant ID is the string between `/cms/api/` and `/imageFileData`. For example, in the URL `../cms/api/fabrikam/imageFileData/..`, the CMS tenant ID is `fabrikam`.



After you've finished changing the .env file, restart the Node server by using the **yarn start** command.

## Additional resources

[Get started with e-commerce online extensibility development](#)

[System requirements for a Dynamics 365 Commerce online extensibility development environment](#)

[Set up a development environment](#)

[Configure an e-commerce development environment against a Commerce cloud environment](#)

[Set up Azure DevOps code sharing and create a build pipeline](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure an e-commerce development environment against a Commerce cloud environment

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to set up an e-commerce online development environment to debug against a Microsoft Dynamics 365 Commerce cloud development environment.

## Overview

An e-commerce development environment can be configured to debug your live e-commerce website or to test e-commerce configuration changes against various Commerce cloud environments, such as "Dev," "Test," "UAT," or "Prod" environments. This environment is useful for testing and debugging e-commerce modules and data actions against Retail Server extensions. After it's configured, modules and data actions that use Retail Server APIs will directly call the Retail Server in the Commerce cloud environment. Otherwise, mock data will be required.

## Install the Commerce online SDK

To get started, you must install the Dynamics 365 Commerce online software development kit (SDK). The online SDK can be installed in any Windows 10 environment. It can even be installed directly on a Commerce development virtual machine (VM). For setup instructions, see [Setup a development environment](#).

## Debug against a Retail Server in a Commerce cloud environment

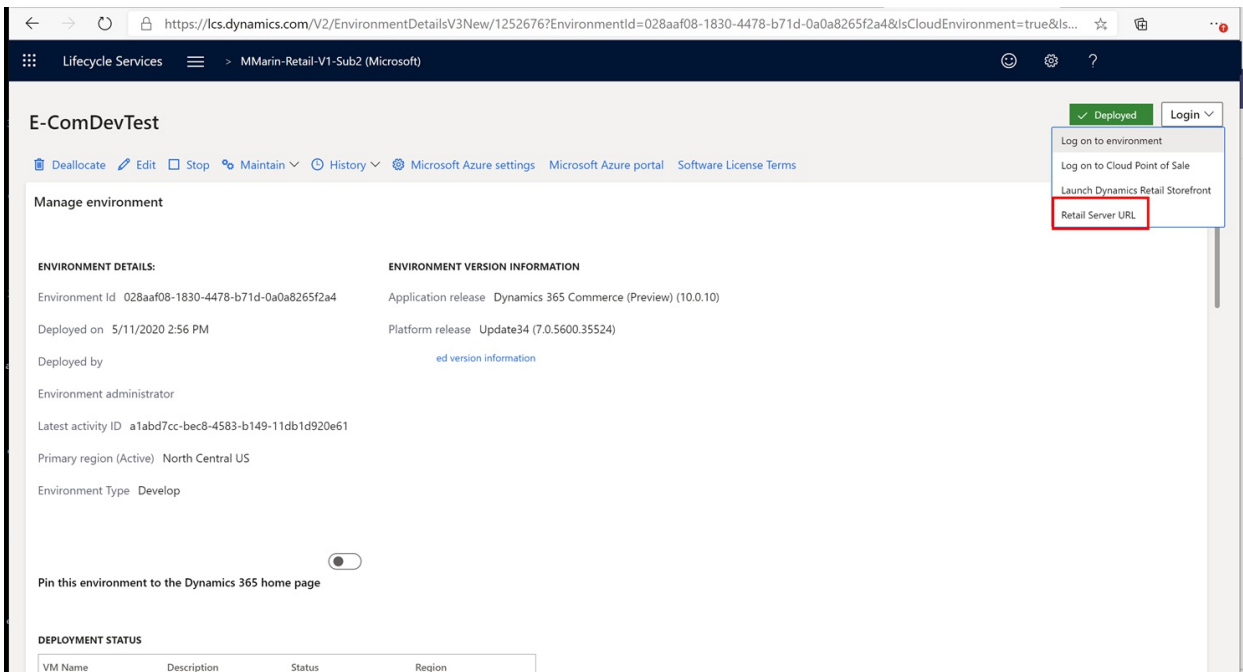
The online SDK uses Node.js as the JavaScript runtime to render modules and e-commerce pages in a development environment. To configure a local development environment so that it points to a Commerce cloud environment, set the `MSDyn365Commerce_BASEURL` variable in the `.env` file to the URL of the Retail Server in the Commerce cloud environment. For information about how to set up the `.env` file, see [Configure a development environment \(.env\) file](#). You must also specify the channel ID, `MSDyn365Commerce_CHANNELID`, and the channel operating unit number (OUN), `MSDyn365Commerce_OUN`.

### NOTE

Catalogs aren't supported in e-commerce. Therefore, the `MSDyn365Commerce_CATALOGID` variable will always be set to 0 (zero).

### Retrieve the URL of a Retail Server in a Commerce development environment

To get the Retail Server URL if you're debugging against a Commerce development environment, go to [Microsoft Lifecycle Services \(LCS\)](#), and select the project and the environment. Then, in the upper-right corner of the page, select **Login > Retail Server URL**, as shown in the following illustration.



A new tab should be opened, and the URL should resemble the following example:

```
https://e-comdevtestf1d01de665c744a7devret.cloud.retail.dynamics.com/Commerce
```

Copy this URL, except the last part (**Commerce**), into the `.env` file as the value of the `MSDyn365Commerce_BASEURL` variable. The `MSDyn365Commerce_CHANNELID` and `MSDyn365Commerce_OUN` variables should be set based on the desired online channel in the environment. For information about how to get these values, see [Configure a development environment \(.env\) file](#).

The following example shows configured variables that use the previously mentioned example URL.

```
MSDyn365Commerce_BASEURL=https://e-comdevtestf1d01de665c744a7devret.cloud.retail.dynamics.com/  
MSDyn365Commerce_CHANNELID=68719478279  
MSDyn365Commerce_CATALOGID=0  
MSDyn365Commerce_OUN=128  
...
```

To ensure that the server picks up these new values after the `.env` file is saved, you should restart the Node.js server by using the `yarn start` command. As you build modules and debug data actions, calls will now be made directly to the Retail Server that is specified in the `.env` file.

## Debug against a production e-commerce site

The Commerce online SDK lets you point your development environment to a production e-commerce site to retrieve page definitions that can be rendered in the local Node.js environment. Therefore, you can see how your local e-commerce changes (modules, data actions, and themes) will be rendered before you upload the configuration package to a live environment. You can then debug, make further changes, and see how those changes will look on a production page.

On a production site, Commerce site builder is used to build e-commerce pages that are stored as JavaScript Object Notation (JSON) files in the Commerce content management system. When you configure an e-commerce development environment's `MSDyn365_HOST` variable in the `.env` file so that it points to a production e-commerce site, the JSON file will be retrieved, and the local Node.js server will use the local online SDK and customizations in that local environment to do the rendering. Therefore, you can test e-commerce changes on your live e-commerce site pages without deploying and potentially destabilizing your production environment.



To support this scenario, configure the **MSDyn365\_HOST** variable in the `.env` file to point to your e-commerce domain name. When this step is complete, you can run the "yarn start" command and navigate to `https://localhost:4000` to view your online website rendered on the local Node.js server. When this happens, the live page will be pulled from the Dynamics 365 Commerce content management system. All data action Retail Server calls will be routed to the Tier 1 environment, as specified in the `.env` file.

The following example `.env` file shows the **MSDyn365\_HOST** variable set to `www.fabrikam.com`. Note that this does not include the `https://` part of the URL.

```
MSDyn365_HOST=www.fabrikam.com
MSDyn365Commerce_BASEURL=https://e-comdevtestf1d01de665c744a7devret.cloud.retail.dynamics.com/
MSDyn365Commerce_CHANNELID=68719478279
MSDyn365Commerce_CATALOGID=0
MSDyn365Commerce_OUN=128
...
```

#### NOTE

If you have multiple e-commerce sites configured for a single domain name, do not include the site name in the **MSDyn365\_HOST** name provided in the `.env` file. Instead, use the site names when navigating the development environment in the local browser. For example, if you have two sites, `www.fabrikam.com/site1` and `www.fabrikam.com/site2`, configure the `.env` file as shown in the example above (`www.fabrikam.com`), and navigate to `https://localhost:4000/site1` or `https://localhost:4000/site2` respectively in the development environment.

### Debug a product details page

To open up a specific product details page (PDP), you can use the product ID to manually construct a URL using the pattern `https://localhost:4000/SITE_NAME/PRODUCT_NUMBER.p`. For example, `https://localhost:4000/site1/68719498121.p`, where "site1" is the site name and "68719498121" is the product ID. To obtain the product ID, you can navigate directly to a product on the live web site and copy the product ID from the URL, or in Commerce headquarters you can navigate to a released product and select the **Record info** link under the **Options** tab, and then copy the **Record-ID**.

## Troubleshooting

### CORS errors

You may get CORS (cross origin) errors when calling Retail Server APIs from your browser. These errors may surface in the browser network trace as **(failed) net::ERR\_FAILED**. To fix these errors, change the **AllowedOrigins** setting in the Retail service `web.config` to allow the call to go through, as shown below.

```
...
<add key="AllowedOrigins" value="*" />
<!-- <add key="AllowedOrigins"
value="https://usnconeboxax1pos.cloud.onebox.dynamics.com;https://usnconeboxax1ecom.cloud.onebox.dynamics.com" /> -->
...
```

### Mixed content errors

When the Retail Server is configured with HTTP instead of HTTPS, you may receive "Mixed Content" errors when rendering e-commerce content. Ensure the Retail Server is configured with an HTTPS end point to avoid this type of error.

### Retail calls are failing 404 error

404 errors may result if the channel ID and OUN are incorrect. To avoid those errors, ensure that the channel ID

and OUN are correct in the `.env` file. See the [Configure a development environment \(.env\) file](#) topic for details.

## Additional resources

[Get started with e-commerce online extensibility development](#)

[System requirements for a Dynamics 365 Commerce online extensibility development environment](#)

[Set up a development environment](#)

[Configure a development environment \(.env\) file](#)

[Set up Azure DevOps code sharing and create a build pipeline](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up Azure DevOps code sharing and create a build pipeline

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to set up code sharing with Microsoft Azure DevOps and create a build pipeline for your Dynamics 365 Commerce online extensibility code.

## Overview

By taking advantage of the capabilities of [Azure DevOps](#), you can help your team plan work, collaborate on code development, and automate the building of deployment packages for Dynamics 365 Commerce e-Commerce.

This topic guides you through the steps that are required to complete the following tasks:

- Create a GitHub repository (repo) for the Commerce online software development kit (SDK).
- Create and configure a build pipeline to generate a Commerce online deployable package.

## Create an Azure DevOps GitHub repo

You can create an Azure DevOps GitHub repo project from a new or existing Azure DevOps service subscription. For more information, see [Azure DevOps Service](#). To get started with a free trial account, see the [Get started with Azure DevOps](#) quickstart guide.

To create an Azure DevOps GitHub repo, follow these steps.

1. After the Azure DevOps service is set up for your organization, create a new Azure DevOps project.

# Create a project to get started

Project name \*

Description

## Visibility



### Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.



### Enterprise

[Members of your enterprise](#) can view the project.



### Private

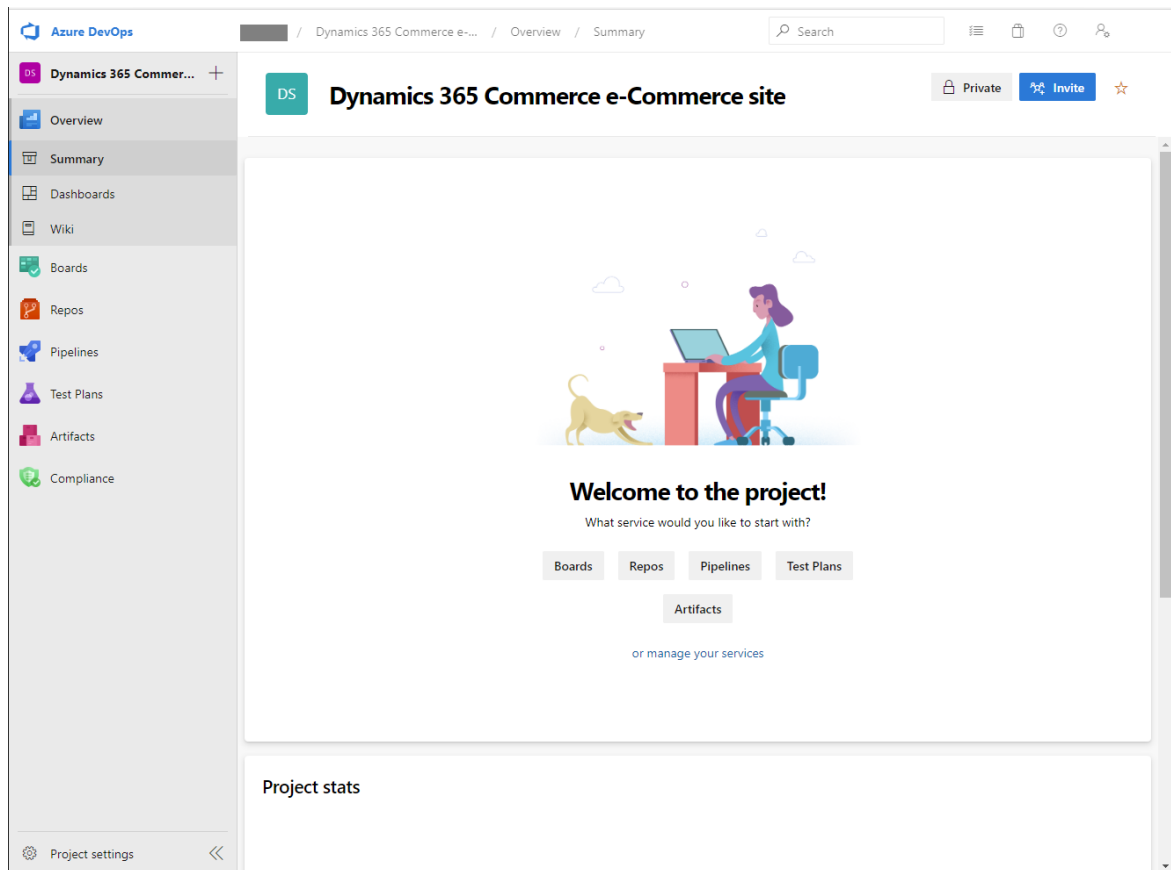
Only people you give access to will be able to view this project.



∨ Advanced

+ Create project

2. Enter a project name and a description. Select **Private** or **Enterprise** visibility, so that the project is accessible to your organization and developers.



3. For this example, you will use Git to clone the SDK code. Git is a free, open-source distributed version control system. Go to <https://git-scm.com/downloads> to download and install the latest build. You should be able to accept all the default installation values.
4. Install Visual Studio Code. Visual Studio Code is a lightweight source code editor that runs on your desktop, and is available for Windows, macOS, and Linux. It includes built-in support for JavaScript, TypeScript, and Node.js. Go to <https://code.visualstudio.com> to download the latest stable build. Then open the installer, and accept the user license agreement. You should be able to accept all the default installation values.
5. Clone the Commerce online SDK. The SDK provides everything that you need to extend your e-Commerce site. For example, you can use it to create new modules, data actions, and themes. The SDK configuration package is available through the following GitHub repo:  
<https://github.com/microsoft/Msdyn365.Commerce.Online>.

There are two ways to get the SDK configuration packages to your development computer. You can either download the packages directly from the GitHub repo, or you can clone the repo.

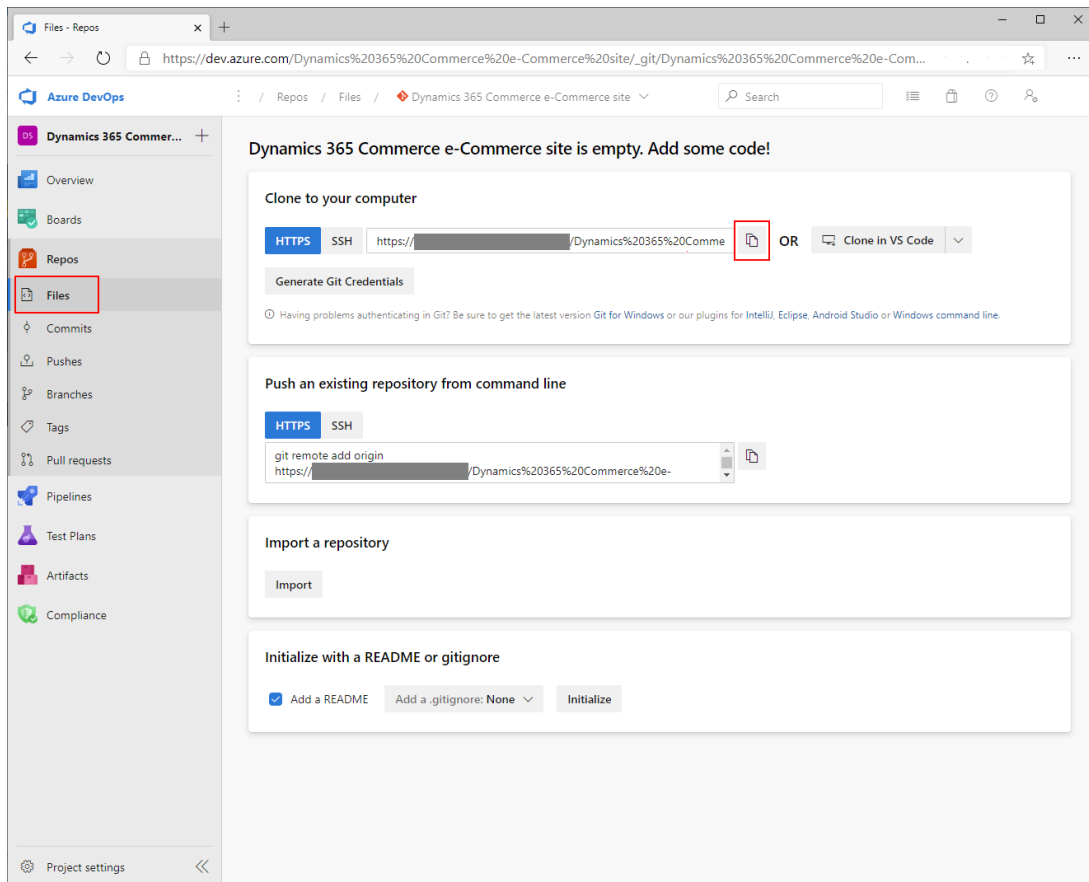
To clone the repo, follow these steps:

- a. Open a Command Prompt window as an admin, and create a directory to hold your e-Commerce site code (for example, `c:\repos`).
- b. From the new directory, enter `git clone <YOUR_GIT_REPO>`, where `<YOUR_GIT_REPO>` is your GitHub repo. Because you're pulling from the GitHub repo only one time, you can remove the `.git` folder, which is a hidden directory under the root.

Here is an example.

```
md c:\repos
cd c:\repos
git clone https://github.com/microsoft/Msdyn365.Commerce.Online.git
cd Msdyn365.Commerce.Online
```

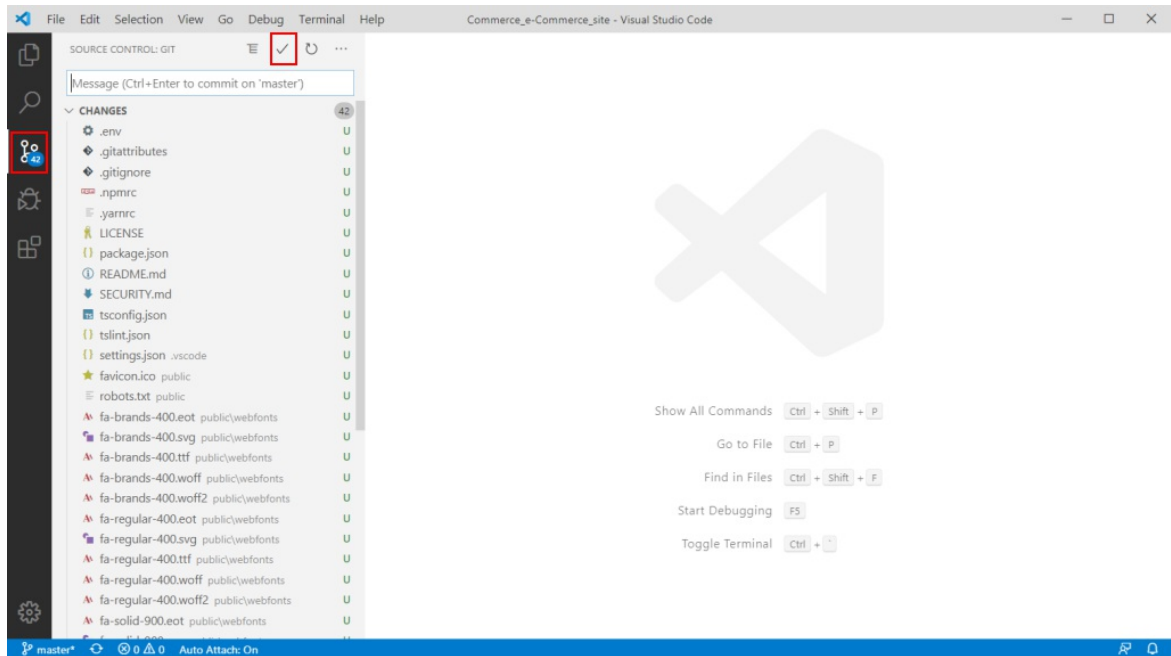
6. Clone the Azure DevOps GitHub project repo:
  - a. In the left navigation pane, under **Repos**, select **Files**.
  - b. Select the **Copy** button to copy the URL.



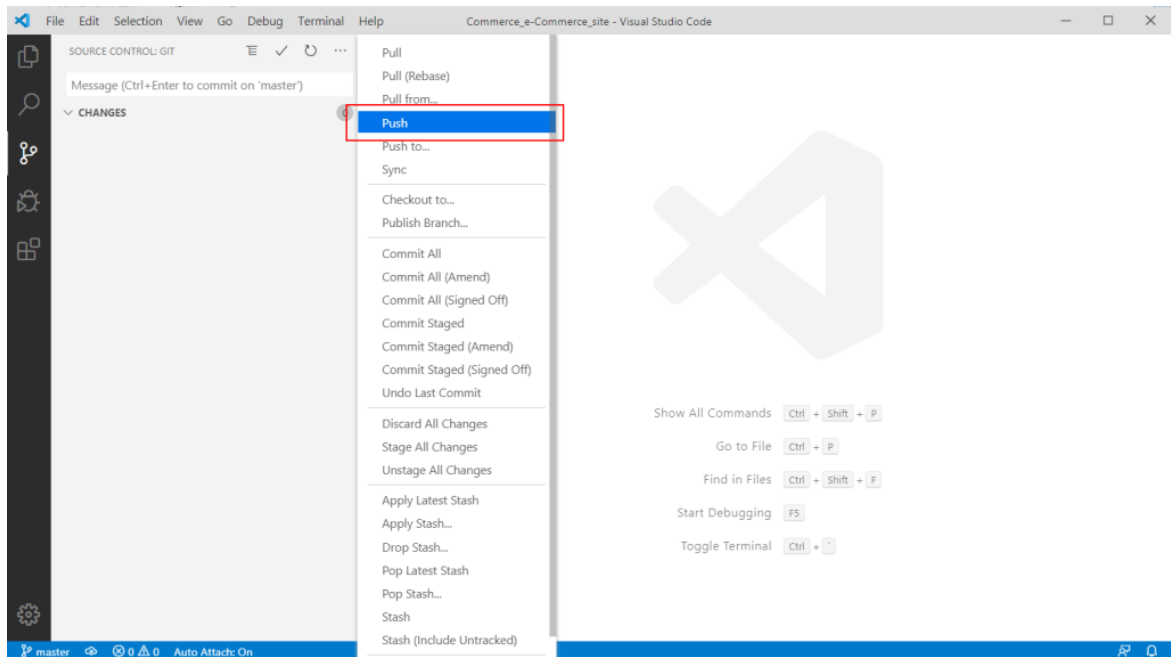
- c. Open a Command Prompt window as an admin, and create a directory to hold your e-Commerce site code (for example, `c:\repos`).
    - d. From the new directory, enter `git clone <AZURE_DEVOPS_GIT_REPO>`, where `<AZURE_DEVOPS_GIT_REPO>` is the Azure DevOps GitHub project repo. A new, empty folder that has the name of the Azure DevOps project will be created.

```
cd c:\repos
git clone https://xxxxxx.dev.azure.com/<DevOpsProjectName>/_git/<DevOpsProjectName>
```

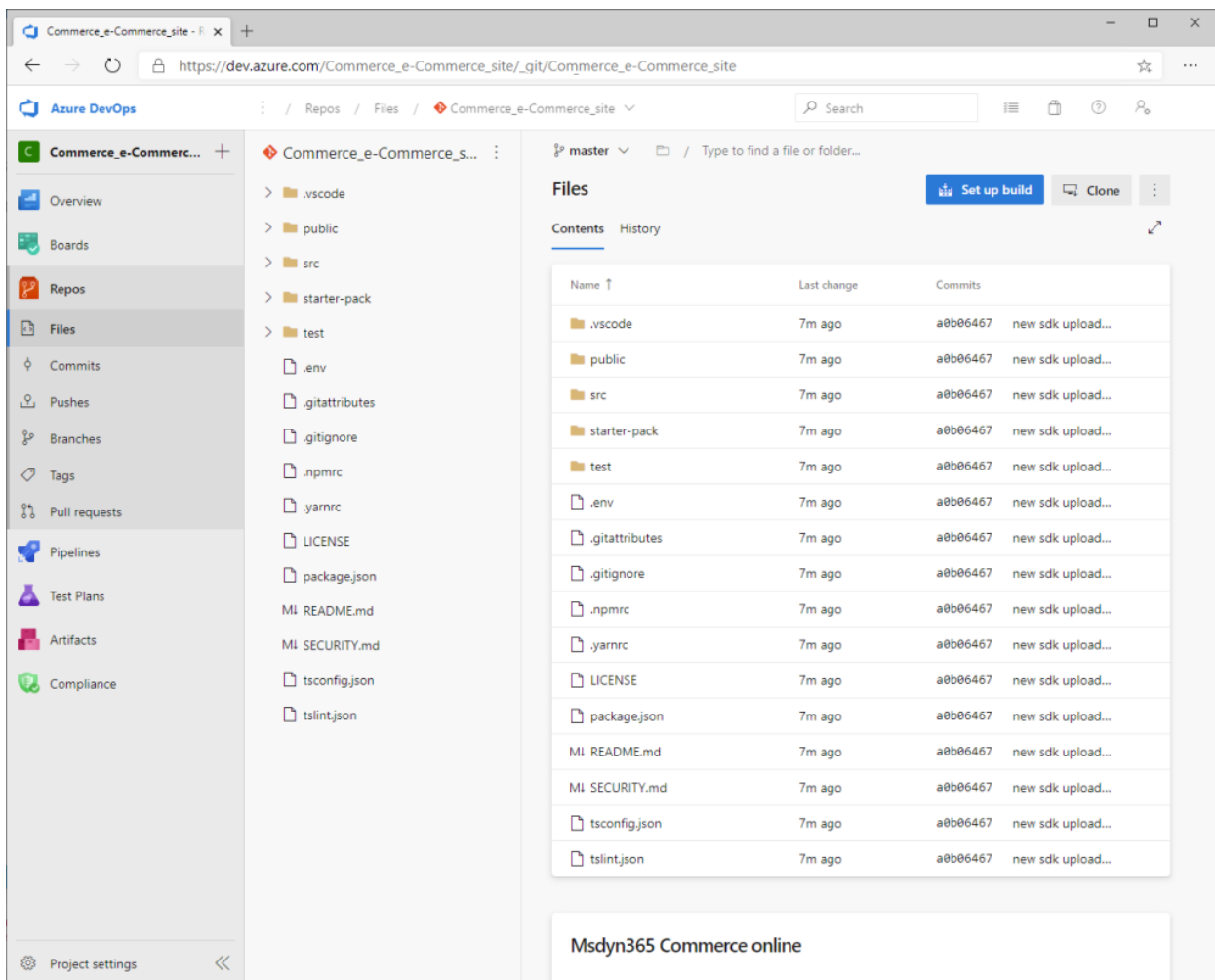
7. Copy all the contents of `C:\repos\Msdyn365.Commerce.Online` to `C:\repos\<DevOpsProjectName>`. Don't copy the hidden `.git` folder.
8. In Visual Studio Code, open the `c:\repos\<DevOpsProjectName>` folder. When you select the **Source Control** button on the left, Visual Studio Code shows the new changes that must be committed.
9. To commit all the changes to Git, enter a description in the field at the top of the **Source Control: Git** pane, and then select the check mark symbol above it. When you're prompted to stage all your changes and commit them directly, select **Yes**.



10. Select the ellipsis (...) to the right of the check mark symbol, and then, on the menu that appears, select Push to push the changes to the repo.



In Azure DevOps, you should now see the new files.

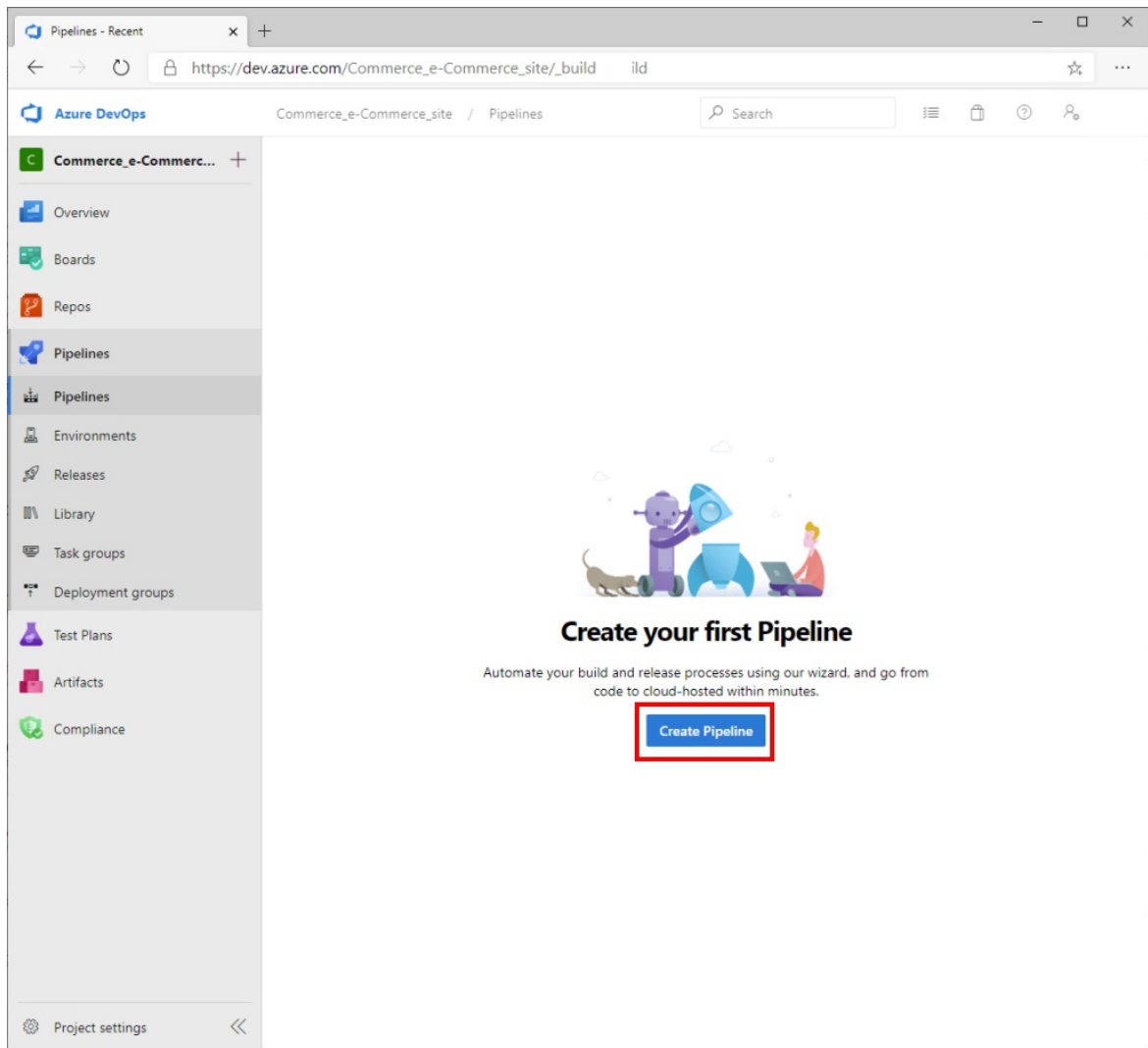


## Create and configure a new build pipeline in Azure DevOps

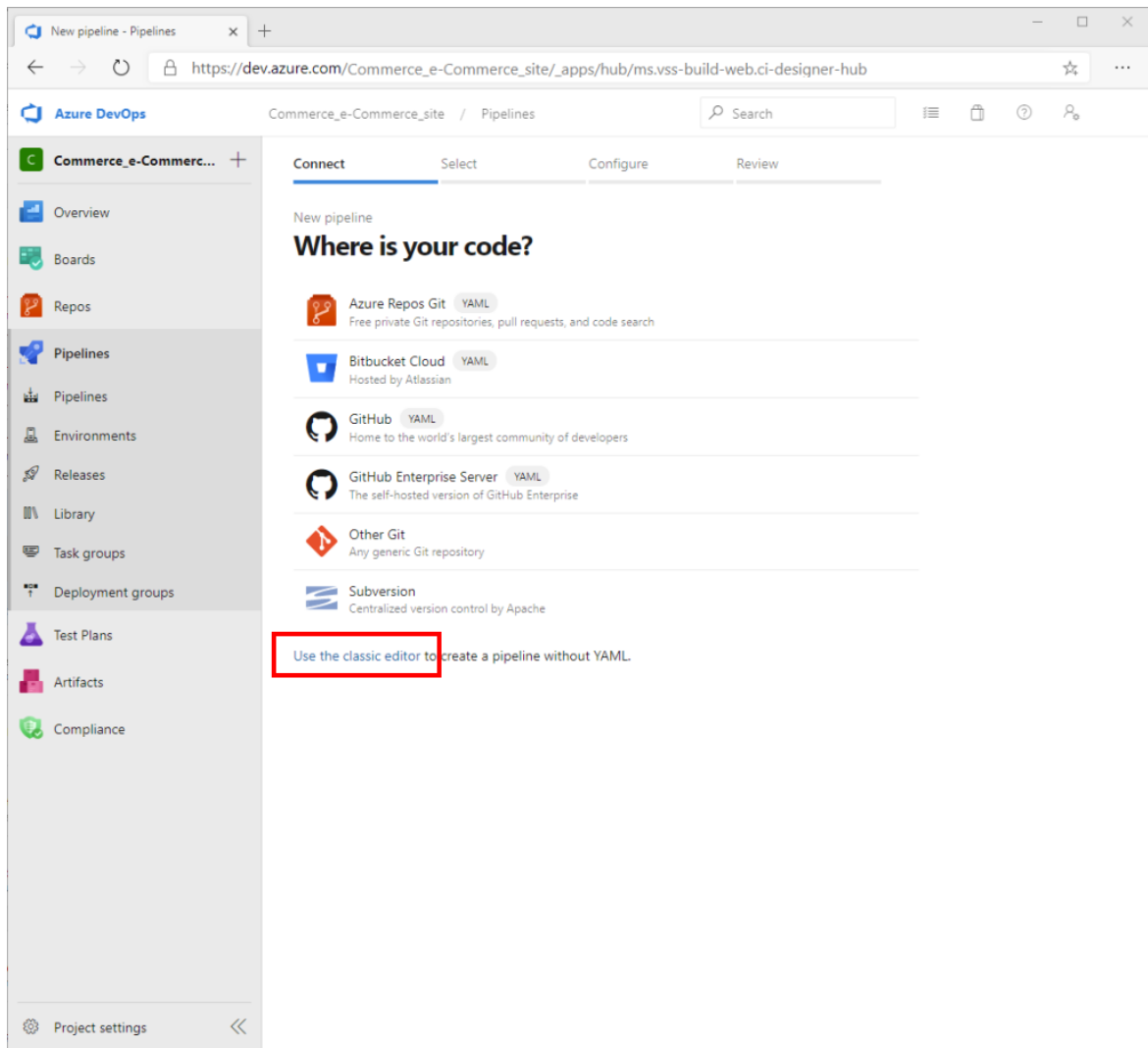
To create and configure a new build pipeline in Azure DevOps, follow these steps.

1. In the left navigation pane, under **Pipelines**, select **Pipelines**, and then select **Create Pipeline** in the main part of the page.

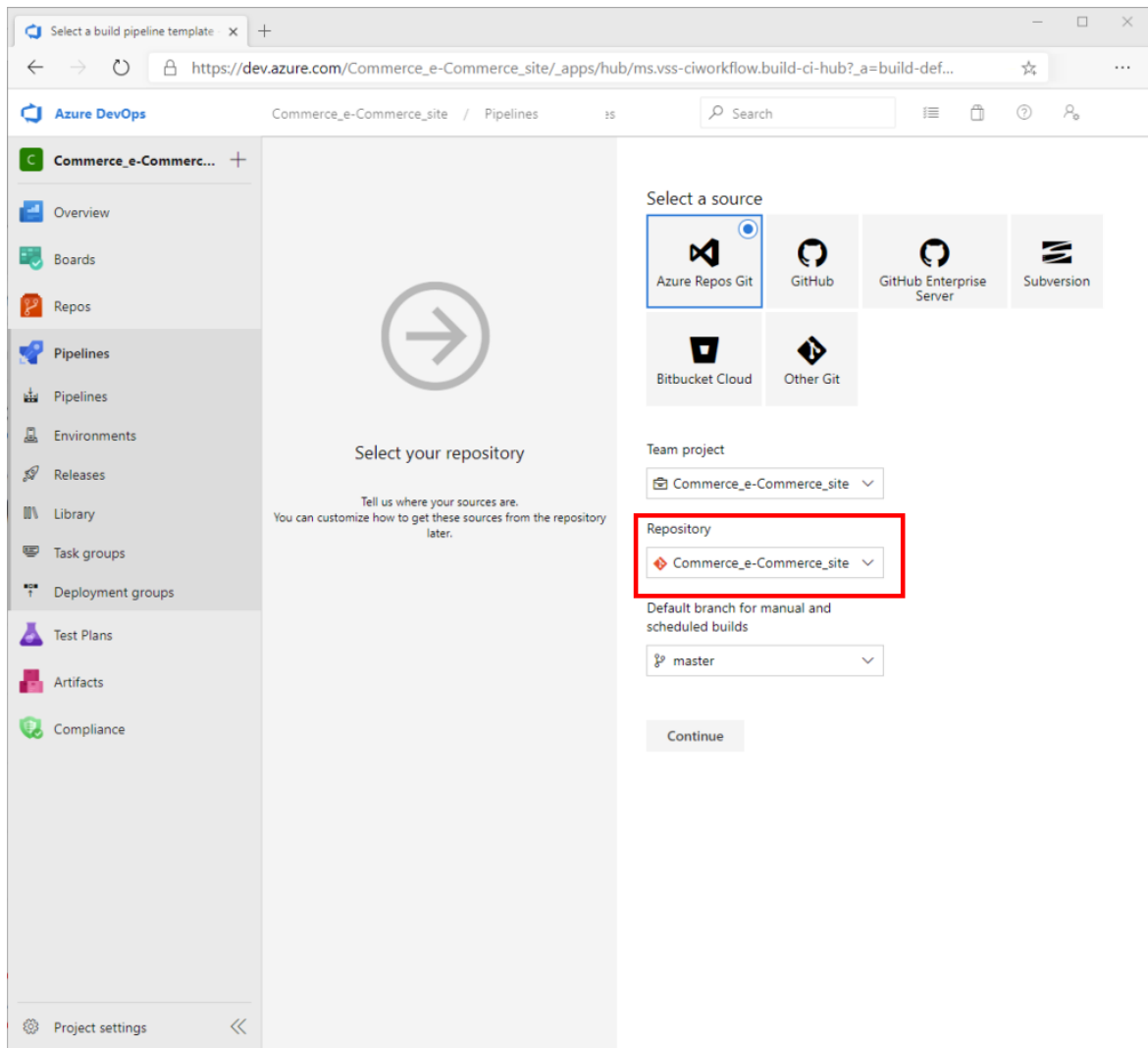




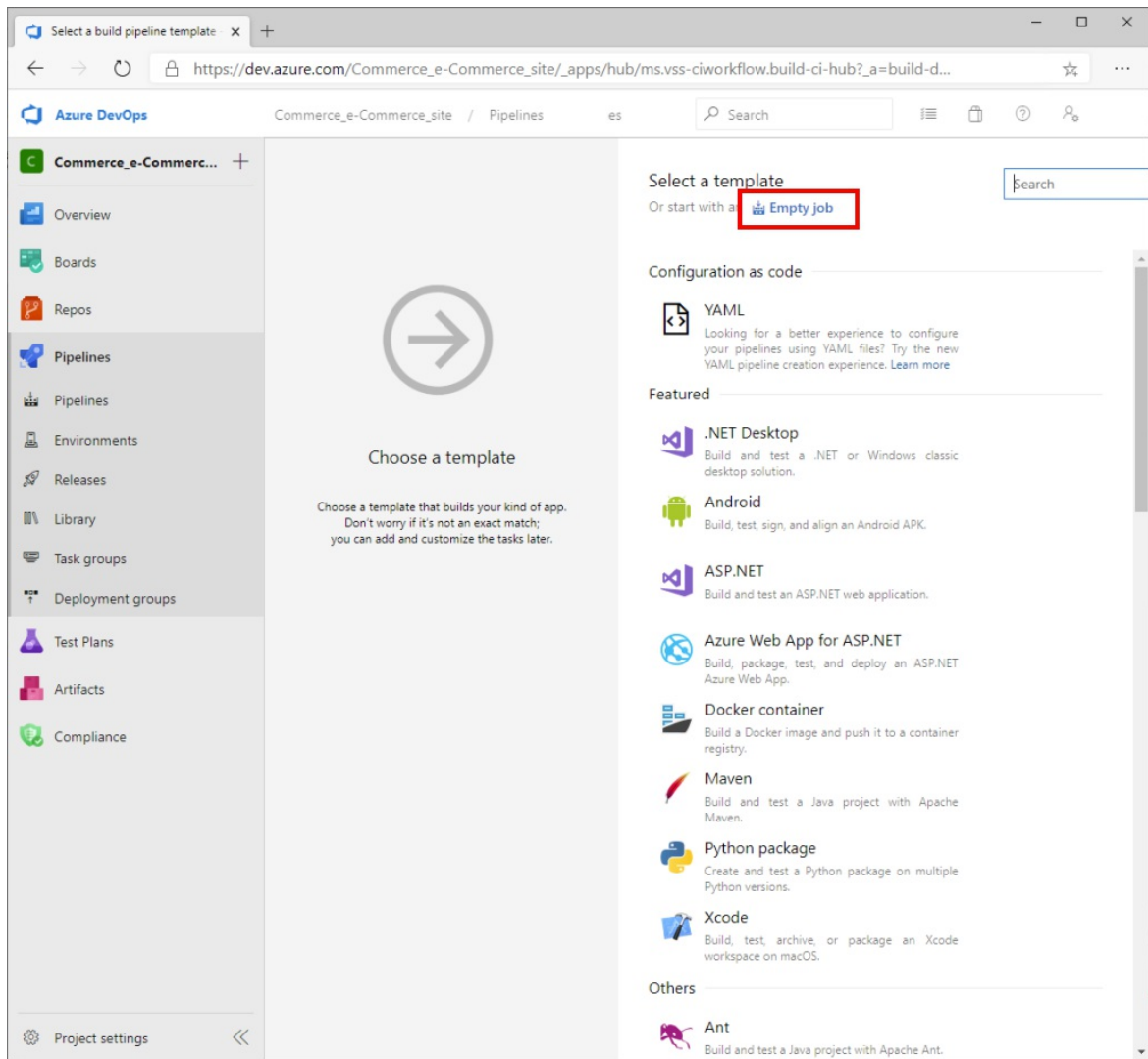
2. Select **Use the classic editor**.



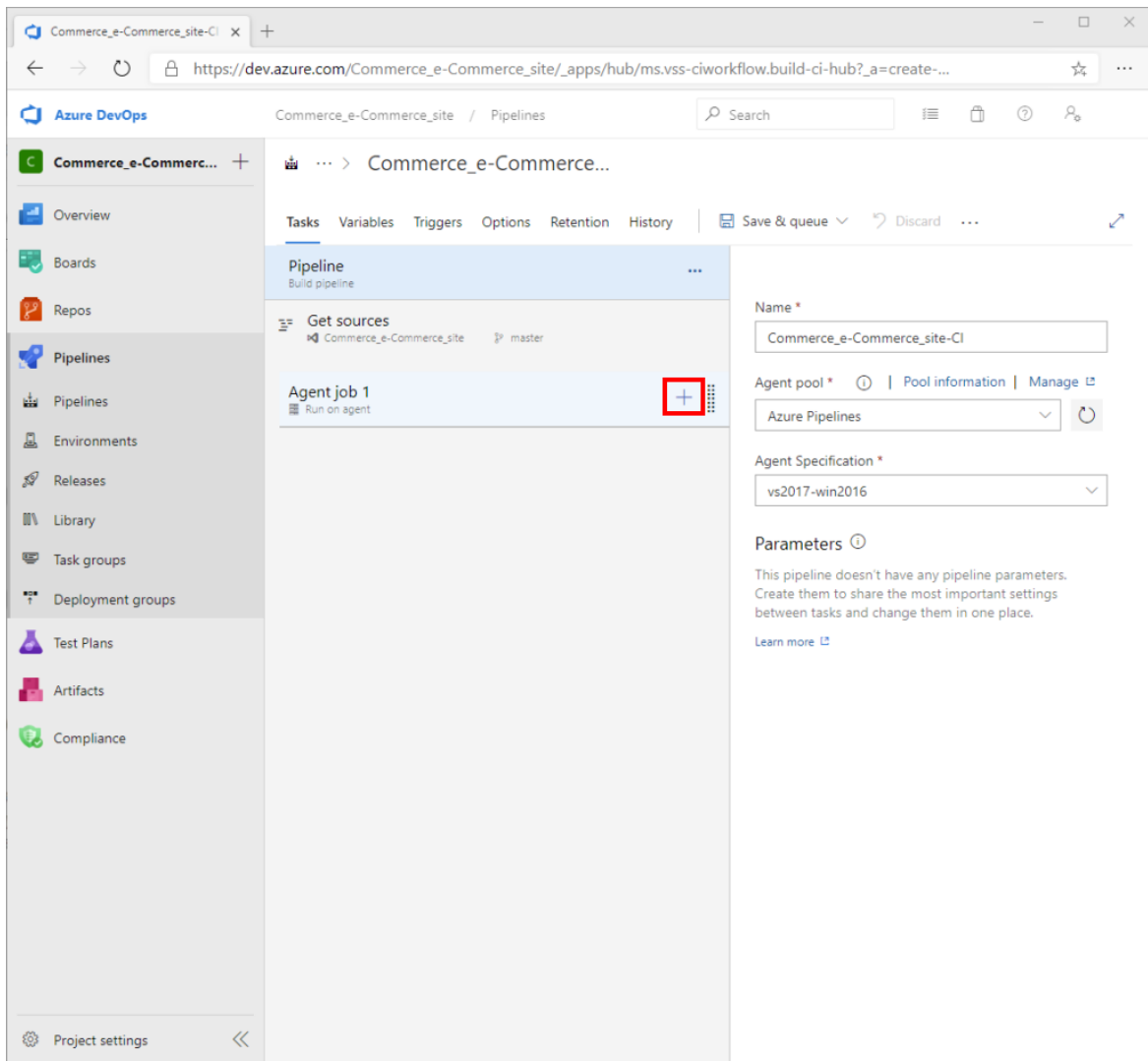
3. In the **Repository** field, select your Azure DevOps GitHub repo project, and then select **Continue**.



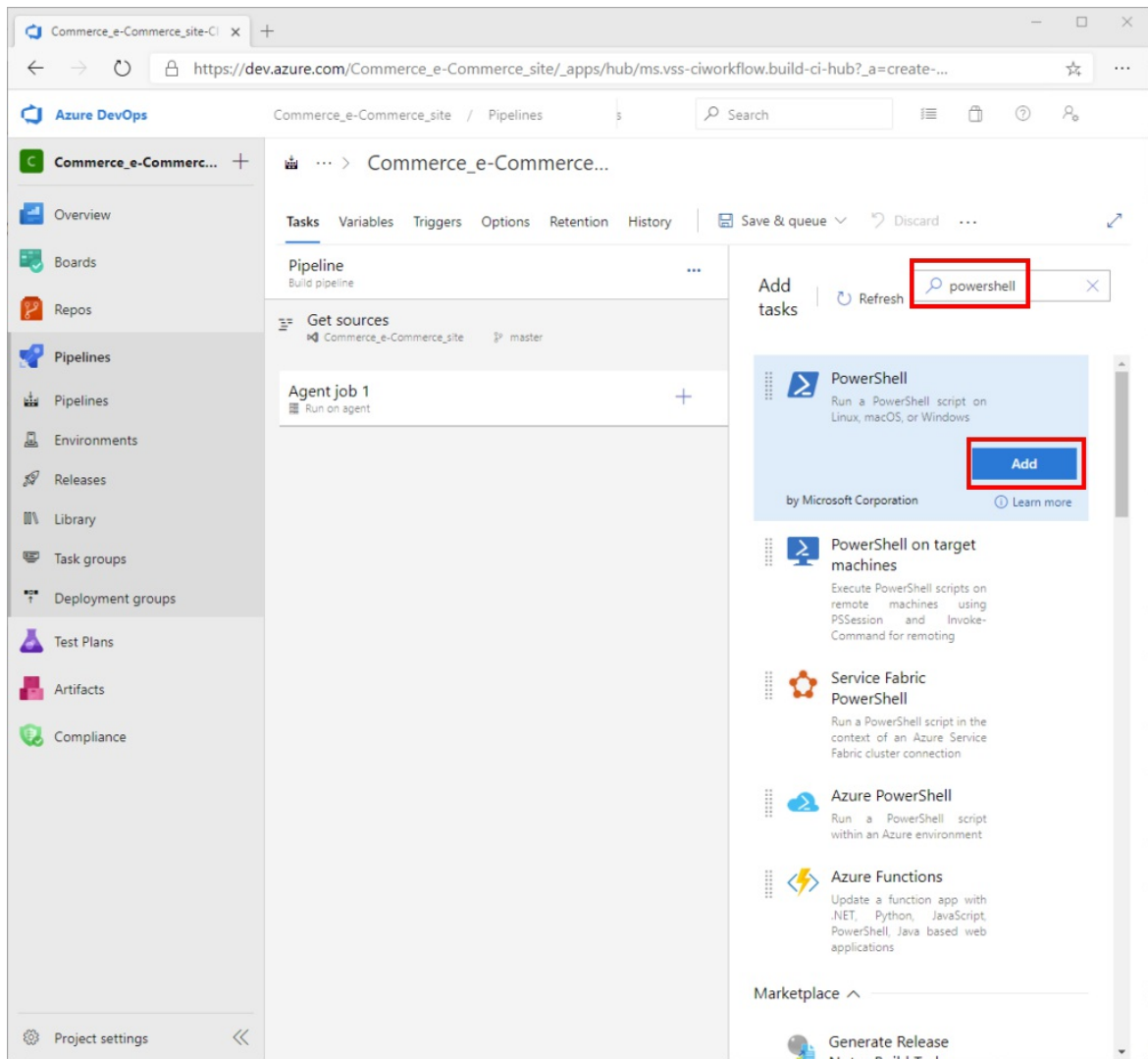
4. Under Select a template, select Empty job.



5. Next to **Agent job**, select the plus sign (+) to add a new agent job.

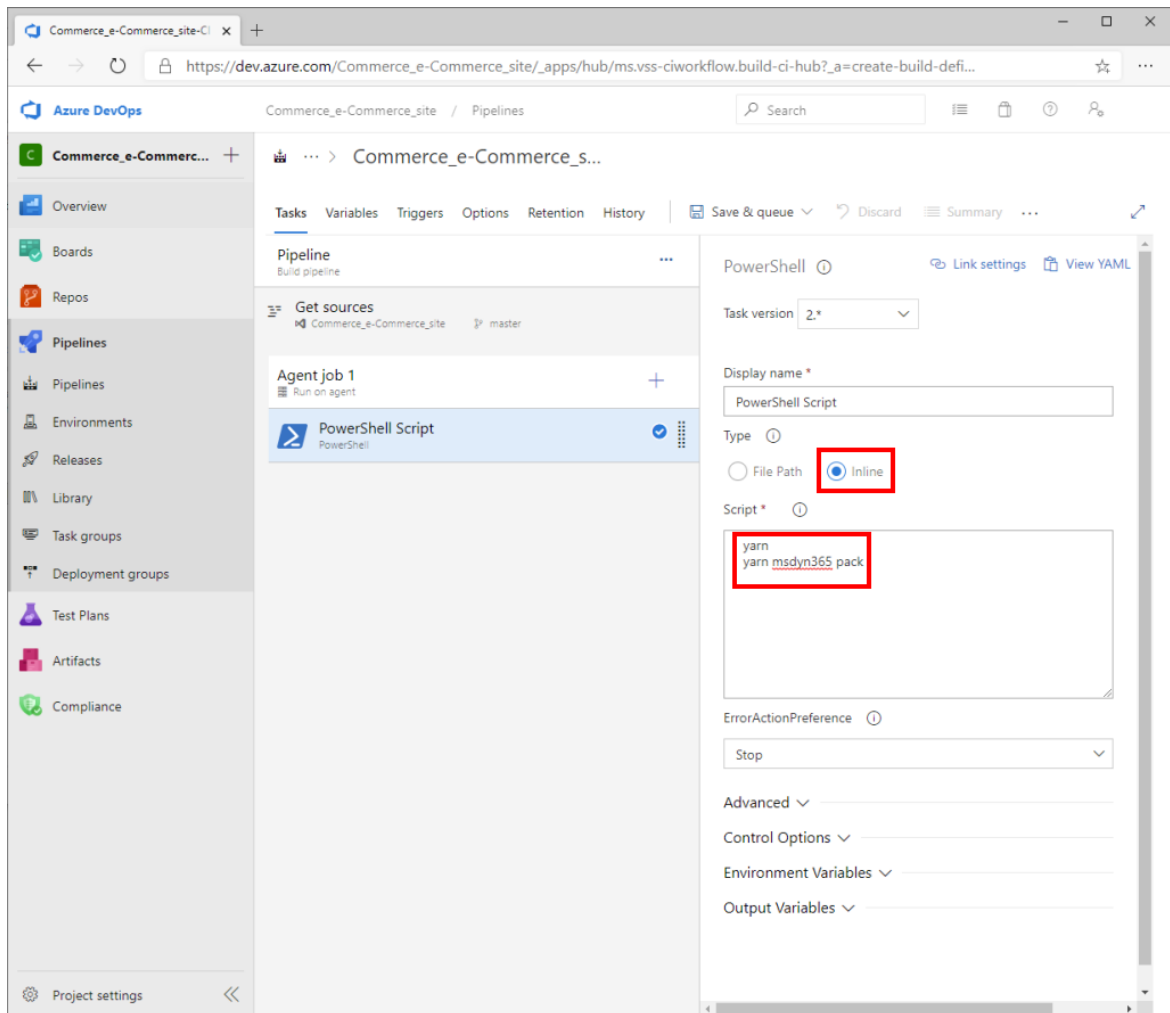


6. In the **Add tasks** pane on the right, search for "PowerShell," and then, in the **PowerShell** task, select **Add**.

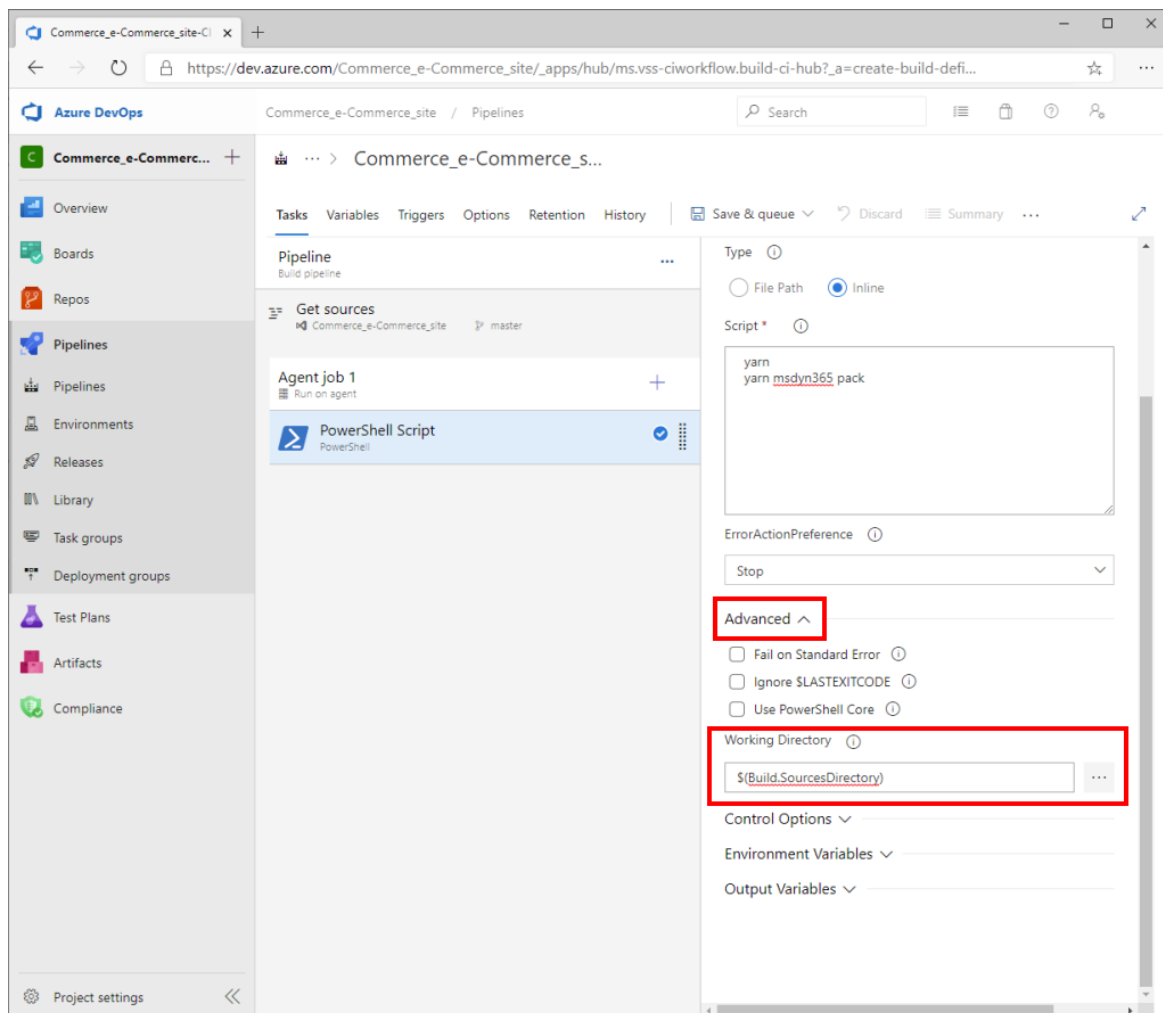


7. In the main part of the page, select **PowerShell Script**. Then, in the **PowerShell** pane on the right, under **Type**, select **Inline**. Copy the following script into the **Script** field.

```
yarn  
yarn msdyn365 pack
```

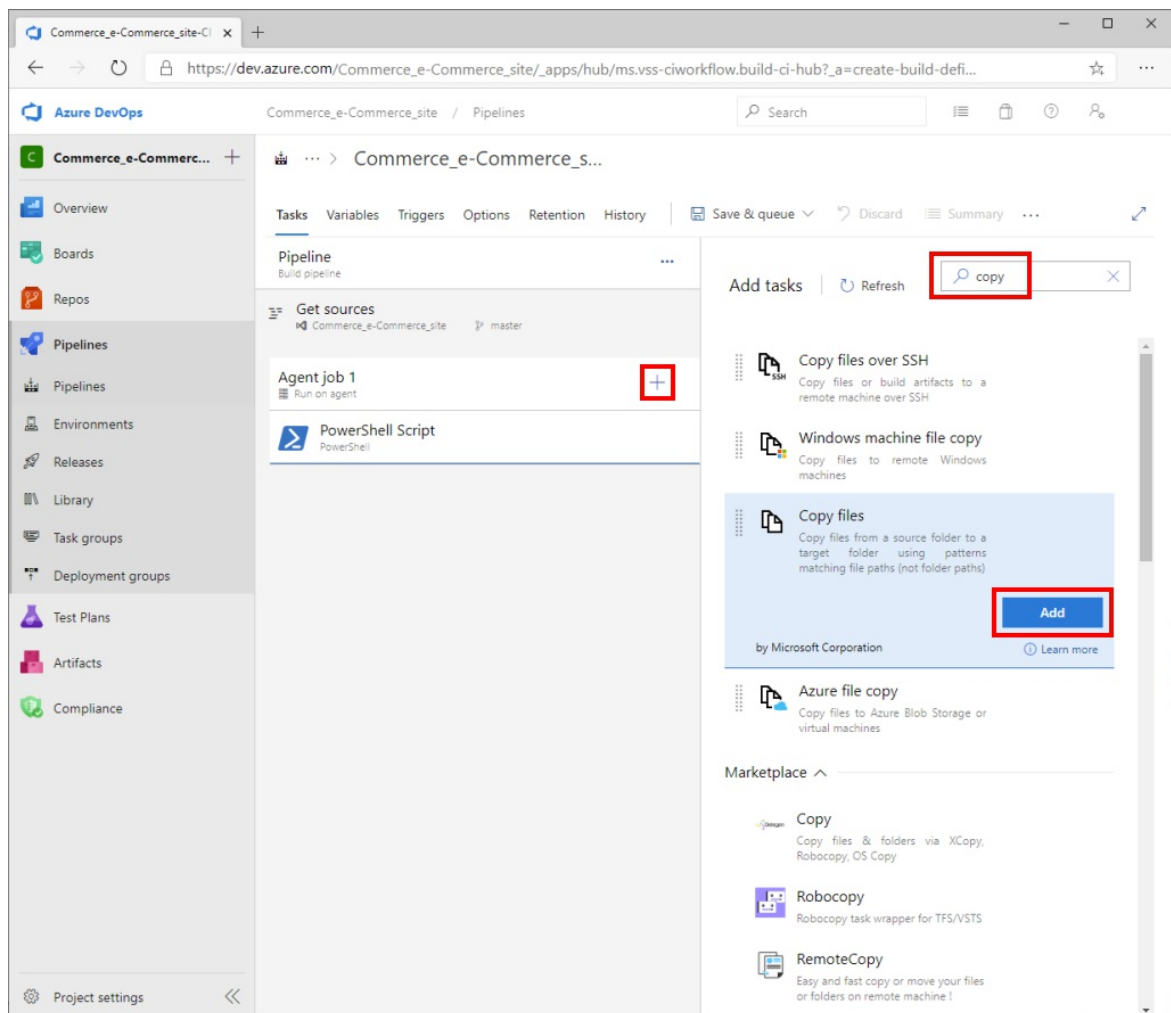


8. In the right pane, on the **Advanced** FastTab, in the **Working Directory** field, enter `$(Build.SourcesDirectory)`.



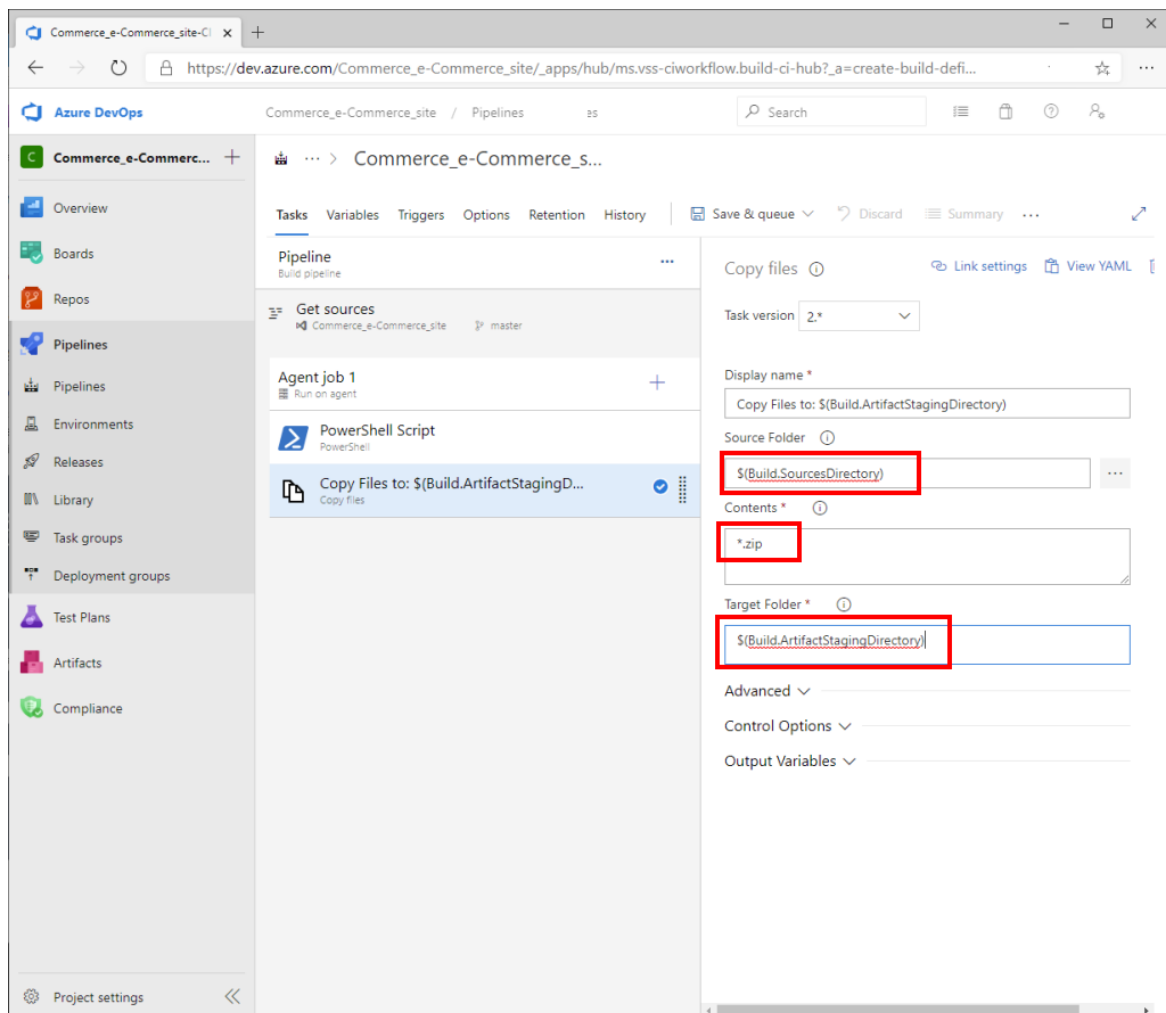
9. In the main part of the page, next to **Agent job**, select the plus sign (+) to add a new agent job.
10. In the **Add tasks** pane on the right, search for "copy," and then, in the **Copy files** task, select **Add**.





11. In the main part of the page, select the Copy files task, and then, in the Copy files pane on the right, follow these steps:

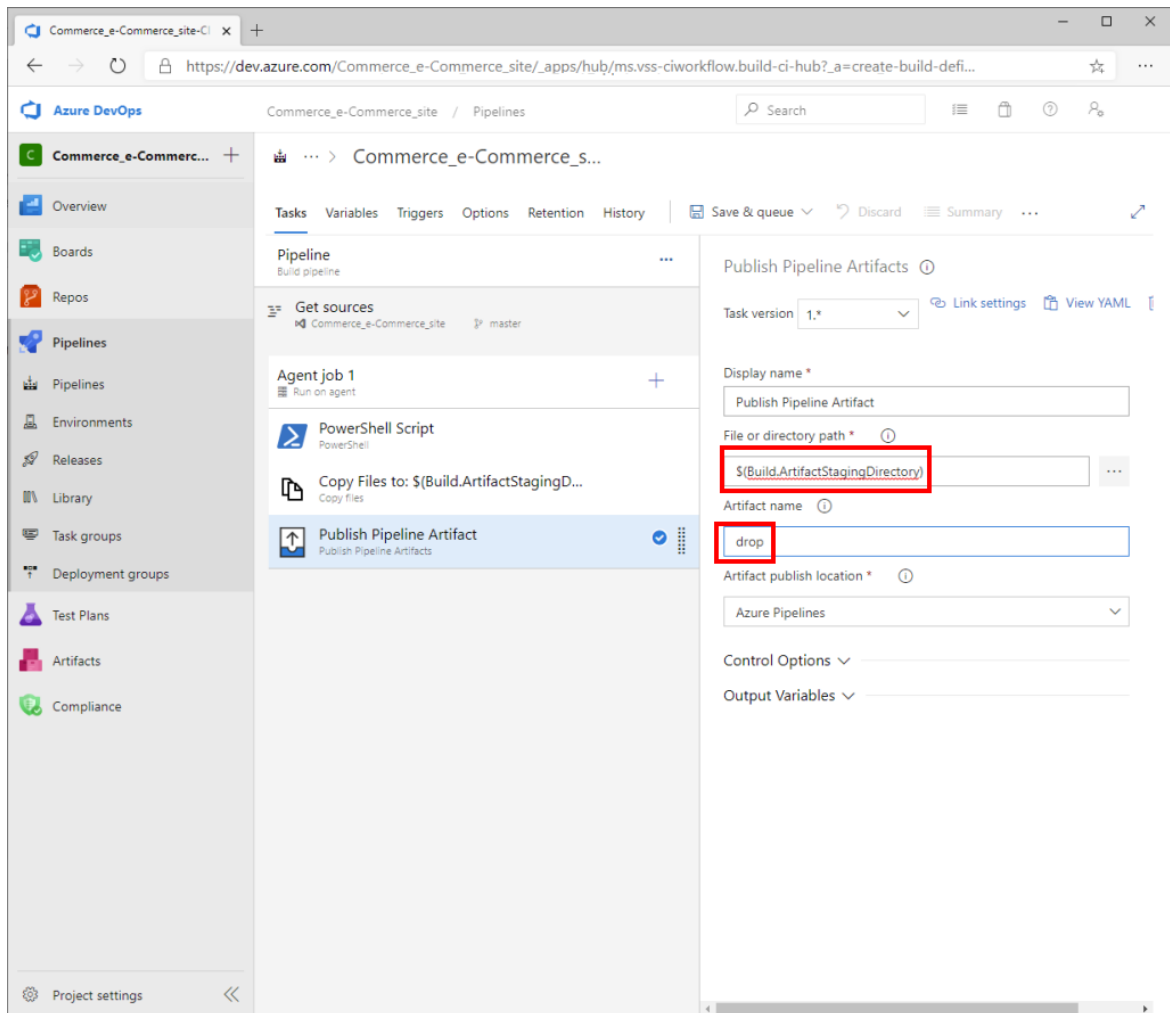
- a. In the Source Folder field, enter `$(Build.SourcesDirectory)`.
- b. In the Contents field, enter `*.zip`.
- c. In the Target Folder field, enter `$(Build.ArtifactStagingDirectory)`.



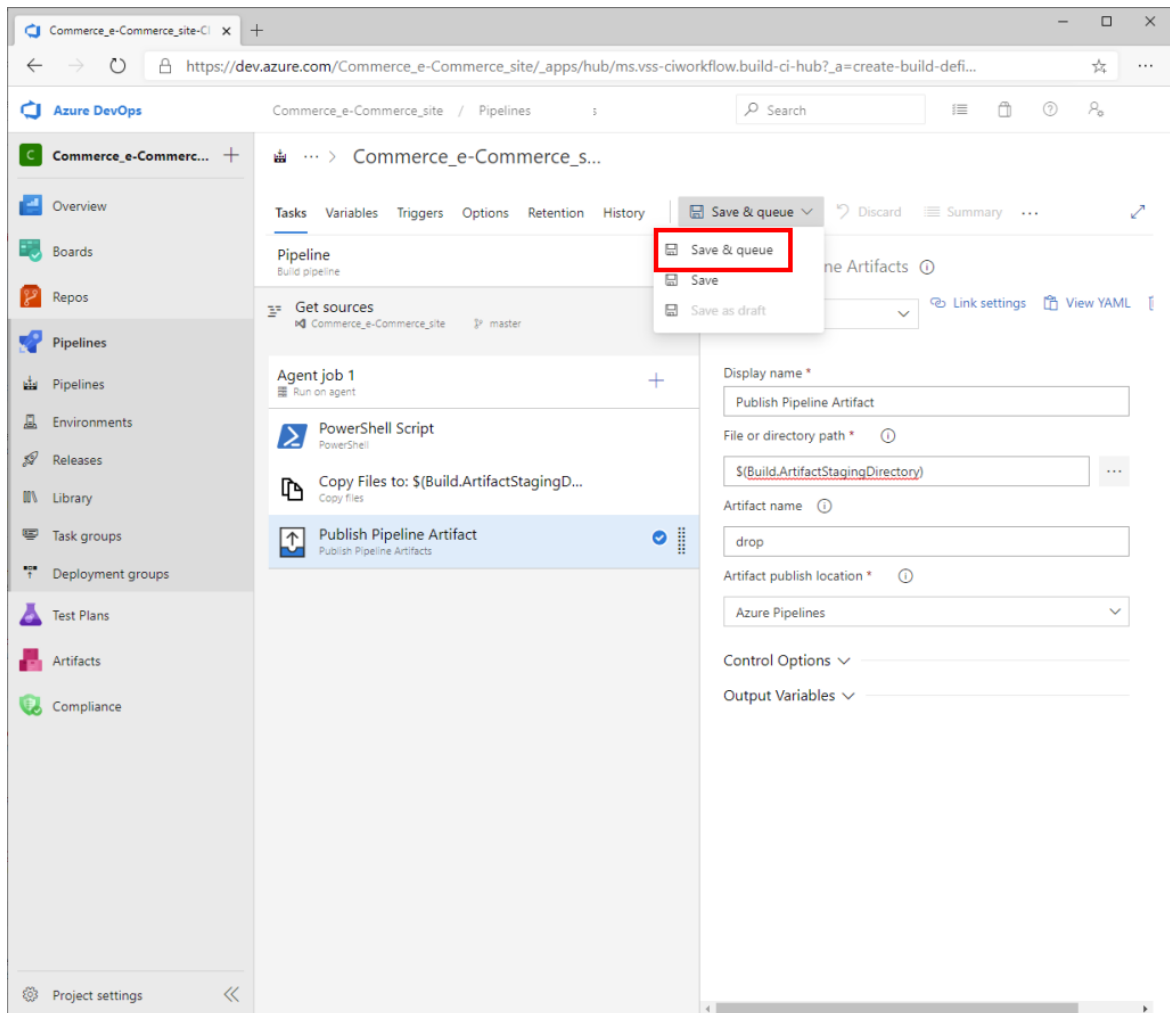
12. In the main part of the page, next to **Agent job**, select the plus sign (+) to add a new agent job.
13. In the **Add tasks** pane on the right, search for "publish," and then, in the **Publish Pipeline Artifacts** task, select **Add**.

![Azure DevOps "Add tasks" pane with search box and "Publish Pipeline Artifacts" task "Add" button highlighted]

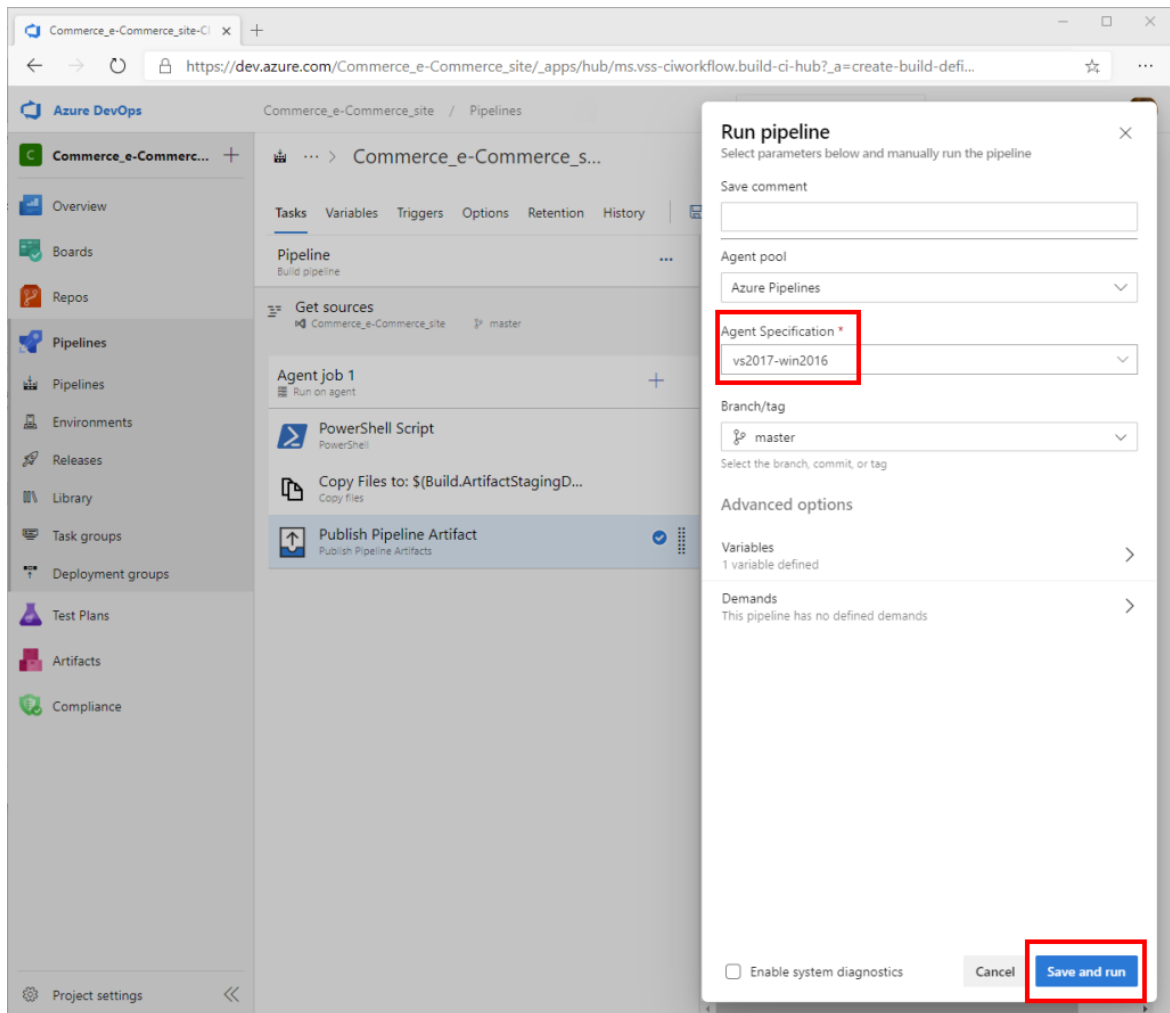
14. In the main part of the page, select the **Publish Pipeline Artifacts** task. Then, in the **Publish Pipeline Artifacts** pane on the right, follow these steps:
  - a. In the **File or directory path** field, enter `$(Build.ArtifactStagingDirectory)`.
  - b. In the **Artifact name** field, enter `drop`.



15. On the toolbar, select **Save & queue**.

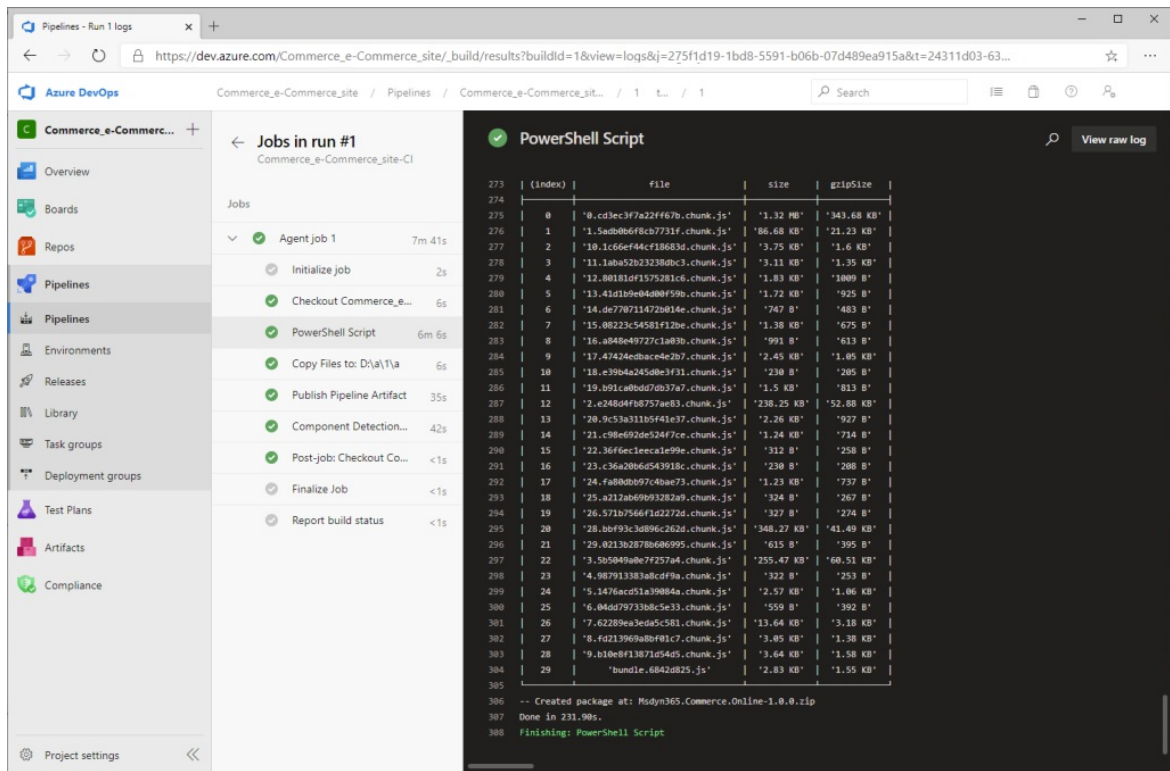


16. In the **Run pipeline** dialog box, make sure that the **Agent Specification** field is set to **vs2017-win2016**, and then select **Save and run**.



Tools that you typically use to build, test, and run JavaScript apps (such as npm, Node, Yarn, and Gulp) are preinstalled on Microsoft-hosted agents in Azure Pipelines. For the exact versions of Node.js and npm that are preinstalled, see the Microsoft-hosted agents. To install a specific version of these tools on Microsoft-hosted agents, add the **Node Tool Installer** task to the beginning of your process. Yarn is preinstalled on VS2017-win2016.

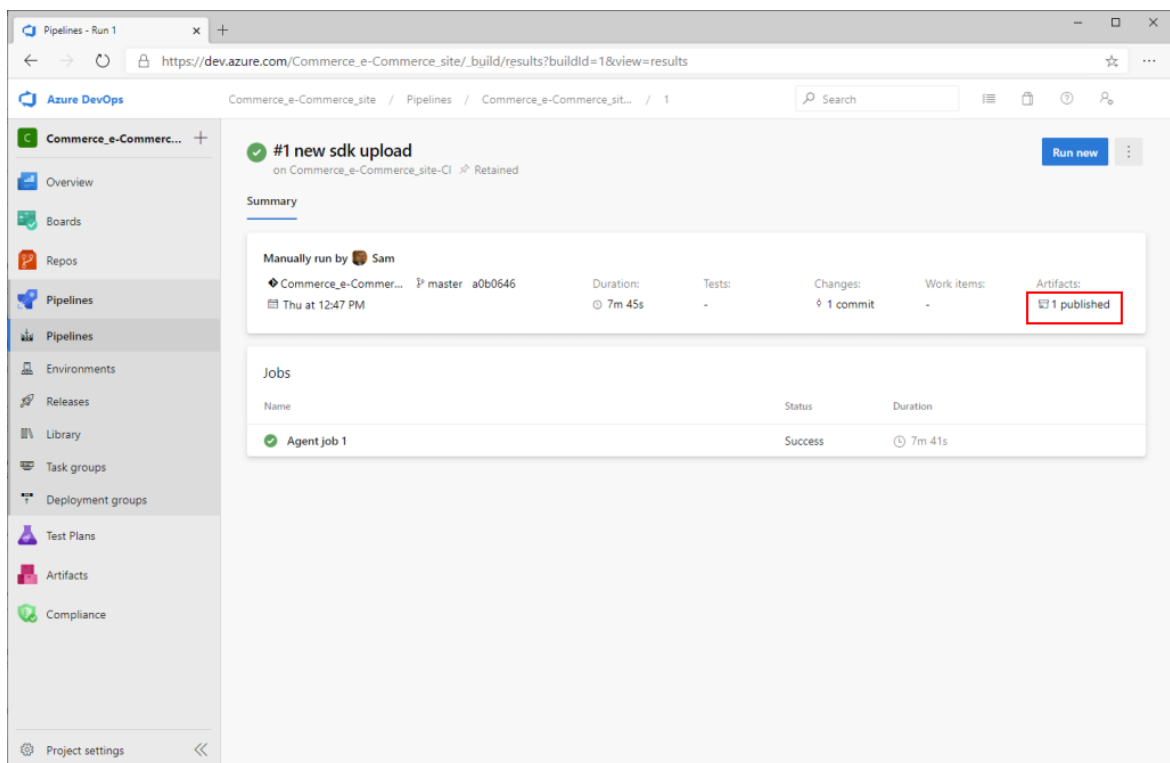
17. Monitor the agent job logs to learn when the job is completed.



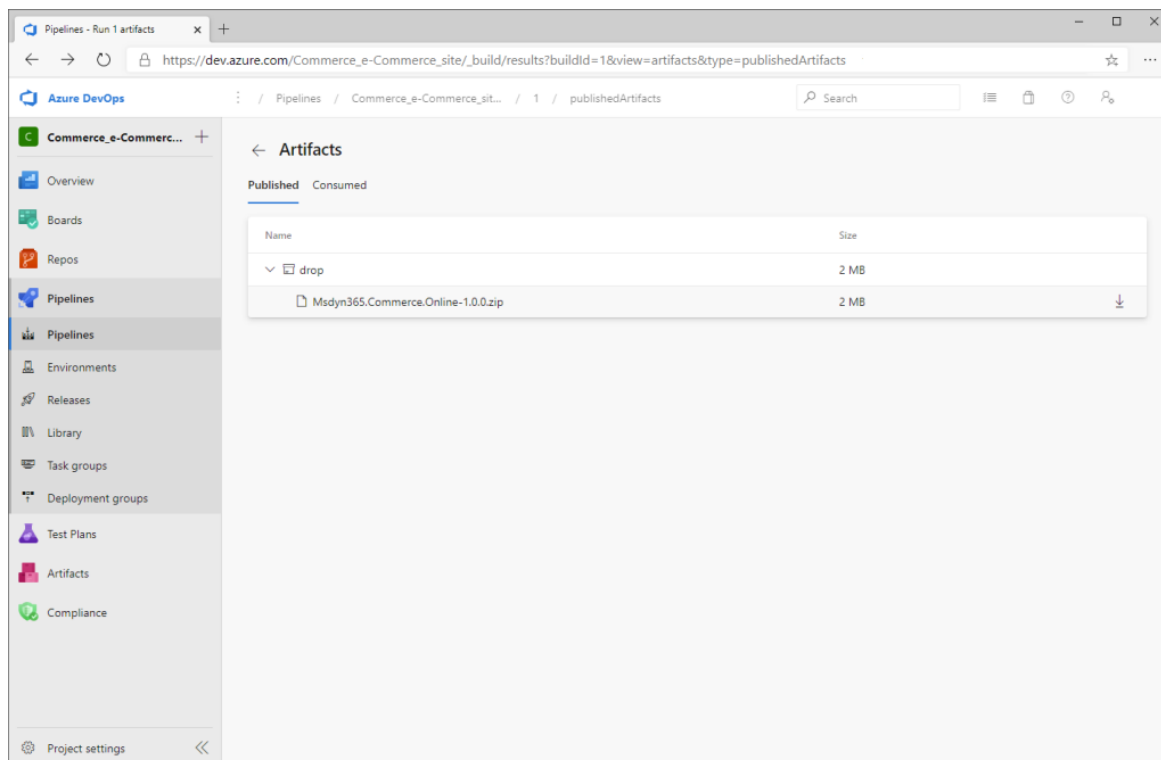
18. After the job is completed, in the left navigation pane, under **Pipelines**, select **Pipelines**. Then, on the **Runs** tab, under **All pipeline runs**, select the pipeline run to download the deployable package.

! [Azure DevOps "Pipelines" page with "Runs" tab and pipeline run highlighted]

19. Under **Summary**, under **Artifacts**, select **1 published**.



20. Select the **drop** folder to expand it and see the zip file that was created as part of the pipeline run. Select the **Download** button to download the file.



## Additional resources

[Get started with e-commerce online extensibility development](#)

[System requirements for a Dynamics 365 Commerce online extensibility development environment](#)

[Set up a development environment](#)

[Configure a development environment \(.env\) file](#)

[Configure an e-commerce development environment against a Commerce cloud environment](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# E-commerce architectural overview

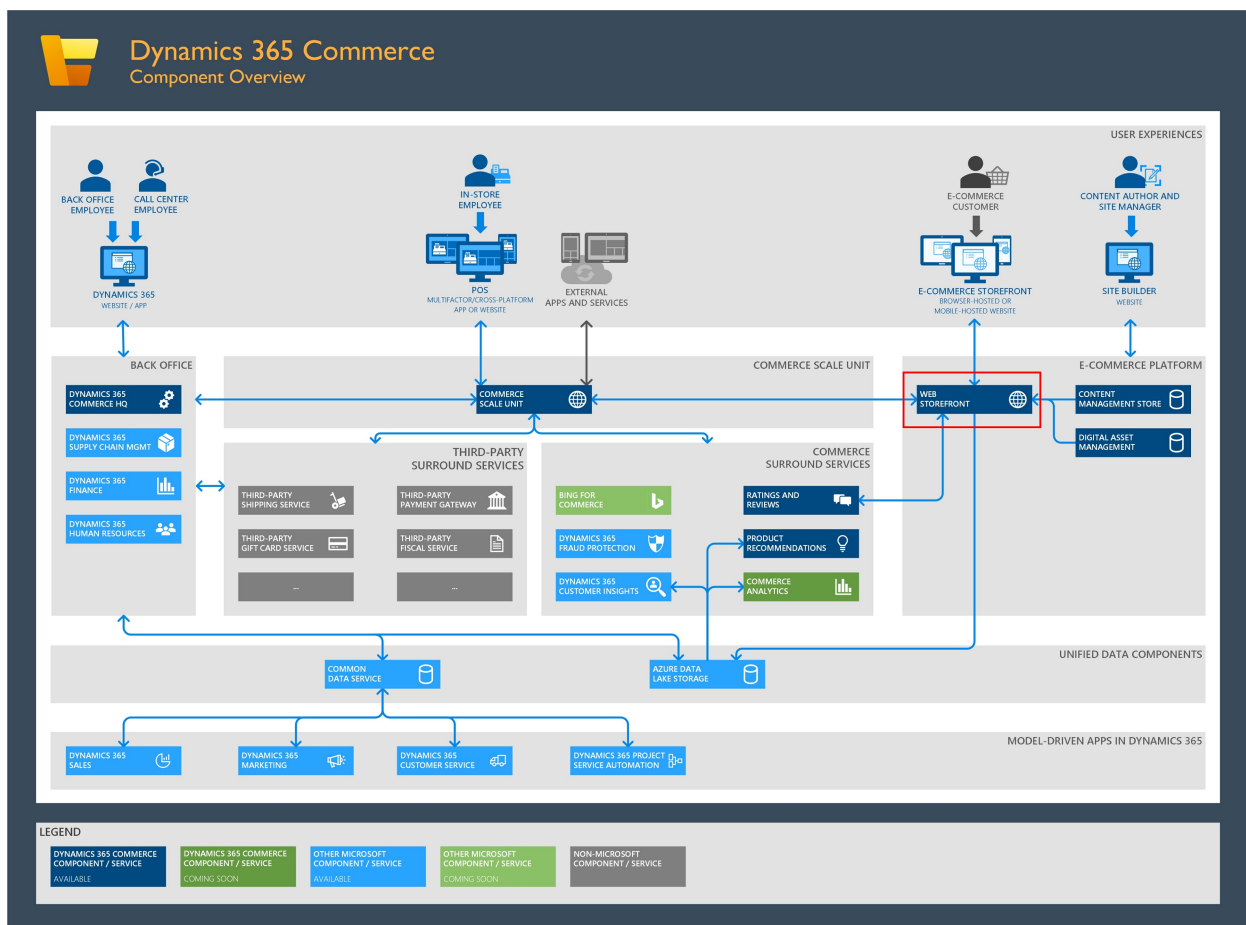
2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic presents an architectural overview of Microsoft Dynamics 365 Commerce.

## Overview

The Dynamics 365 Commerce online extensibility software development kit (SDK) lets partners easily extend their website so that it includes additional business logic and user experience (UX) logic. Partners can do this extension by using open-source technologies that are freely available.

This architectural overview will focus on the "Web Storefront" box highlighted in the following [Commerce architecture](#) illustration.



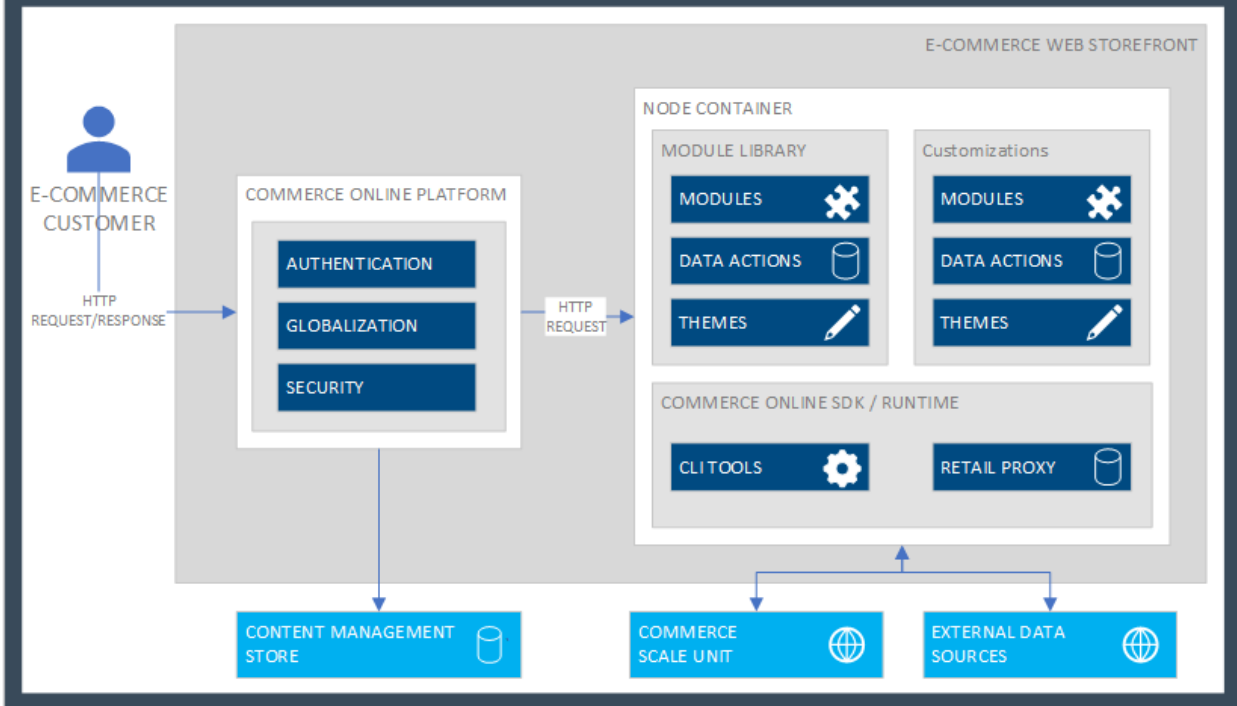
## Web storefront component-based architecture

The architecture of the platform takes advantage of a reuse-based approach to define, implement, and compose loosely coupled independent components. In this approach, the emphasis is on separation of concerns.

## High-level design

The extensibility design avoids dependency between the platform and the application by running the application as a microservice that is built on Node.js and that uses React as an underlying UX framework. The platform is run as a separate service, and handles all routing, integration with the content management system (CMS), and security.



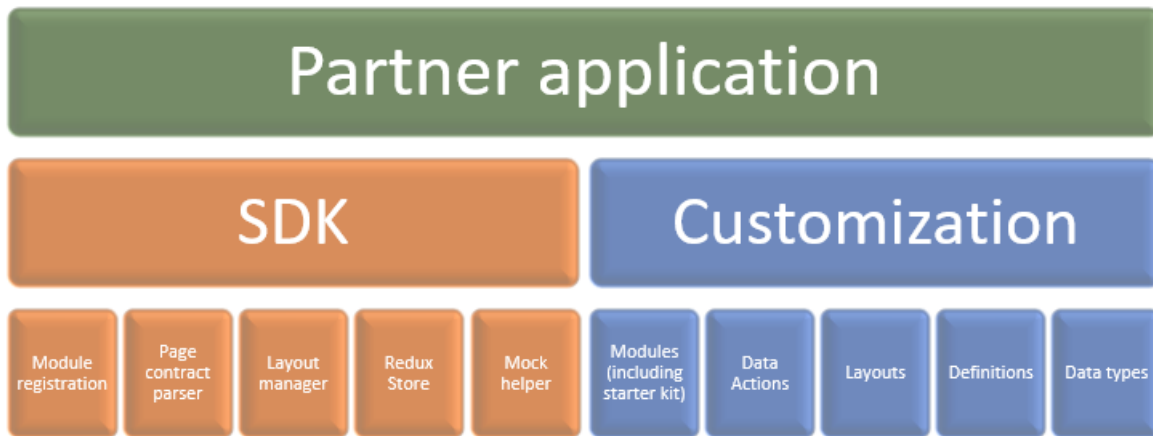


## Request flow

Here is a typical architectural flow when a customer requests a page from an online store that uses the Dynamics 365 Commerce platform.

1. The platform gets a request, and does authentication, routing, and locale detection.
2. The platform generates a hydrated page object that contains page configuration data and data from external services that use data actions.
3. Data actions allow for batching, aggregation, and chaining of multiple service calls, including deduplication.
4. The platform provides a set of core data actions that provide connections to Dynamics 365 Commerce services (for example, the catalog and ratings).
5. After the hydrated page object is generated, the platform calls the React application that is running on a node express server.
6. The React application parses the view model, runs the React views on the server side, and generates HTML.
7. The React application consists of core SDK modules, module library modules, and custom modules.
8. The platform sends back the HTML together with the appropriate cookies, headers, and so on.
9. The React script initializes the components, takes over client-side execution on the browser, and renders the modules.
10. Node.js supports tool pages and other developer productivity tools that are provided by the platform.

## Partner application



The compiled partner package contains both the SDK and a module library. The SDK isn't extensible, but module library modules can be cloned and completely customized. Partner customizations (modules, data actions, and themes) can be packaged by using a command-line interface (CLI) command. The package can then be uploaded by using Microsoft Dynamics Lifecycle Services (LCS). In this way, the customizations are incorporated into the partner's e-Commerce site.

## Additional resources

[E-commerce components](#)

[CLI command reference](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# E-commerce components

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic contains a high-level summary of some frequently used e-commerce configuration components that the Microsoft Dynamics 365 Commerce online software development kit (SDK) provides access to.

## Modules

Modules represent the core building blocks that make up an online page. Here are some examples of modules:

- A feature module that is featured on a page, and that shows a product image and description, together with a "call-to-action" button that can be used to purchase the product or get more information about it
- A hero module that highlights a campaign or provides marketing information
- A header module that is made up of smaller module components, such as a search module, a sign-in module, and a navigation module

## Data actions

Data actions are used to get data, apply business logic to a module, and share data across modules.

### Core data actions

The e-Commerce platform module library includes a set of data actions for performing typical actions, such as getting product data from the Dynamics 365 Commerce database, or getting ratings and reviews information for a product. Core data actions can't be modified.

### Custom data actions

You can create custom data actions and use them in your modules. Custom data actions can be globally scoped so that they can be used across multiple modules. Alternatively, they can be used in a single module. Custom data actions can call Dynamics 365 Commerce proxy application programming interfaces (APIs), or any other service provider, to get data. Custom business logic can be applied as required.

## Themes

Themes contain site-wide Sassy Cascading Style Sheets (SCSS) style definitions. They also let you add custom, module-specific SCSS style definitions. You can set a site theme in the authoring tools. All pages then use that theme by default. You can add more themes, and set them on a template, a layout, or a specific page. This capability is useful if you want to change the theme for a campaign or a temporary seasonal change. You can set the theme on the whole site or a subset of pages.

Themes can also override module views (including module library modules), and can add module definition extensions to extend the configurations and resources on a module. Because themes can change the look and feel of all modules (custom and module library) without requiring that modules be cloned, they allow for better serviceability.

## Script injectors

The online platform provides a built-in script injector module that makes it easy to inject scripts into the head, bodybegin, and so on, from the admin tool. Script injectors make it easy to add scripts such as third-party analytics. You might have advanced requirements to use custom script injector modules to inject custom HTML into your online site.

# Additional resources

[E-commerce architectural overview](#)

[CLI command reference](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# CLI command reference

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the command-line interface (CLI) commands that are available in the Microsoft Dynamics 365 Commerce online software development kit (SDK).

## Overview

All the following commands must be run by using Yarn. They all have the following structure.

```
yarn msdyn365 {command} {command-arguments}
```

For information about each command, see the entries in this topic, or use the `yarn --help` or `yarn msdyn365 {command} --help` command.

## add-component-override

### Usage

```
yarn msdyn365 add-component-override <themeName> <componentName> <--list-components>
```

This command adds a component to the specified theme component folder. The component can then be modified as desired from that folder.

The `--list-components` option is used to show a list of components.

### Examples

```
yarn msdyn365 add-component-override spring-theme add-to-cart.component
```

```
yarn msdyn365 add-component-override --list-components
```

## add-data-action

### Usage

```
yarn msdyn365 add-data-action <action-name>
```

This command adds a template data action to the root/src/actions folder.

### Example

```
yarn msdyn365 add-data-action getMyData
```

## add-module

### Usage

```
yarn msdyn365 add-module <module-name>
```

This command adds a module to the root/src/modules folder. Note that module names are case-insensitive.

#### Example

```
yarn msdyn365 add-module product-feature
```

## add-theme

#### Usage

```
yarn msdyn365 add-theme <theme-name>
```

This command adds a theme to the root/src/themes folder.

#### Example

```
yarn msdyn365 add-theme spring-theme
```

## add-view-extension

#### Usage

```
yarn msdyn365 add-view-extension <theme-name> <Module-name>
```

This command adds a module view extension to the root/src/themes/<theme-name>/views folder. The theme can then add more module definition items, such as configurations, resources, and slots.

#### Example

```
yarn msdyn365 add-view-extension spring-theme product-feature
```

## clone

#### Usage

```
yarn msdyn365 clone <module-library-module-name> <new-module-name>
```

This command creates a renamed copy of a module library module and adds the source code to the local root/src/modules folder.

#### Example

```
yarn msdyn365 clone content-block super-content-block
```

## pack

#### Usage

```
yarn msdyn365 pack
```

This command creates a package of the local site configurations (modules, data actions, themes, and so on). This package will then be uploaded to the node server by using Microsoft Dynamics Lifecycle Services (LCS). This command should be run from the root directory of your local SDK files.

The output is a zip file in the directory that the command was run from. The file name is built by using the name and version that are found in your SDK package.json file. For example, a zip file might be named **@msdyn365-commerce-partners-fabrikam-1.2.73.zip**.

#### Example

```
yarn msdyn365 pack
```

## packages

### Usage

```
yarn msdyn365 packages
```

This command prints the packageVersions.json file that was generated at build time to the console.

The packageVersions.json file includes information about Dynamics 365 Commerce packages and Dynamics 365 Commerce module packages, their versions, and how the versions that are used were determined.

#### Example

```
yarn msdyn365 packages
```

## update-versions

### Usage

```
yarn msdyn365 update-versions <tag>
```

This command updates the SDK versions to the latest alpha/rc/release, based on the tag. The default tag is **latest**.

#### Example

```
yarn msdyn365 update-versions latest
```

## validate

### Usage

```
yarn msdyn365 validate <path/to/directory>
```

This command runs a series of validation checks on your package and any modules in the package. Specifically, it makes sure that each module has a valid definition .json file.

The path is the full path of the package folder that contains the package.json file.

### Example

```
yarn msdyn365 validate ./
```

## Additional resources

[E-commerce architectural overview](#)

[E-commerce components](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Modules overview

2/18/2021 • 14 minutes to read • [Edit Online](#)

This topic provides an overview of module development in Microsoft Dynamics 365 Commerce. It includes best practices about when and how to extend a module's style or features.

## Modules in Dynamics 365 Commerce

In Dynamics 365 Commerce, modules represent the core building blocks that make up an online Commerce page.

Here are some examples of modules that are used on Commerce site pages:

- A content block module that is featured on a page, and that shows a product image and description, together with a "call-to-action" button that can be used to purchase the product or get more information about it
- A promo banner module that highlights a promotion as a banner on a page
- A header module that consists of smaller module components, such as a search module, a sign-in module, and a navigation module
- A script module that injects JavaScript into the HTML on a page

## Anatomy of a module

Modules are built by using React components, and they consist of various files. For an overview of each file type, see the following topics:

- [Module definition file](#)
- [Module React component file](#)
- [Module view file](#)
- [Module data file](#)
- [Module mock file](#)
- [Module test file](#)
- [Module props.autogenerated.ts file](#)

For more information about how and when to use modules in Dynamics 365 Commerce, see [Work with modules](#).

## Get started with module development

The Commerce module library provides a broad set of modules that you can use on your online Commerce site. However, to meet your business requirements, you might have to do some customization of modules or themes. To customize modules and themes, you use the Dynamics 365 Commerce online software development kit (SDK) in a development environment.

To get started, you must install a set of tools and the online SDK. For step-by-step instructions that show how to set up a develop environment and create your first custom module, see [Set up a development environment](#).

## Module library modules

Dynamics 365 Commerce provides a library of modules that you can use when you build your online Commerce pages. These modules can be customized in various ways. For example, you can make code-free

style changes, you can make configuration changes through code in the online SDK. For more information about the module library, see [Module library overview](#).

### Code-free style changes

By using Commerce site builder, you can make the following code-free style changes:

- [Style presets](#) allow for a limited set of style changes for each module.
- [CSS overrides](#) are used to override any module's Cascading Style Sheets (CSS).

### Configuration changes through the online SDK

By using the online SDK, you can make custom theme changes to achieve the following results:

- [Modify the module CSS](#) to change any of a module's styling.
- [Extend a module's view](#) to change the rendered HTML layout.
- [Extend a module's definition file](#) to add or remove configuration fields, slots, data actions, or resources.
- [Override a module library component](#) to change the component behavior.

Here are some additional customization options:

- [Data action overrides](#) allow for the customization of any built-in data action.
- [Clone a module library kit module](#) to create a new copy of a module in the module library and change any part of it. This option is used for changes that can't be made through any of the other methods.
- [Create a custom module](#) when you require a custom solution that the modules in the module library don't provide.

The following section provides more details about the best option for various needs.

## Choose the correct customization option for module library modules

As you saw earlier, many options are available for customizing a module library module. It's important that you choose the correct option for your needs.

For example, if you choose to clone a module library module, you will make a brand-new copy of the module. However, if the module library module that the cloned module is based on is updated in a future module library release, you won't receive updates to the code and must manually update the cloned module. Therefore, cloning should be used only when it's absolutely necessary.

The following information should help you choose the correct solution for your needs.

### Style changes

If you have a scenario where a module library module does everything that you need it to do, but you want to change the module's look and feel, you can usually achieve this goal through CSS changes. For example, CSS changes let you hide unwanted elements in a module. A site's [theme](#) contains the CSS for all modules in Sassy Cascading Style Sheets (SCSS) files. In some cases, if the CSS changes don't require any module or theme customizations, you can make them directly in Commerce site builder. Both of the following options don't require that you use the online SDK. They are the easiest and fastest ways to make CSS changes to a module.

#### Style presets

A [style preset](#) is a stored set of all authorable style values across a site's theme. It can be used in Commerce site builder to immediately change the look of a site. In site builder, you can access style presets on the **Design** tab. Both global styles and module-specific styles can be modified.

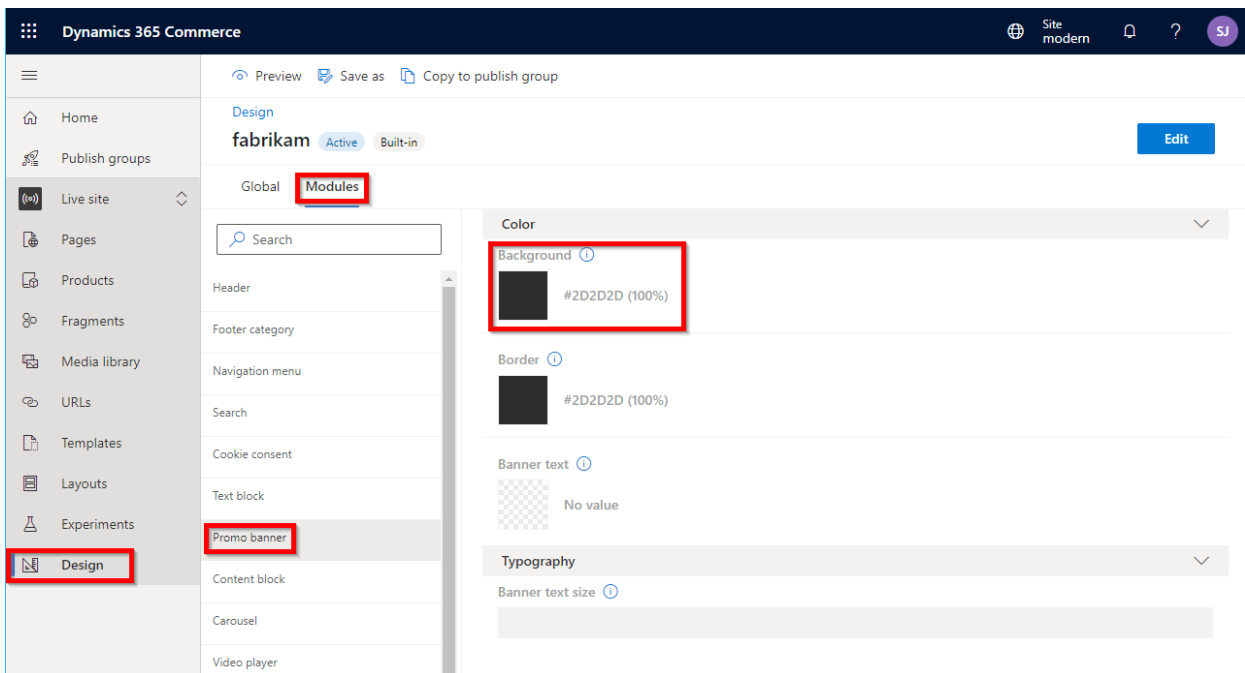
When you make CSS changes to a module, you should first determine whether the changes can be implemented by using a custom [style preset](#) in site builder. Each module has style options that can be changed in the **Style presets** section on the **Design** tab in site builder. This functionality allows for simple changes of some CSS in a module, such as the CSS for the background color, text color, and text size.

[NOTE] The style preset option in site builder is available in the 10.0.12 release of the module library and later. It's limited to a specific set of options for each module library module.

One limitation of style presets is that style changes to a module type are global. In other words, any module changes that you make will appear on all pages that use the edited module type. For example, if you change the background color of a module, the new color will be shown on all site pages that use that module. If you must change the style of a module that is used only on a single site page, consider using CSS overrides, or use module definition extensions to add new custom configurations to an existing module.

If you're building custom modules, you can expose style presets so that site authors can use them. For more information, see [Configure theme style presets](#).

The following illustration shows the style presets that are available for the promo banner module. If the background color is changed, the new color will be used for all instances of the module on the site after the change is published.



### CSS override file

A [CSS override file](#) can be uploaded by using Commerce site builder. This file contains additional CSS code to change any style on a page as required. When an Commerce page is rendered, the CSS for the theme module is applied first, and then the CSS overrides file is applied to override the CSS of the specific theme for that module. Because styles can easily be removed if they aren't required, this option is great for fast experimentation against the live site.

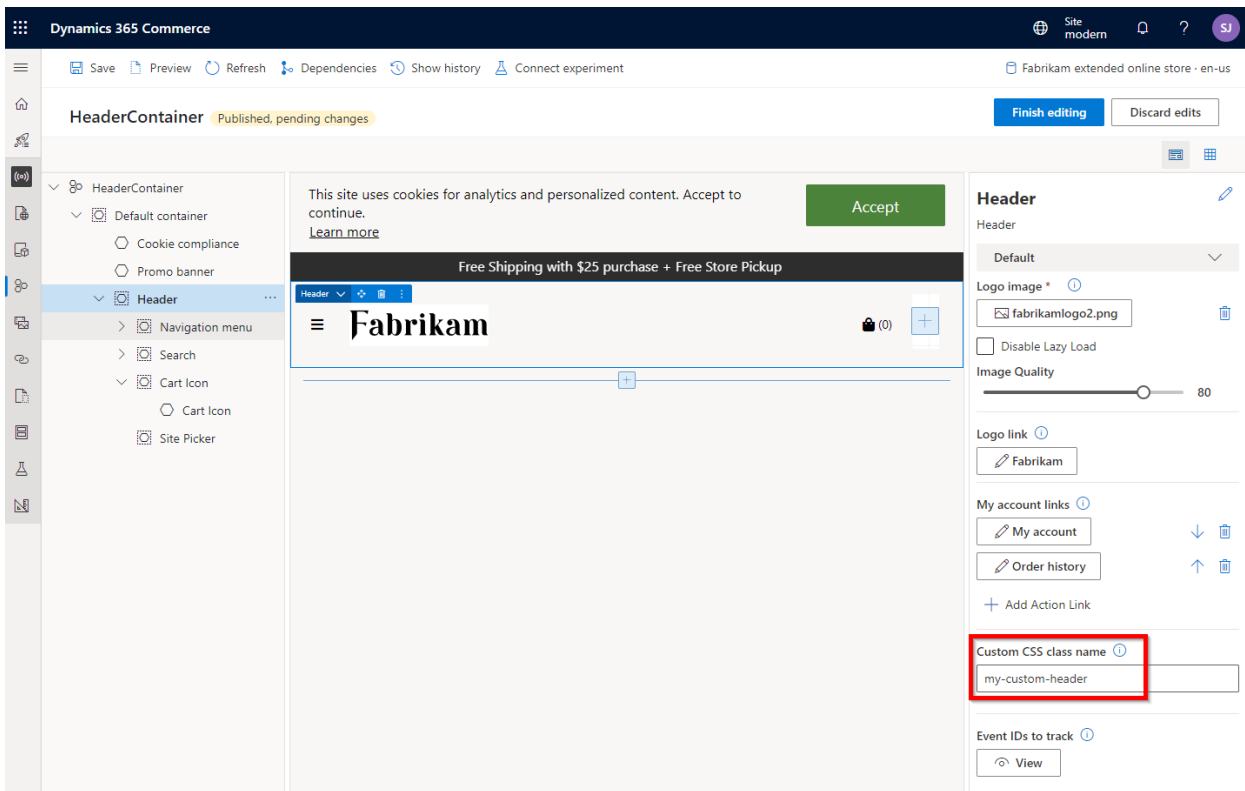
In general, if you can use tools such as a modern browser's HTML/CSS debugging tools to modify CSS as you want, you can then copy those changes into a single CSS override file. For example, if you want to hide the wishlist icon in the header module, you should use a CSS override to hide the element, because the module doesn't have a configuration setting that lets you show or hide the wishlist icon. By using F12 web browser tools, you can find the CSS class name to use, and then experiment with the formatting until you get the look that you want. The following example shows CSS code that hides the wishlist icon in a header module.

```
a.msc-wishlist-icon.ms-header__wishlist-desktop.msc-btn {
  visibility: hidden;
}
```

You can save this CSS code in a text file and upload it by using site builder. For more information, see [Work with CSS override files](#).

If you must target a change to one or a few instances of module, but not all of them, you can add a custom CSS class name to the instance of the module in site builder. After you publish the change, if you examine the HTML that is generated for the page instance, you will see that the new class name has been added to the HTML on the page.

The following illustration shows a header module in site builder after publication. The **Custom CSS class** name setting is highlighted.



## Theme changes

If your CSS changes are more complex, or if you want to make the changes permanent, the next level of customization involves using the online SDK to change the site theme. A site theme uses SCSS files to hold all the CSS code that modules require. The **fabrikam** starter theme is provided as a default theme that you can use on your online site. However, in general, we recommend that you create a new custom theme for your site. The process of creating a custom theme is easy, and you can even clone the default **fabrikam** theme and use the copy as the basis for your own theme.

If you're hosting multiple sites in a single instance of Dynamics 365 Commerce, you can create as many themes as you require. You can even create a base theme that other themes inherit from. In this way, you can minimize common changes across themes. For more information, see [Extend a theme from a base theme](#).

A theme not only lets you customize the module CSS, but it also supports extension of module views and definitions, and overrides of module library components.

### Create a new theme

By creating a new theme for your site from scratch, you will get an empty boilerplate theme that has no SCSS files for the module library module. For information about how to create a new theme, see [Create a new theme](#). If you want to create a new theme that uses module library styles as a starter, see the next section.

### Clone the fabrikam theme

To clone the **fabrikam** theme as a starter for your own theme, follow these steps.

1. Use the [clone](#) command-line interface (CLI) command to create a copy of the **fabrikam** theme. For example, use `yarn msdyn365 clone fabrikam MY_NEW_THEME`, where `MY_NEW_THEME` is the name of the new theme.

The **clone** command makes a copy of a module and puts it under the `/src/modules` directory. A theme is just a special module that is stored under the `/src/themes` directory.

2. Manually copy the new theme from the `src/modules` into the `src/themes` directory. If no `/themes` directory exists, you can manually create it first.
3. In the theme directory, find the file that is named `MY_NEW_THEME.definition.json`, where **MY\_NEW\_THEME** is the name that you provided in the **clone** command in step 1. This file lists the metadata that is used in site builder, such as the friendly name and description of the theme.

#### **Modify module CSS in a theme**

After you've created a custom theme, you can modify the SCSS as you require. The SCSS file for each module library module is stored under a theme's `/styles/04-modules` directory. There are many other SCSS files that you can also change as you require. For example, under the `//styles/00-settings` directory, you will find a `colors.scss` file that contains common color definitions.

After you create or modify a theme by using the online SDK, you can build a configuration package and upload it via Microsoft Dynamics Lifecycle Services (LCS). For more information, see [Package configurations and deploy them to an online environment](#). After a theme is deployed, it can be set for a site in site builder. For more information, see [Select a site theme](#).

#### **Extend a module's view in a theme**

You might have a requirement that involves more than just CSS changes. You might have to change the HTML in a module library module. Module library modules have been built in such a way that there is a separation between the React view file and the main component. Therefore, the view file can be overridden inside a theme.

For example, you want to reorder the HTML of the buy box module on the product details page (PDP), so that the product details appear above the price. You might be able to achieve this result through CSS changes alone. However, if you extend the module view, the changes to the page's HTML might also make the layout design more performant and responsive.

To extend a module view, you can override a module's view inside a theme. These changes are scoped to a site theme. Therefore, by changing a site's theme, you can change the whole look and feel of a module. One advantage of changing a module view instead of cloning a module is that updates to module library modules will still be applied. For example, if a module library module's business logic is improved and updated, the view override will continue to work on the code of the updated module library module.

You can find examples of module theme extensions in the starter **fabrikam** theme. For more information about how to extend modules, see [Extend a theme to add module extensions](#).

#### **Extend a module definition extension**

In addition to extending a module's view, you can extend a module library module definition to add or remove module configurations, slots, data actions, or resources. You can then access the new configurations from a module's view extension. Therefore, in general, when you add a theme definition extension, you also extend the module's view. For more information, see [Theme definitions extensions](#).

For example, if a module library module has a configuration for a title string, you can add a configuration for a subtitle string configuration to the module definition extension. This additional configuration can be set in site builder and rendered in the module's extended view.

#### **Override a module library component**

The module library contains a set of helper component files. These files contain APIs that some modules use to render HTML or to handle events that call server APIs. The components can be overridden and changed as required. One example of this type of component is the **price** component that renders the appropriate price. This component includes the markup for a price strikethrough, if a strikethrough is applicable. If you require a change to the rendering of the price, you can override the **price** component. For more information, see [Override a module library component](#).

# Data action overrides

[Data action overrides](#) allow for the customization of any core data action. Data actions generally call server-side APIs. They can also apply any additional business logic that is required on the return data before a module uses that data. When you override a data action, a copy is added to the `/src/action` directory. There, you can modify the code as you want. After a data action is overridden, all modules that called the original data action will now call the new one. Note that data action overrides are global. They aren't limited to a theme.

# Clone a module library module

If you can't achieve the desired changes to a module library module, but you want to use that module as a starter, you can [clone](#) it. You might have to use this approach in scenarios where you must change module business logic that isn't exposed in the module's view, and a view override won't do what you require. When you clone a module library module, you create a new module that has a copy of the code from the original module. You can then change the new module as required.

You should clone modules only as a last resort, because future updates to module library modules won't automatically be picked up by the cloned module. In this case, you must decide whether module library updates are required in the cloned module. If they are required, you must manually add them. To manually add updates after a module library module is updated, you can create a new clone and then compare the code with your cloned module.

For more information about how to get module library updates, see [SDK and module library updates](#).

# Create a custom module

If you require a feature that isn't available in the set of modules in the module library, you can [create a custom module](#). The process of creating a custom module is easy and can be done by using the [add-module](#) CLI command. As for module library modules, SCSS files can be added to the site theme, and different themes can optionally change the module's view through [module view extensions](#). Alternatively, the module definition can be extended as required.

# Additional resources

[SDK and module library kit updates](#)

[Set up a development environment](#)

[Module library overview](#)

[CLI command reference](#)

[Work with style presets](#)

[Work with CSS override files](#)

[Override a module library component in a theme](#)

[Theming overview](#)

[Create a new module](#)

[Data action overrides](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module definition file

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic covers the module definition file in Microsoft Dynamics 365 Commerce.

## Overview

A module definition file, `MODULE_NAME.definition.json`, is used to register a module and provide metadata to the Dynamics 365 Commerce site builder tool. This metadata includes the module name, description, categories, and configurations.

Here is an example of a module definition file.



```

{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "categories": ["marketing"],
  "tags": [""],
  "dataActions": {
    "products":{
      "path": "@msdyn365-commerce-modules/retail-actions/dist/lib/get-simple-products",
      "runOn": "server"
    }
  },
  "config": {
    "imageAlignment": {
      "friendlyName": "Image Alignment",
      "description": "Sets the desired alignment of the image, either left or right on the text.",
      "type": "string",
      "enum": {
        "left": "Left",
        "right": "Right"
      },
      "default": "left",
      "scope": "module",
      "group": "Layout Properties"
    },
    "productTitle": {
      "type": "string",
      "friendlyName": "Product Title",
      "description": "Product placement title",
      "required" : true
    },
    "productDetails": {
      "type": "richText",
      "friendlyName": "Product Details",
      "description": "Rich text representing the featured product details"
    },
    "productImage": {
      "type": "image",
      "friendlyName": "Product Image",
      "description": "Image representing the featured product"
    },
  },
  "resources": {
    "buyNowButtonText": {
      "value": "Buy Now!",
      "comment": "Text for the buy now button"
    }
  }
}

```

A module definition file also exposes configuration fields, so that a page author can configure module settings and resource definitions. In the example above, there is a configuration field for an image alignment setting (where the available values are **left** and **right**). Other examples could include a module title or heading, a rich text description, a "call to action" link, an image URL, or Commerce product data.

The page author can configure the settings of a module on a specific page without affecting the settings of that module on other pages. Module configurations can be implemented per module instance or globally across all instances of the module.

## Module definition schema

- **"\$type"** – The type of the module. A module can be either a content module (**contentModule**), a container

module (**containerModule**), a page module (**PageModule**), a script injector module (**scriptModule**), or a theme module (**themeModule**). Container and page modules also define "slots" that are used for layout regions. Script injector modules also define an "attributes" section that is used to specify where script can be injected.

- **"friendlyName"** – The friendly name of the module. This name is shown to page authors. The minimum length is three characters.
- **"name"** – The name of the module. This name must be unique across the application. It's used as the ID of the module and is referenced by the site builder tool. It should not be changed.
- **"description"** – The description of the module. The description provides a friendly string that is shown in the site builder tool when modules are added to pages.
- **"categories"** – The categories that the module can subscribe to. Container modules use the values that are specified here to allow or disallow some modules in specific slots.
- **"tags"** – The tags that are used to search for the module. All the categories are automatically added as tags.
- **"dataActions"** – The **dataActions** node is used to register the data actions that should be run for the module. By default, the data actions will run on the server side. However, they can also be configured to run on the client side.
- **"slots"** – Slots are defined only in container modules. They are exposed in the site builder tool. You can define allow and deny lists for a slot to allow or disallow specific modules from being accepted in that slot.
- **"attributes"** – Attributes are used to control script injector properties. For more information, see [Script injectors](#).
- **"config"** – The **config** section is used to add module configuration properties. For more information, see [Add module configuration fields](#).
- **"resources"** – This property is used to localize resources that are used in the module. For more information, see the [Module resource schema](#) section later in this topic.
- **"dependentSchemas"** – The **dependentSchema** section is used to show or hide configuration properties, based on the contextual values of other configuration properties. For more information, see [Configure module properties to display based on context](#).

## Register data actions to a module

If a module depends on data from a data action, the data action must be registered in the **dataActions** section of the module definition file.

The following example shows a module definition file that includes data action registrations.

```
// test-module.definition.json
{
  "$type": "contentModule",
  "friendlyName": "test-module",
  "name": "test-module",
  "description": "Module for testing observable data actions",
  "categories": ["test-module"],
  "tags": ["samples"],
  "dataActions": {
    "products":{
      "path": "@msdyn365-commerce-modules/retail-actions/dist/lib/get-simple-products",
      "runOn": "server"
    },
    "productWarranty":{
      "path": "../actions/getProductWarrantyInfo",
      "runOn": "server"
    }
  }
  ...
}
```

Each data action is declared with its name and the following properties:

- **"path"** – The path of the data action. The path can be a local path or the path of a core action included in the SDK (for example, "@msdyn365-commerce-modules/retail-actions/dist/lib/get-selected-variant").
- **"runOn"** – A setting that controls where the data action is run. Valid values are **server** or **client**. If no value is specified, the default value is **server**.

In the previous example, after the data action is registered, the module automatically runs it on the server. The module then binds the result to the **testResult** property that should be defined in the module's `data.ts` file.

```
// test-module.data.ts
import { AsyncResult, SimpleProduct } from '@msdyn365-commerce/retail-proxy';
import { IGetProductWarrantyInfoData } from '../../actions/getProductWarrantyInfo';

export interface IAsyncTestModuleData {
  products: AsyncResult<SimpleProduct>[];
  productWarrantyInfo: AsyncResult<IGetProductWarrantyInfoData>;
}
```

You can then access the results of the data action in your module.

## Module resource schema

- **"resources"** – This property is used to localize resources. When resources strings are defined, the localized strings are pulled from corresponding JavaScript Object Notation (JSON) files. These files are stored under the `/src/resources/modules/` directory. They include a `global.json` file for default locale values and any localized JSON files that are required, such as `fr-fr.json`.
- **"resourcekey"** – The name of the resource. Resource keys can then be accessed in code via the `this.props.resources.resourceKey` property.
- **"comment"** – A string that identifies the purpose of the string, to help with localization.
- **"value"** – The resource string data that will be used in the module.

## Additional resources

[Modules overview](#)

[Module React component file](#)

[Module view file](#)

[Module data file](#)

[Module mock file](#)

[Module test file](#)

[Module props.autogenerated.ts file](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module React component file

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides information about the module React component file in Microsoft Dynamics 365 Commerce.

## Overview

The module React component file is a TypeScript (.ts) file that contains business logic and controls a module's view. The React `render()` method is responsible for generating the module's HTML.

The default template of a module breaks up the React component and view into two files: the module React component (MODULE\_NAME.tsx) and the module view file (MODULE\_NAME.view.tsx). Generally, business logic is performed in the React component file and any data needed by the view is passed into the view with the `props` object. Separating into two files allows themes to provide a view extension file that will override the view for a module. This allows greater flexibility in changing a module's view based on the selected theme.

## Example

The following example shows the default React component file for a new module.

```
import * as React from 'react';

import { IProductFeatureData } from './product-feature.data';
import { IProductFeatureProps } from './product-feature.props.autogenerated';

export interface IProductFeatureViewProps extends IProductFeatureProps<IProductFeatureData> {

}

/**
 *
 * ProductFeature component
 * @extends {React.PureComponent<IProductFeatureProps<IProductFeatureData>>}
 */
class ProductFeature extends React.PureComponent<IProductFeatureProps<IProductFeatureData>> {
    public render(): JSX.Element | null {

        return this.props.renderView(this.props);
    }
}

export default ProductFeature;
```

Configuration properties that are defined in the module definition file (MODULE\_NAME.definition.json) and data properties file (MODULE\_NAME.data.ts) can be accessed in the module view file by using the `this.props.config.*` and `this.props.data.*` application programming interface (API) properties.

## Additional resources

[Modules overview](#)

[Module definition file](#)

[Module view file](#)

Module data file

Module mock file

Module test file

Module props.autogenerated.ts file

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module view file

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the module view file in Microsoft Dynamics 365 Commerce.

## Overview

A module view file is a TypeScript (.ts) file that controls a module's view. It's called from the module's React component. Additional module views can be provided in various themes to render a module differently, depending on the requirements of the theme.

## Example

The following example shows the default module view file for a new module.

```
import * as React from 'react';
import { IProductFeatureViewProps } from './product-feature';

export default (props: IProductFeatureViewProps) => {
  return (
    <div className='row'>
      <h2>Config Value: {props.config.showText}</h2>
      <h2>Resource Value: {props.resources.resourceKey}</h2>
    </div>
  );
};
```

## Additional resources

[Modules overview](#)

[Module definition file](#)

[Module React component file](#)

[Module data file](#)

[Module mock file](#)

[Module test file](#)

[Module props.autogenerated.ts file](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module data file

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the module data file in Microsoft Dynamics 365 Commerce.

## Overview

A module data file contains the typings for data actions that the module uses to fetch data. The naming convention for module data files is *MODULE\_NAME.data.ts*.

The software development kit (SDK) includes a set of core data actions that can get data from Dynamics 365 Commerce, ratings and reviews, or the recommendations service. You can find the list of data actions under the SDK's `\node_modules\@msdyn365-commerce-modules\retail-actions\dist\lib` directory.

## Example

The following example shows a sample data file for a new module.

```
import { SimpleProduct } from '@msdyn365-commerce/commerce-entities';
import { AsyncResult } from '@msdyn365-commerce/retail-proxy';
import { IGetProductReviewsData } from '../../actions/getProductReviews';

export interface IProductFeatureData {
  products: AsyncResult<SimpleProduct>[];
  productReviews: AsyncResult<IGetProductReviewsData>;
}
```

## Additional resources

[Modules overview](#)

[Module definition file](#)

[Module React component file](#)

[Module view file](#)

[Module mock file](#)

[Module test file](#)

[Module props.autogenerated.ts file](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module mock file

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the module mock file in Microsoft Dynamics 365 Commerce.

## Overview

Module mock files are used to configure mock data for modules. Therefore, they can help you do local testing. The default module mock file is located at `\src\MODULE_NAME\MOCKS\MODULE_NAME.json`.

### Example of a mock file

Here is an example of a mock file.

```
{
  "id": "R1Module1",
  "config": {
    "imageAlignment": "left",
    "productTitle": "Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses",
    "productDetails": "High-quality and pioneered with the perfect blend of timeless classic and modern technology with hint of old school glamor.",
    "productImage": {
      "src": "https://bit.ly/33cMGxr",
      "altText": "Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses"
    },
    "buttonText": "Buy Now",
    "productIds": "68719498121"
  },
  "data": {
    "actionResponse": {
      "text": "Sample Action Response"
    }
  },
  "typeName": "product-feature"
}
```

Several configuration properties are used to set values for configuration fields that are set in the module definition file. When modules are run on a production server, this data comes from the Microsoft Dynamics content management system (CMS) database, as configured by the page authors, instead of a mock file.

## Render a module by using mock file data

To render a module by using data from a mock file, run the **yarn start** command, and then open the following URL in a web browser: `https://localhost:4000/modules?type=MODULE_NAME`.

## Add mock files

In addition to the default mock file, you can add multiple other mock files. In this way, you can view and test various scenarios.

To add a mock file, create a new JavaScript Object Notation (JSON) file in the mocks folder, and give it a unique name (for example, `\src\MODULE_NAME\MOCKS\MODULE_NAME_TEST.json`).

To render a module by using a different mock file, run the **yarn start** command, and then open the following URL in a web browser: `https://localhost:4000/modules?type=MODULE_NAME:MODULE_NAME_TEST`.



# Additional resources

[Modules overview](#)

[Module definition file](#)

[Module React component file](#)

[Module view file](#)

[Module data file](#)

[Module test file](#)

[Module props.autogenerated.ts file](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module test file

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the module test file in Microsoft Dynamics 365 Commerce.

## Overview

The module test file is used for local unit testing. It contains the mock data that is required to run the tests.

## Example

The following example shows a default test file that is created for a new module.

```
import { buildMockModuleProps } from '@msdyn365-commerce/core';
/// <reference types="jest" />

// tslint:disable-next-line:no-unused-variable
import * as React from 'react';
import * as renderer from 'react-test-renderer';

import ProductFeature from '../productFeature';
import { IProductFeatureData } from '../productFeature.data';
import {
    IProductFeatureConfig,
    IProductFeatureProps
} from '../productFeature.props.autogenerated';

const mockData: IProductFeatureData = {
    actionResponse: {
        text: 'Sample Response Data'
    }
};

const mockConfig: IProductFeatureConfig = {
    showText: 'productFeature'
};

const mockActions = {};

describe('ProductFeature', () => {
    let moduleProps: IProductFeatureProps<IProductFeatureData>;
    beforeAll(() => {
        moduleProps = buildMockModuleProps(mockData, mockActions, mockConfig) as
IProductFeatureProps<IProductFeatureData>;
    });
    it('renders correctly', () => {
        const component: renderer.ReactTestRenderer = renderer.create(
            <ProductFeature {...moduleProps} />
        );
        const tree: renderer.ReactTestRendererJSON | null = component.toJSON();
        expect(tree).toMatchSnapshot();
    });
});
```

Note that the mock data fields are set inside this file.

## Additional resources

[Modules overview](#)

[Module definition file](#)

[Module React component file](#)

[Module view file](#)

[Module data file](#)

[Module mock file](#)

[Module props.autogenerated.ts file](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Module props.autogenerated.ts file

2/18/2021 • 2 minutes to read • [Edit Online](#)

## Overview

The props.autogenerated.ts file is a TypeScript file that is automatically generated for a module. It shouldn't be manually changed. If new configuration values are added to the module definition file and module data files, this file is automatically built so that it includes the appropriate interfaces.

## Example

The following example shows the props.autogenerated.ts file for a module.

```
/**
 * Copyright (c) 2018 Microsoft Corporation
 * IProductFeature contentModule Interface Properties
 * THIS FILE IS AUTO-GENERATED - MANUAL MODIFICATIONS WILL BE LOST
 */

import * as Msdyn365 from '@msdyn365-commerce/core';

export const enum imageAlignment {
    left = 'left',
    right = 'right'
}

export interface IProductFeatureConfig extends Msdyn365.IModuleConfig {
    imageAlignment?: imageAlignment;
    productTitle?: string;
    productDetails?: Msdyn365.RichText;
    productImage?: Msdyn365.IImageData;
    buttonText?: string;
    productIds?: string;
}

export interface IProductFeatureResources {
    resourceKey: string;
}

export interface IProductFeatureProps<T> extends Msdyn365.IModule<T> {
    resources: IProductFeatureResources;
    config: IProductFeatureConfig;
}
```

## Additional resources

[Modules overview](#)

[Module definition file](#)

[Module React component file](#)

[Module view file](#)

[Module data file](#)

[Module mock file](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a new module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create a new module in Dynamics 365 Commerce.

## Overview

To create a new module in Commerce, the online software development kit (SDK) provides the **add-module** command-line interface (CLI) command. When you run the command as in the following example, you replace **MODULE\_NAME** with the name that you want to give to the new module.

```
yarn msdyn365 add-module MODULE_NAME
```

## Example

The following example shows how to create a module that is named **product-feature**.

```
yarn msdyn365 add-module product-feature
```

It can take 20 to 30 seconds to create a module and generate all the template files for it. After the command has finished running, you can find the new module in the `\src\modules\` directory.

## Preview a module

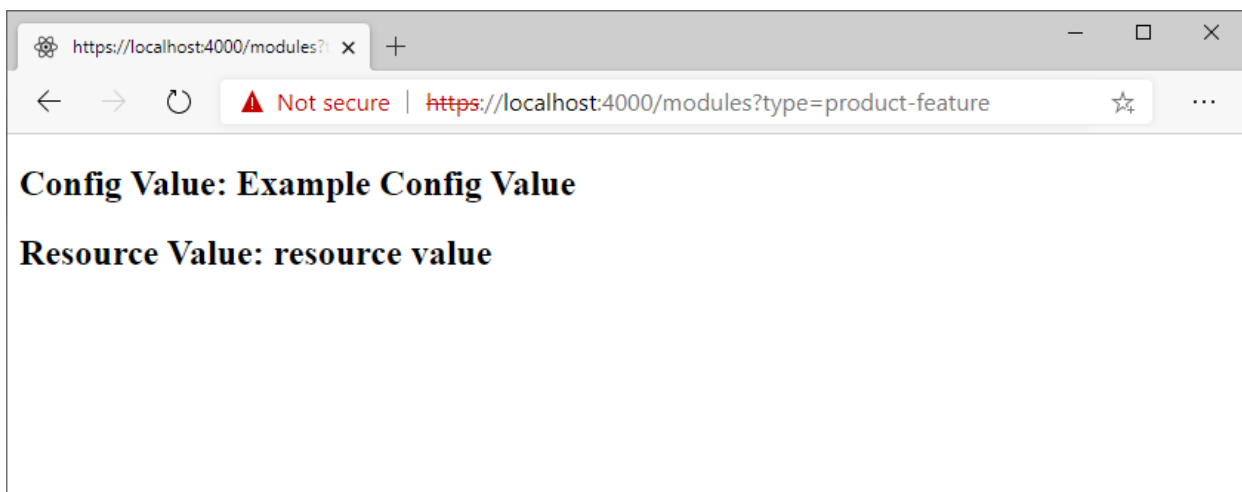
To preview the new module in a local web browser, follow these steps.

1. At a command prompt, go to your root SDK folder, and run the **yarn start** command. Here is an example.

```
c:\repos\Msdy365.Commerce.Online>yarn start
```

2. In a web browser, open the following URL to view the module:

`https://localhost:4000/modules?type=product-feature`. Notice the module name in the **type=MODULE\_NAME** query string parameter.



## Module naming conventions

Module names are case-insensitive. We recommend that you use whole words for module names whenever you can.

## Deferred module rendering

By default all modules are rendered server-side, but deferred loading of some modules may be needed to improve page load performance. For more information, see [Page load data actions](#).

Any references to window or document objects that are only available in the context of a browser should be handled appropriately during server-side rendering. This will avoid unexpected rendering behavior such as page flicker and Document Object Model (DOM) mismatch issues. The following SDK utility function can be used for this purpose.

```
import MsDyn365 from '@msdyn365-commerce/core';

if (MsDyn365.isBrowser) {
  return new URL(window.location.href);
}
```

## Module error handling

If a module encounters an error during server-side rendering, the failed module is then wrapped into an **ErrorModule** component to prevent any module-level render error from breaking the page. For example, a module using window or document objects during a server-side render would fail because these objects are non-existent on the server-side. In this case, the module would then be wrapped in an error component. The module would then attempt to render again on the client. In development mode, to determine if a module failed on server side use a `?debug=true` query string parameter.

## Additional resources

[Page load data actions](#)

[CLI command reference](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

[Test modules by using module mocks](#)

[Test modules by using page mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Create a page container module](#)

[Localize a module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Clone a module library module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to clone a module library module.

## Overview

The Microsoft Dynamics 365 Commerce online software development kit (SDK) includes a set of module library modules that can be used on an e-Commerce site. Although these modules can't be modified directly, they can be cloned into new modules and then updated.

You can also create module extension views. In this way, you can provide alternative layout views without having to clone a module. We recommend that you avoid cloning if you can, because clones will be copies of module library modules and won't receive any automatic service updates that the module library modules get. For more information, see [Theming overview](#).

## Clone and update a module

To clone a module and then update it, use the `clone` command-line interface (CLI) command. When you run the below command, you replace `SDK_MODULE_NAME` with the name of the module that you want to modify and `NEW_MODULE_NAME` with the name of the new module.

```
yarn msdyn365 clone SDK_MODULE_NAME NEW_MODULE_NAME
```

This command adds the source code for the module to the `/src/modules/` directory and pulls in any required dependencies for the module.

## Example

The following example shows how to clone the hero SDK module so that you can update it.

```
yarn msdyn365 clone content-block super-content-block
```

It can take up to a minute to clone a module. After the command has finished running, you can find the new module in the `\src\modules\` directory.

### NOTE

Module dependencies aren't automatically pulled down when you clone a module. Before you build the module, you must run Yarn and fix any missing dependencies. If any errors occur when you run the `yarn start` command, you might also have to fix some references inside the module source code.

## Preview a module

To preview the new module in a local web browser, follow these steps.

1. At a command prompt, go to your root SDK folder, and run the `yarn start` command. Here is an example.

```
c:\repos\Msdy365.Commerce.Online>yarn start
```



2. In a web browser, open the following URL to view the module:

`https://localhost:4000/modules?type=super-content-block`. Notice the module name in the `type=MODULE_NAME` query string parameter.

You can now update the module code as needed.

## Additional resources

[Create a new module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

[Test modules by using module mocks](#)

[Test modules by using page mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Create a page container module](#)

[Localize a module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add module configuration fields

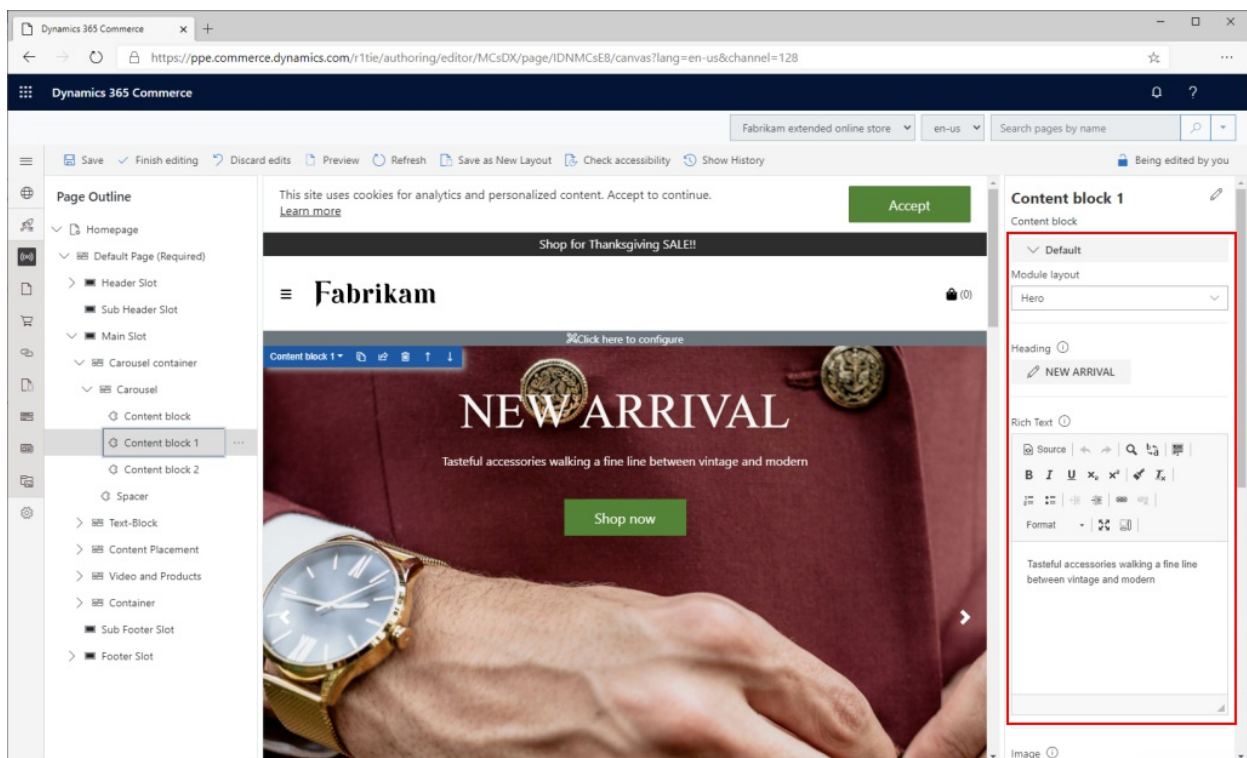
2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes how to add module configuration fields in Microsoft Dynamics 365 Commerce.

## Overview

Configuration fields can be added to a module to expose them to page authors and give them control of various module features. Examples of these features include different views, alignment properties, Boolean switches to turn features on or off, module titles or headings, rich text descriptions, call-to-action links, image URLs, and Commerce product data.

The following illustration shows how these fields appear in the page authoring tools.



## Add new module configuration fields

To add configuration fields, you add an entry in the **config** section of the module definition file, **MODULE\_NAME.definition.json**.

### Example

In the following example of a module definition file, an **imageAlignment** configuration field has been added so that page authors can configure the alignment of an image inside a module. There are two enumeration (enum) options: **"Left"** (the default option) and **"Right"**.

```

{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "categories": [
    "storytelling"
  ],
  "tags": [
    ""
  ],
  "dataActions": {
    "products": {
      "path": "@msdyn365-commerce-modules/retail-actions/dist/lib/get-simple-products",
      "runOn": "server"
    }
  },
  "config": {
    "imageAlignment": {
      "friendlyName": "Image Alignment",
      "description": "Sets the desired alignment of the image, either left or right on the text.",
      "type": "string",
      "enum": {
        "left": "Left",
        "right": "Right"
      },
      "default": "left",
      "scope": "module",
      "group": "Layout Properties"
    }
  }
}

```

## Module configuration schema

The `config` section of the module definition file contains a list of all the module's exposed configuration fields that will be used in the authoring tools.

- **configuration name** – The local name that is used to access the configuration values from your React source code. This name is case-insensitive.
- **"friendlyName"** – The friendly name that is shown as the configuration name in the authoring tools.
- **"description"** – The description that is shown as the configuration description in the authoring tools.
- **"type"** – The type of the configuration. The possible values are **"string"**, **"bool"**, **"number"**, **"integer"**, **"richText"**, **"image"**, **"imageSettings"**, **"css"**, **"video"**, and **"array"**.
- **"enum"** – For an enumerator type, the value must be set to **"string"**.
- **"default"** – The default value that is set if no value is set in the authoring tools.
- **"scope"** – This field is used to scope the configuration to either a specific module instance or all modules on the site. Possible values are **"module"** and **"site"**. If the value is set to **"site"**, the module configuration doesn't appear on a page and can't be configured there. It appears and can be configured only at the site level. In this way, the value can be set one time for the entire site. If you don't set this field, the default value is **"module"**.
- **"group"** – Groups are used to organize the configurations into organized groups in the authoring tools.
- **"required"** – A Boolean flag that specifies whether a property must be set on the module. If the value is set to **true**, the authoring tools will show an error if the required property isn't set, and an error will be shown when the module is rendered.
- **"resources"** – This field is used for localization resources.
- **"definitions"** – This field can contain complex config type definitions, which can be referenced in the config

sections as extended types.

The following example shows how the various supported data types are used.

```
{
  "$type": "contentModule",
  "friendlyName": "Sample Config",
  "name": "sample-config",
  "description": "Sample Config",
  "categories": ["sample-config"],
  "tags": ["samples"],
  "dataActions": {},
  "config": {
    "title": {
      "type": "string",
      "friendlyName": "Title",
      "description": "Example config value",
      "default": "",
      "scope": "module"
    },
    "subTitle": {
      "type": "richText",
      "friendlyName": "SubTitle",
      "description": "Sub title rich text field"
    },
    "bgImage": {
      "type": "image",
      "friendlyName": "Background image",
      "description": "Background image"
    },
    "images": {
      "type": "array",
      "friendlyName": "Images",
      "description": "Image array",
      "items": {
        "type": "image"
      }
    },
    "backgroundImageSettings": {
      "type": "imageSettings",
      "friendlyName": "Background image settings",
      "description": "Image settings for background image settings"
    },
    "ambientVideo": {
      "type": "video",
      "friendlyName": "Ambient video",
      "description": "Ambient video",
    },
    "headingArray": {
      "type": "array",
      "friendlyName": "Headings",
      "description": "Heading array",
      "items": {
        "$ref": "#/definitions/heading"
      }
    },
    "heading": {
      "$ref": "#/definitions/heading"
    },
    "heading2": {
      "type": "object",
      "friendlyName": "Heading2",
      "description": "Heading2 property with its own enum",
      "properties": {
        "style": {
          "type": "string",
          "enum": {
            "bold": "Bold"
          }
        }
      }
    }
  }
}
```

```
        "underline": "Underline",
        "italics": "Italics",
        "strong": "Strong",
        "emphasized": "Emphasized",
        "none": "None"
    },
    "friendlyName": "Style",
    "description": "Heading style"
}
}
},
"textPlacement":{
    "friendlyName": "Text placement",
    "description": "Placement of the text",
    "type": "object",
    "enum":{
        "left": "Left",
        "right": "Right",
        "center": "Center"
    },
    "default": "left"
}
},
"definitions": {
    "heading": {
        "type": "object",
        "friendlyName": "Heading",
        "description": "Heading property",
        "properties": {
            "text": {
                "type": "string",
                "friendlyName": "Text",
                "description": "Heading Text"
            },
            "style": {
                "type": "string",
                "enum": {
                    "bold": "Bold",
                    "underline": "Underline",
                    "none": "None"
                },
                "friendlyName": "Style",
                "description": "Heading style"
            },
            "showImage":{
                "type":"boolean",
                "friendlyName": "Show image?",
                "description": "Should Show Image"
            },
            "bgImage": {
                "type": "image",
                "friendlyName": "Background image",
                "description": "Background image"
            },
            "imageArray":{
                "type": "array",
                "friendlyName": "Images",
                "description": "Image Array",
                "items": {
                    "type": "image"
                }
            }
        }
    }
}
}
}
```

## css configuration type

The configuration type of module configuration properties can also be declared as **"type": "css"**. Module configuration properties declared as the **css** type must specify a set of string enums of the classes that can be applied to a module. Only one of the enum options can be selected for a given module configuration property. When an enum option is selected, the selected class is appended to the list of classes that is passed down to the module in the format *propertyName\_\_propertyValue* via the **this.props.config.className** property. Note that **css** configuration types can't be accessed directly from the **this.props.config** property, because they are merged into the **this.props.config.className** property.

## className property

Every content module includes a built-in configuration field that is named **className**. This configuration field can be accessed inside the module's view via the **this.props.config.className** property. This configuration field will appear in the site authoring tools. Therefore, page authors can add a string of space-separated Cascading Style Sheets (CSS) class names that should be appended to the module root class.

## \_\_cssClassName\_\_ property

**\_\_cssClassName\_\_** is another special property that is declared inside a module definition file. It provides a way for the module creator to specify a non-authorable, non-editable, read-only **className** property that will always be applied to the module. Therefore, its **editable** property must be set to **false**, and a default value must be specified.

In the following example, the module creator has given this field a default value of **hero**. Therefore, every instance of this module will always have the **hero** class as part of the **this.props.config.className** property.

```
...
  "__cssClassName__": {
    "default": "hero",
    "editable": false,
    "friendlyName": "Readonly CSS Class Name(s)",
    "description": "Provides a way to set constant, unchangeable default css class(es) to apply to your
module.",
    "type": "string"
  },
  ...
```

## Use mock data in configuration fields for local testing

The following example shows how to set a mock value for a new configuration field in the **mocks/MODULE\_NAME.json** file. Mock data is useful when a module is rendered in a local development environment.

```
{
  "id": "R1Module1",
  "config": {
    "imageAlignment": "left",
    "productTitle": "Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses",
    "productDetails": "High-quality and pioneered with the perfect blend of timeless classic and modern
technology with hint of old school glamor.",
    "productImage": {
      "src": "https://bit.ly/33cMGxr",
      "altText": "Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses"
    },
    "buttonText": "Buy Now",
    "productIds": "68719498121"
  },
  "data": {
    "actionResponse": {
      "text": "Sample Action Response"
    }
  },
  "typeName": "product-feature"
}
```

### Access configuration fields in the module React component

To access configuration fields in the React component, use the `props.config` application programming interface (API).

The following example creates a `props` property that has configuration values that will be sent to the module view file to render the appropriate HTML.

```

import * as React from 'react';

import { IProductFeatureData } from './product-feature.data';
import { imageAlignment, IProductFeatureProps } from './product-feature.props.autogenerated';

export interface IProductFeatureViewProps extends IProductFeatureProps<IProductFeatureData> {
  productName: string;
  productInfo: string;
  productImageUrl: string;
  productPrice: string;
  buttonInfo: string;
  alignment: imageAlignment;
}

/**
 *
 * ProductFeature component
 * @extends {React.PureComponent<IProductFeatureProps<IProductFeatureData>>}
 */
class ProductFeature extends React.PureComponent<IProductFeatureProps<IProductFeatureData>> {
  public render(): JSX.Element | null {
    const { config } = this.props;

    // set default product info values
    const ProductName = config.productTitle ? config.productTitle : 'No product name defined';
    const ProductInfo = config.productDetails ? config.productDetails.toString() : 'No product details
defined';
    const ProductImageUrl = config.productImage ? config.productImage.src : '';
    const ButtonInfo = config.buttonText ? config.buttonText : 'No button text defined';
    const ProductPrice = '129';

    const ProductFeatureViewProps = {
      ...this.props,
      productName: ProductName,
      productInfo: ProductInfo,
      productImageUrl: ProductImageUrl,
      productPrice: ProductPrice,
      buttonInfo: ButtonInfo,
      alignment: config.imageAlignment
    };

    return this.props.renderView(ProductFeatureViewProps);
  }
}

```

The following example shows the corresponding module view file that handles the HTML layout.



```

import * as React from 'react';
import { IProductFeatureViewProps } from './product-feature';
import { imageAlignment } from './product-feature.props.autogenerated';

const _renderImage=(productImageUrl: string, productName: string): JSX.Element => {
    return <img src={productImageUrl} alt={productName} className='img-fluid p-3' />;
};

const _renderInfo=(productName: string, productInfo: string, productPrice: string, buttonInfo: string):
JSX.Element => {
    return (
        <div className='container'>
            <h2>{productName}</h2>
            <p>{productInfo}</p>
            <p>{productPrice}</p>
            <button type='button' className='btn btn-primary'>{buttonInfo}</button>
        </div>
    );
};

export default (props: IProductFeatureViewProps) => {
    const { productName, productInfo, productImageUrl, productPrice, buttonInfo, alignment } = props;

    let left;
    let right;

    if (alignment === imageAlignment.left) {
        left = _renderImage(productImageUrl, productName);
        right = _renderInfo(productName, productInfo, productPrice, buttonInfo);
    } else {
        right = _renderImage(productImageUrl, productName);
        left = _renderInfo(productName, productInfo, productPrice, buttonInfo);
    }

    return (
        <div className='row align-items-center'>
            <div className='col-sm-6'>
                {left}
            </div>
            <div className='col-sm-6'>
                {right}
            </div>
        </div>
    );
};

```

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Preview and debug a module](#)

[Test modules by using module mocks](#)

[Test modules by using page mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Create a page container module](#)

[Localize a module](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Preview and debug a module

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to test a module by previewing and debugging it in a web browser.

## Overview

During development of a module, you can preview and debug the module in your local web browser.

## Preview a module

To preview a module in a local web browser, follow these steps.

1. At a command prompt, go to your root SDK folder, and run the **yarn start** command. Here is an example.

```
c:\repos\Msdyn365.Commerce.Online>yarn start
```

2. In a web browser, open the following URL to view the module:

```
https://localhost:4000/modules?type=product-feature
```

Notice the module name in the **type=MODULE\_NAME** query string parameter. This parameter is shorthand for **type=:MODULE\_NAME:MOCK\_FILE\_NAME**, where the default mock file that matches the name of the module is loaded. Therefore, the preceding URL is equivalent to

```
https://localhost:4000/modules?type=product-feature:product-feature
```

By adding **&debug=true**, you can get more verbose debug information in the Yarn output window, and the browser will show the details of any error details that occur.

```
https://localhost:4000/modules?type=product-feature&debug=true
```

## Debug a module

A debugger can be attached to both the client and the server. Sometimes, you might have to use both methods.

### Debug the client (browser)

You can debug in the browser in two ways:

- Add a **debugger**; statement in your code, and then open the developer tools window in your browser. (In modern browsers, you can typically open the developer tools by pressing the **F12** key.) Your breakpoint should be hit, and you can then use the debugging capabilities of your web browser (for example, watch, locals, or expression evaluation).
- Open the developer tools window, and open the file where you want to set a breakpoint. Depending on your browser, you might be able to use the **Ctrl+P** keyboard shortcut to go to the file and show full source maps. After you set a breakpoint at the desired line number and refresh the page, execution should pause at that breakpoint.

### Debug the server (Node.js)

#### Open debugging tools

Node.js includes dedicated tools that let you attach a debugger to a running application.

To open debugging tools in Google Chrome, go to **chrome://inspect**.

From the view that is opened, you can attach a debugger to the running Node.js applications that expect a

debugger to connect. Keep this view open while you attach the debugger.

### Attach the debugger

To inform Node.js that you want to attach a debugger, start your application by using the following command.

```
yarn start --inspect-brk
```

As you can see, the **inspect-brk** argument is added to the regular **start** command. After the application is compiled, a message that resembles the following message should appear in your console: "Waiting for debugger..."

The previously opened window should gain focus. The application isn't actually run until you select **Play** in the debugging window. After you allow the application to continue, you can set breakpoints by using the methods that are described in the [Debug the client \(browser\)](#) section earlier in this topic.

### Additional reading

- [Documentation for debugging Node.js in Microsoft Visual Studio Code \(VS Code\)](#)

### Troubleshooting

- If the debugger doesn't stop at the breakpoint that you set, it's usually a good idea to restart the server. In this way, you help guarantee that you have a clean build, because the Hot Module Replacement (HMR) functionality doesn't always achieve the best results.
- Sometimes, the transpiled code (TypeScript to JavaScript) makes debugging more challenging, and you must review the raw JavaScript to understand the code that is running. In these cases, you should turn off JavaScript source maps and follow standard instructions for adding breakpoints in JavaScript code.
- If you're debugging a node, make sure that the debugging port is configured for auto-connection by using the dedicated DevTools for Node.js.

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Test modules by using module mocks](#)

[Test modules by using page mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Create a page container module](#)

[Localize a module](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Test modules with module mocks

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to test modules by using module mocks.

## Overview

Modules typically get their data from data actions and configuration fields, and the HTML that they present is then based on that data. Because modules might not have direct access to the data (such as the Commerce Scale Unit data) when they run in a local development environment, module mock data files can be used for testing. These files are used to set data that can be used to render a module when it runs locally through a web browser.

## Module mock files

Module mock data files are stored under the `/src/<MODULE_NAME>/mocks` directory. The default mock file uses the `<MODULE_NAME>.json` file, but you can add other mock files. To specify different module mock data files when you test in a web browser, append a colon (`:`) and a comma-separated list of mock file names (but without the `.json` file name extension) to the module name.

For example, for a module mock data file that is named `mockTest1.json` and added to the module `mocks` directory, use the URL `http://localhost:4000/modules?type=product-feature:mockTest1`. In this example, the mock name is `product-feature`, which is used in the URL.

## Mock configuration and data fields

To mock the configuration fields of a module, put the configuration names under the `config` section in the mock `.json` file. To mock the data fields of a module, put the data names under the `data` section in the mock `.json` file.

The following example shows a configured module mock `.json` file.

```
{
  "id": "R1Module1",
  "config": {
    "imageAlignment": "left",
  },
  "data": {
    "actionResponse": {
      "text": "Sample Action Response"
    }
  },
  "typeName": "product-feature"
}
```

## Example

The following example shows a module config file that includes various configuration definitions, and the corresponding mock file.

Sample module definition file:

```

{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "categories": [
    "storytelling"
  ],
  "tags": [
    ""
  ],
  "dataActions": {
  },
  "config": {
    "imageAlignment": {
      "friendlyName": "Image Alignment",
      "description": "Sets the desired alignment of the image, either left or right on the text.",
      "type": "string",
      "enum": {
        "left": "Left",
        "right": "Right"
      },
      "default": "left",
      "scope": "module",
      "group": "Layout Properties"
    },
    "productTitle": {
      "type": "string",
      "friendlyName": "Product Title",
      "description": "Product placement title"
    },
    "productDetails": {
      "type": "richText",
      "friendlyName": "Product Details",
      "description": "Rich text representing the featured product details"
    },
    "productImage": {
      "type": "image",
      "friendlyName": "Product Image",
      "description": "Image representing the featured product"
    },
    "buttonText": {
      "type": "string",
      "friendlyName": "Button Text",
      "description": "Text to show on the call to action button"
    },
    "productIds": {
      "friendlyName": "Product ID",
      "description": "Provide a Product Id that the module will display",
      "type": "string",
      "scope": "module",
      "group": "Content Properties"
    }
  },
  "resources": {
    "resourceKey": {
      "comment": "resource description",
      "value": "resource value"
    }
  }
}

```

Sample mock file:

```
{
  "id": "R1Module1",
  "config": {
    "imageAlignment": "left",
    "productTitle": "Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses",
    "productDetails": "High-quality and pioneered with the perfect blend of timeless classic and modern
technology with hint of old school glamor.",
    "productImage": {
      "src": "https://bit.ly/33cMGxr",
      "altText": "Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses"
    },
    "buttonText": "Buy Now",
    "productIds": "68719498121"
  },
  "data": {
    "actionResponse": {
      "text": "Sample Action Response"
    }
  },
  "typeName": "product-feature"
}
```

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

[Test modules by using page mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Create a page container module](#)

[Localize a module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Test modules by using page mocks

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to test modules by using page mocks.

## Overview

Some modules are built to interact with other modules. You can use page mocks to test those modules together in a local development environment.

Page mock files are stored under the `/src/pageMocks` directory. They can be loaded by using the URL `https://localhost:4000/page?mock=PAGE MOCK`, where **PAGE MOCK** is the file name of the mock file, but without the `.json` file name extension.

## Create a new page mock

To create a new page mock, create a blank `.json` file under the `/src/pageMocks` directory, such as `/src/pageMocks/campaign-page.json`.

## Example

The following example shows a page mock that adds two instances of the same module to a page but uses different mock data for each instance.



```

{
  "exception": null,
  "pageRoot": {
    "id": "core-root_0",
    "typeName": "core-root",
    "modules": {
      "body": [
        {
          "id": "default-page_0",
          "typeName": "default-page",
          "modules": {
            "primary": [
              {
                "id": "ProductFeature_0",
                "typeName": "product-feature",
                "config": {
                  "imageAlignment": "left",
                  "buttonText": "Buy Now",
                  "productIds": "68719490621"
                }
              },
              {
                "id": "ProductFeature_1",
                "typeName": "product-feature",
                "config": {
                  "imageAlignment": "right",
                  "productIds": "68719498121",
                  "buttonText": "Buy Now"
                }
              }
            ]
          }
        }
      ]
    }
  },
  "renderingContext": {
    "gridSettings": {
      "xs": {
        "w": 767
      },
      "sm": {
        "w": 991
      },
      "md": {
        "w": 1199
      },
      "lg": {
        "w": 1599
      },
      "xl": {
        "w": 1600
      }
    },
    "staticContext": {
      "staticCdnUrl": "/_scnr/"
    },
    "locale": "en-us"
  },
  "statusCode": 200
}

```

Every page defines a root (**core-root** in the example above) that controls the page HTML structure with slots for "HTML Head", "Body Begin", "Body" and "Body End". The **body** then must have a page container module. In this example, the **default-page** page container is used.

The **modules** section lists the modules that are included inside the page arranged by named slots. The **default-page** page container has a slot that is named **primary**. This container is responsible for laying out the modules that are included inside it. In this example, the **productFeature** module is rendered two times in a row with the mock data defined in the **config** section for each.

The preceding example can be accessed by using the following URL:

```
https://localhost:4000/page?mock=campaign-page .
```

## Rendering context mocks

The **renderingContext** node provides additional mock data that modules can access via the **this.props.context.request** object. This mock data can include the bootstrap grid breakpoints, the locale, and the user context. For more information, see [Request properties object](#).

### Simulate the signed-in state

Some modules might have logic that must check the signed-in state of the user before they take the appropriate action. Typically, a module gets the state from the **this.props.context.request.user** object. Because business-to-consumer (B2C) sign-in isn't supported in a development environment, the user object can be mocked by using the **user** node, as shown in the following example.

```
{
  "exception": null,
  "pageRoot": {
    "id": "core-root_0",
    "typeName": "core-root",
    "modules": {
      "body": [
        {
          "id": "default-page_0",
          "typeName": "default-page",
          "modules": {
            "primary": [
              {
                "id": "ProductFeature__0",
                "typeName": "product-feature",
                "config": {
                  "imageAlignment": "left",
                  "buttonText": "Buy Now",
                  "productIds": "68719490621"
                }
              }
            ]
          }
        }
      ]
    }
  },
  "renderingContext": {
    "gridSettings": {
      "xs": {
        "w": 767
      },
      "sm": {
        "w": 991
      },
      "md": {
        "w": 1199
      },
      "lg": {
        "w": 1599
      },
      "xl": {
        "w": 1600
      }
    }
  }
}
```

```

    },
    "staticContext": {
      "staticCdnUrl": "/_scnr/"
    },
    "user": {
      "token": "<TOKEN>",
      "isAuthenticated": true,
      "signInUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signin",
      "signOutUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signout",
      "signUpUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signup",
      "editProfileUrl": "https://dev.fabrikam.com/fedev/_msdyn365/editprofile",
      "signInName": "<User Name>",
      "firstName": "<User First Name>",
      "lastName": "<User Last Name>",
      "tenantId": "",
      "customerAccountNumber": "<User Account Number(HQ)>",
      "name": "<User Name>",
      "emailAddress": "<User Email Address>"
    },
    "locale": "en-us"
  },
  "statusCode": 200
}

```

If you must simulate real data or the token that is returned from Commerce Server after a user signs in, sign in to your production e-Commerce site, and use the F12 browser tools to copy the data. The user information is available in the `___initialData___requestContext.user` global JavaScript variable. While the F12 browser tools are open and a user is signed in, open a console window, and enter `___initialData___requestContext.user` to see the object. You can then update the `userContext` properties in the preceding example as required. Those properties include `token`, `signInName`, `firstName`, `lastName`, `customerAccountNumber`, `name`, and `emailAddress`.

## Create a dynamic page mock from a production e-Commerce page

You can create dynamic page mocks that mimic live e-Commerce site pages and can be used locally to test module interaction. For example, these page mocks can mock the signed-in state and other page contextual properties as required. Therefore, they can be helpful when you locally test pages such as account pages and wishlist pages, or interactions such as order checkout flow.

### Save a live e-Commerce page's raw JSON structure

The raw JavaScript Object Notation (JSON) structure of any live e-Commerce page can be captured and saved so that it can be used as a page mock. Open the e-Commerce site page that you want to capture, and sign in if the signed-in state is desired. Next, append the query string parameter `?item=nodeserviceproxy:true` to the page URL, and then reload the page to obtain the JSON of the raw page context.

#### NOTE

For this operation to work, you must have secure Azure Active Directory (Azure AD) access to your production site, and you might be prompted to sign in if you aren't already signed. Use the same Azure AD account that you use to sign in to Commerce site builder.

Next, in your development environment, create a new page mock JSON file under the `src/pageMocks` directory. Paste in the JSON file that you obtained from the capture and save operation.

The structure of the JSON file should have sections that resemble the sections in the following example. These sections should include the `pageRoot` section that defines the set of modules for the page and the `renderingContext` section that includes the `user` context for signed-in user information.

```
{
  "exception": null,
  "pageRoot": {
    "id": "core-root_0",
    "typeName": "core-root",
    "modules": {
    }
  },
  "renderingContext": {
    ...
    "user": {
      ...
    }
  },
  "statusCode": 200
  ...
}
```

To test the page mock locally, start your Node server by using **yarn start** command. Access the page mock by using the URL format `https://localhost:4000/page?mock=PAGE MOCK`, where **PAGE MOCK** is the file name of the mock file without the `.json` file name extension.

You can now modify the page mock as desired.

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

[Test modules by using module mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Create a page container module](#)

[Localize a module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Container modules

2/18/2021 • 2 minutes to read • [Edit Online](#)

Container modules help you control the layout when you build complex modules or pages out of small component modules. For example, a container module might be a header, footer, or buy box module that has nested modules inside it.

Container modules can define "slots" that are exposed to template authors. You can think of slots as regions inside the container module. Page authors can configure which modules go into each slot. The code of the container module is responsible for the HTML layout of the slots.

Configuration settings can also be exposed to page authors. In this way, page authors can do additional configuration of the layout of container modules. The container module is responsible for building a responsive design, to help guarantee that the module will look good at any size, regardless of whether it's viewed on a mobile device screen or in a full web browser.

Container modules are created just like regular modules. However, in the `MODULE_NAME.definition.json` file, you must change the `$type` value as shown in the following example.

```
"$type": "containerModule"
```

There are two types of container modules: layout container modules and page container modules.

## Layout container modules

Layout container modules are useful when you must make a complex module out of multiple simple modules, and you want to control the layout of those simple modules. For example, you can use a header layout container module that is made up of required or optional sub-modules, such as search, sign-in, and navigation modules. The purpose of the layout container is to provide an adaptive layout.

For more information, see [Create a layout container module](#).

## Page container modules

Page container modules contain the core structure for page authoring. For example, you can create a page container where slots are defined for the header area, main content area, and footer area. A page container is just a module that controls the layout of a set of named slots, and that can be embedded only at the root of a page. Each page must have only one page container. This page container is added to a template in the authoring tools.

Like layout container modules, page container modules can define named slots that are exposed to template authors. Page authors can configure which modules go into each slot, and the rendering code for the container controls the layout of those slots. Configuration settings can also be exposed to page authors, so that they can do additional configuration of the layout.

For more information, see [Create a page container module](#).

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

[Test modules by using module mocks](#)

[Test modules by using page mocks](#)

[Localize a module](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a layout container module

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to create, test, and preview a layout container module.

## Overview

The Microsoft Dynamics 365 Commerce online software development kit (SDK) provides a `yarn msdyn365 add-module MODULE_NAME` command-line interface (CLI) command. To create a new layout container module, you run this command to create a standard module, replacing `MODULE_NAME` with the name that you want to give to your module. Then change the `$type` value to `containerModule`.

### Examples

The following example shows how to create a container module that is named `campaign-container`.

```
yarn msdyn365 add-module campaign-container
```

After the command has finished running, open the new module's definition file, `campaignContainer.definition.json`, and change the `$type` value to `containerModule`.

The following example shows the addition of a `slots` section that contains two slots for this container.

```
{
  "$type": "containerModule",
  "friendlyName": "Sample Container",
  "name": "campaign-container",
  "description": "Sample container module",
  "categories": ["container"],
  "tags": [],
  "config": {
  },
  "slots": {
    "slot1": {
      "friendlyName": "Content Slot 1",
      "description": "Content to be rendered in container. Max 2",
      "allowedTypes": ["*"],
      "max": 2,
      "min": 0
    },
    "slot2": {
      "friendlyName": "Content Slot 2",
      "description": "Content to be rendered in container. Max 2",
      "allowedTypes": ["*"],
      "max": 2,
      "min": 0
    }
  }
}
```

The following example shows the module's React view file, `campaignContainer.view.tsx`, which uses the slots for the container. Note that this example uses a single React/View file, and that the `campaignContainer.tsx` can be deleted.

```

import * as React from 'react';
import { ICampaignContainerData } from './campaignContainer.data';
import { ICampaignContainerProps } from './campaignContainer.props.autogenerated';

const CampaignContainerView: React.FC<ICampaignContainerViewProps> = props => {
  return (
    <div {...this.props.renderModuleAttributes(this.props)} id={this.props.id}>
      <div className='row'>
        {slots.slot1[0]}
        {slots.slot2[0]}
      </div>
    </div>
  );
};

export default CampaignContainer;

```

## Test a layout container module

To test a container module in a local development environment, you must use a page mock.

The following example shows a sample page mock, campaign-containerMock.json, that you can use for testing. The file is saved in the \src\pageMocks directory.

```

{
  "exception": null,
  "pageRoot": {
    "id": "core-root_0",
    "typeName": "core-root",
    "modules": {
      "body": [
        {
          "id": "default-page_0",
          "typeName": "default-page",
          "modules": {
            "primary": [
              {
                "id": "primaryArea__0",
                "typeName": "campaign-container",
                "modules": {
                  "slot1": [
                    {
                      "id": "ProductFeature__0",
                      "typeName": "productFeature",
                      "config": {
                        "imageAlignment": "left",
                        "productTitle": "Retro Horn Rimmed Keyhole Nose Bridge Round
Sunglasses",
                        "productDetails": "High-quality and pioneered with the
perfect blend of timeless classic and modern technology with hint of old school glamor.",
                        "productImage": {
                          "src": "https://bit.ly/33cMGxr",
                          "altText": "Retro Horn Rimmed Keyhole Nose Bridge Round
Sunglasses"
                        },
                        "buttonText": "Buy Now",
                        "productIds": "68719498121"
                      },
                    },
                  ],
                  "slot2": [
                    {
                      "id": "ProductFeature__1",
                      "typeName": "productFeature",
                      "config": {

```



```

    "imageAlignment": "right",
    "productIds": "22565430170",
    "buttonText": "Buy Now"
  }
}
]
}
]
},
"renderingContext": {
  "gridSettings": {
    "xs": {
      "w": 767
    },
    "sm": {
      "w": 991
    },
    "md": {
      "w": 1199
    },
    "lg": {
      "w": 1599
    },
    "xl": {
      "w": 1600
    }
  },
  "staticContext": {
    "staticCdnUrl": "/_scnr/"
  },
  "locale": "en-us"
},
"statusCode": 200
}

```

## Preview the page

To preview the page in a local web browser, follow these steps.

1. At a command prompt, go to your root SDK folder, and run the **yarn start** command. Here is an example.

```
c:\repos\Msdyn365.Commerce.Online>yarn start
```

2. In a web browser, open the following URL to view the module:

`https://localhost:4000/page?mock=campaign-containerMock` . Notice the name of the page mock in the `mock=` query string parameter.

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

Test modules by using module mocks

Test modules by using page mocks

Container modules

Create a page container module

Localize a module

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a page container module

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to create a page container module in Microsoft Dynamics 365 Commerce.

## Overview

A page container is a module that controls the core structure of a page through specific layout regions that are known as *slots*. For example, a page container might have slots that are defined for the header area, main content area, and footer area. A page container can be embedded only at the root of a page, and each page must have only one page container.

Like layout container modules, page container modules can define *named slots* that are exposed in the template authoring tool. Page authors can configure which modules go into each slot, and the container rendering code then controls the layout of those slots. Configuration settings can also be exposed to page authors so that they can further configure the layout.

## Create a new page container module

The Microsoft Dynamics 365 Commerce online software development kit (SDK) provides a `yarn msdyn365 add-module MODULE_NAME` command-line interface (CLI) command. To create a page container module, you use this command to create a new module, and then change the `$type` value to `pageModule`.

For example, run the following command to create a module that is named `campaign-page-container`.

```
yarn msdyn365 add-module campaign-page-container
```

Then open the definition file for the new module, `campaign-page-container.definition.json`, and change the `$type` value to `pageModule`.

In the following example, notice that the `slots` section contains the various named slots that the page container supports. To restrict a slot so that only a specific set of modules can be put into it, use the `"allowedTypes"` array to define the list of allowed modules. Alternatively, use an asterisk (\*) to allow any module to be put into the slot.

```

{
  "$type": "pageModule",
  "friendlyName": "Campaign Page Container",
  "name": "campaign-page-container",
  "description": "This is a test page container.",
  "categories": ["page"],
  "tags": [],
  "config": {},
  "slots": {
    "header":
    {
      "friendlyName": "Header Slot",
      "description": "This is the header slot",
      "allowedTypes": ["header"],
      "max": 1,
      "min": 0
    },
    "primary":
    {
      "friendlyName": "Main Slot",
      "description": "This is the Primary slot",
      "allowedTypes": ["campaign-container", "carousel", "content-block", "product-collection"],
      "max": 5,
      "min": 0
    },
    "footer":
    {
      "friendlyName": "Footer Slot",
      "description": "This is the footer slot",
      "allowedTypes": ["footer"],
      "max": 1,
      "min": 0
    }
  }
}

```

In the `MODULE_NAME.tsx` view file, you can define the HTML structure for the slots on the page.

The following example shows an excerpt from a React view file (`campaign-page-container.view.tsx`) that takes advantage of the slots for the container.

```

...
return (
  <div {...this.props.renderModuleAttributes(this.props)} id={this.props.id}>
    <div className='row'>
      {slots.header[0]}
    </div>
    <div className='row'>
      {slots.primary[0]}
    </div>
    <div className='row'>
      {slots.footer[0]}
    </div>
  </div>
);
}
...

```

## Test a page container module

To test a page container module in a local development environment, you must use a page mock.

The following example shows a page mock that can be used for testing. It's saved in the `/src/pageMocks`

directory as campaignContainerMock.json.

```
{
  "exception": null,
  "pageRoot": {
    "id": "core-root_0",
    "typeName": "core-root",
    "modules": {
      "body": [
        {
          "id": "default-page_0",
          "typeName": "campaign-page-container",
          "modules": {
            "primary": [
              {
                "id": "primaryArea__0",
                "typeName": "myContainer"
                "modules": {
                  "slot1": [
                    {
                      "id": "ProductFeature__0",
                      "typeName": "productFeature",
                      "config": {
                        "imageAlignment": "left",
                        "productTitle": "Men's Grand Wingtip Shoe",
                        "productDetails": "Genuine leather crafted with
perfection.",
                        "buttonText": "Buy Now"
                      }
                    }
                  ],
                  "slot2": [
                    {
                      "id": "ProductFeature__1",
                      "typeName": "productFeature" ,
                      "config": {
                        "imageAlignment": "right",
                        "productTitle": "Men's Grand Wingtip Shoe",
                        "productDetails": "Genuine leather crafted with
perfection.",
                        "buttonText": "Buy Now"
                      }
                    }
                  ]
                }
              }
            ]
          }
        }
      ]
    }
  },
  "renderingContext": {
    "staticContext": {
      "staticCdnUrl": "/_scnr/"
    },
    "locale": "en-us"
  },
  "statusCode": 200
}
```

Notice that the preceding page mock defines only the **primary** slot in the page container. However, you can also include the **header** and **footer** slots as you require.

## Preview the page

To preview the page in a local web browser, follow these steps.

1. At a command prompt, go to your root SDK folder, and run the **yarn start** command. Here is an example.

```
c:\repos\Msdyn365.Commerce.Online>yarn start
```

2. In a web browser, open the following URL to view the module:

`https://localhost:4000/page?mock=campaign-containerMock` . Note the name of the page mock in the **mock=** query string parameter.

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

[Test modules by using module mocks](#)

[Test modules by using page mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Localize a module](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Localize a module

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to localize a module for rendering. It also describes how to localize general module information, such as the module name, description, and configuration fields.

## Localize module-rendered strings

For modules that render strings, the data might already be localized when it's received from the content management system (CMS). For example, a product title might come back from Dynamics 365 Commerce in a localized format. Therefore, it doesn't require localization. However, some strings that are defined in the module might require localization. For example, a module that renders a list page might have **Next** and **Previous** buttons that are used for page navigation. The labels for those buttons must be localized.

### Create a new resource string

Resources are stored in locale-specific JavaScript Object Notation (JSON) files under the `/src/resources/` directory. A `global.json` file is used as the base for all locales, and [n-part locales](#) are supported.

Here is an example of the file structure:

- `/src`
  - `/resources`
    - `/modules`
      - `global.json`
      - `en-us.json`
      - `de-de.json`
      - `fr-fr.json`
      - `fr.json`

Each resource file contains key/value pairs, and an optional **comment** property can provide additional context about the resource key. All modules share a single set of resource files. Therefore, modules can share the same resource keys.

The following example shows a resource schema.

```
{
  "<property_name>":
  {
    "value": "",
    "_value.comment": ""
  }
}
```

The following example shows a resource file.

```
{
  "noResultsForRefinersText": {
    "value": "No results found for refinement criteria",
    "_value.comment": "Message to display when no products are returned with applied refinement
criteria"
  },
  "resultNotFoundText": {
    "value": "Result not found text",
    "_value.comment": "Result not found text for a category"
  },
}
```

### Reference a resource key in a module

To use a resource string in a module, you must reference the resource string keys under the **resources** node in the module definition. Default values can also be provided. Here is an example.



```

{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "categories": [
    "storytelling"
  ],
  "tags": [
    ""
  ],
  "dataActions": {},
  "config": {
    "imageAlignment": {
      "friendlyName": "Image Alignment",
      "description": "Sets the desired alignment of the image, either left or right on the text.",
      "type": "string",
      "enum": {
        "left": "Left",
        "right": "Right"
      },
      "default": "left",
      "scope": "module",
      "group": "Layout Properties"
    },
    "productTitle": {
      "type": "string",
      "friendlyName": "Product Title",
      "description": "Product placement title"
    },
    "productDetails": {
      "type": "richText",
      "friendlyName": "Product Details",
      "description": "Rich text representing the featured product details"
    },
    "productImage": {
      "type": "image",
      "friendlyName": "Product Image",
      "description": "Image representing the featured product"
    },
    "buttonText": {
      "type": "string",
      "friendlyName": "Button Text",
      "description": "Text to show on the call to action button"
    },
    "productIds": {
      "friendlyName": "Product ID",
      "description": "Provide a Product Id that the module will display",
      "type": "string",
      "scope": "module",
      "group": "Content Properties"
    }
  },
  "resources": {
    "nextButtonText": {
      "value": "next",
      "comment": "Text for the next button"
    },
    "previousButtonText": {
      "value": "previous",
      "comment": "Text for the previous button"
    }
  }
}

```

**Access resources in the module view file**

Resources can be accessed in the module React file and module view file by using the **this.props.resources** property, as shown in the following example.

```
<button className="nextButton">
  {this.props.resources.nextButtonText}
</button>
```

## Localize module fields for authoring tools

Modules should be built so that they support localization. This guideline applies to any authoring metadata in the module definition file, including the module name, description, and configuration fields.

Each resource file must contain modules and properties groups. All the module-related authoring strings should be grouped under the **modules** section. There should be a child section for each module, and each module section should, in turn, include the related authoring property pairs as children. Each property is an object that has a **"value"** property and an optional **"\_value.comment"** property inside the module section.

The following example shows a resource schema.

```

{
  "modules": {
    "<module_name>": {
      "friendlyName": {
        "value": "",
        "_value.comment": ""
      },
      "description": {
        "value": "",
        "_value.comment": ""
      },
    },
    "config": {
      "<property_name>": {
        "friendlyName": {
          "value": "",
          "_value.comment": ""
        },
        "description": {
          "value": "",
          "_value.comment": ""
        },
        "errorMessage": {
          "value": "",
          "_value.comment": ""
        },
        "properties": {
          ...
        }
      }
    },
    "slots": {
      "content": {
        "friendlyName": {
          "value": "",
          "comment": ""
        }
      }
    },
    "options": {
      "enumKey": {
        "value": "",
        "_value.comment": ""
      }
    },
  },
}

```

The following example shows a resource file.

```

{
  "modules": {
    "hero": {
      "friendlyName": {
        "value": "Hero Module",
        "_value.comment": ""
      },
      "description": {
        "value": "Hero with slides",
        "_value.comment": ""
      },
      "config": {
        "headerText": {
          "friendlyName": {
            "value": "Partner hero",
            "_value.comment": ""
          }
        },
        "alignment": {
          "friendlyName": {
            "value": "Hero Image Alignment",
            "_value.comment": ""
          },
          "options": {
            "left": {
              "value": "Left Image",
              "_value.comment": ""
            },
            "right": {
              "value": "Right Image",
              "_value.comment": ""
            }
          }
        }
      },
      "slots": {
        "content": {
          "friendlyName": {
            "value": "Content Slots",
            "_value.comment": ""
          },
          "description": {
            "value": "Content to be rendered in container. Max 2",
            "_value.comment": ""
          }
        }
      },
      "options": {
        "text": {
          "value": "text",
          "_value.comment": ""
        },
        "glyph": {
          "value": "glyph",
          "_value.comment": ""
        }
      }
    }
  }
}

```

### Generate a resource global.json file

You can generate global.json files for module resources and authoring resources by running the `yarn msdyn365 generate-resources src` command in the SDK root folder. This command picks up both the module and authoring strings that are defined in the \*.definition.json files, and it generates the

resources/modules/global.json and resources/authoring/global.json files.

### Example global.json file

```
{
  "nextButtonText": {
    "value": "next",
    "_value.comment": "Text for the next button"
  },
  "previousButtonText": {
    "value": "previous",
    "_value.comment": "Text for the previous button"
  }
}
```

### Example fr-fr.json localized file

```
{
  "nextButtonText": {
    "value": "prochain",
    "_value.comment": "Text for the next button"
  },
  "previousButtonText": {
    "value": "précédent",
    "_value.comment": "Text for the previous button"
  }
}
```

## Override a resource string for a theme

The built-in module library set of modules and themes may have localized resource strings that you want to override. To override resource strings for a theme, modify the global.json resource file located in the src/resources/modules directory using the pattern in the following example.

```
"{ThemeNamespace}.{ThemeName}.{ResourceString}": {
  "value" : "",
  "_value.comment": ""
}
```

### Override resource strings for preinstalled themes

To override resource strings for preinstalled themes (fabrikam or starter), use **@msdyn365-commerce-modules** as the theme namespace. The following example shows how to change the sign-in link text on the fabrikam theme.

```
"@msdyn365-commerce-modules.fabrikam.signInLinkText": {
  "value": "Sign in now",
  "_value.comment": "Sign-in Link Text"
}
```

### Override resource strings for custom or local themes

For custom or local themes, use **local** for the theme namespace. The following example shows how to change the sign-in link text for a custom theme called "adventureworks."

```
"__local__.adventureworks.signInLinkText": {
  "value": "Log in",
  "_value.comment": "Log in Link Text"
}
```

#### NOTE

For [shared themes](#), child themes inherit all of the resources string overrides tied to the parent theme.

## Test localized content

To test localized content, you must use a page mock and change the locale to the locale you're testing. For more information about page mocks, see [Test modules by using page mocks](#).

## Additional resources

[Create a new module](#)

[Clone a module library module](#)

[Add module configuration fields](#)

[Preview and debug a module](#)

[Test modules by using module mocks](#)

[Test modules by using page mocks](#)

[Container modules](#)

[Create a layout container module](#)

[Create a page container module](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Request properties object

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic covers the request properties object in Microsoft Dynamics 365 Commerce.

## Overview

The request properties object represents an HTTP request and includes various data properties, such as the requested URL, locale, device, user, cookies, and query string parameters. To get the request information, modules can access a read-only request context object that is named **this.props.context**, and they can change module behavior as required.

## Example

The following example shows how to access the request properties object from within the request context.

```
if (this.props.context.request.user.isAuthenticated) {
    userName = this.props.context.request.user.signInName ? this.props.context.request.user.signInName : '';
    firstName = this.props.context.request.user.firstName ? this.props.context.request.user.firstName : '';
    lastName = this.props.context.request.user.lastName ? this.props.context.request.user.lastName : '';
}
```

## General properties

- **url** – The requested URL.
- **locale** – The locale context (for example, "en-us").
- **textDirection** – The text direction (the possible values are "rtl" and "ltr").
- **sitePath** – The full path of the site.
- **device** – The device that the request came from.
  - **Type** – The device type (for example, "pc").
- **user** – Information about the user. The following properties are included:
  - **token**
  - **isAuthenticated**
  - **signInURL**
  - **signoutURL**
  - **signUpUrl**
  - **editProfileUrl**
  - **signInName**
  - **name**
  - **firstName**
  - **lastName**
  - **emailAddress**
  - **customerAccountNumber**
- **query** – A list of query string parameters.

- cookies – Cookie information.

## Interface

```
interface IParsedQSP<TValue> {
    hasValue: boolean;
    isTruthy: boolean;
    value: TValue | undefined;
}

interface IRequestContextUrl {
    serverUrl: string;
    serverPageUrl: string;
    requestUrl: URL;
    staticCdnUrl: string;
}

interface IRequestContextParams {
    mock?: string;
    isDebug: boolean;
    isEditor: boolean;
    concatJs: IParsedQSP<boolean | string | number | undefined>;
    theme: string;
}

interface IRequestContextDevice {
    Type: string;
}

interface IRequestContextUser {
    token: string;
    isAuthenticated: boolean;
    signInUrl?: string;
    signOutUrl?: string;
    signUpUrl?: string;
    editProfileUrl?: string;
    signInName?: string;
    name?: string;
    firstName?: string;
    lastName?: string;
    emailAddress?: string;
    customerAccountNumber?: string;
}

interface IRequestContextFeatures {
    [switchName: string]: boolean;
}

interface IRequestContextHeaders {
    readonly [header: string]: string;
}

interface IRequestContext {
    url: IRequestContextUrl;
    locale: string;
    market?: string;
    textDirection: string;
    sitePath?: string;
    params: IRequestContextParams;
    device: IRequestContextDevice;
    user: IRequestContextUser;
    app: IGeneric<IAny>;
    query?: IDictionary<string>;
    apiSettings: ICommerceApiSettings;
    channel?: IChannelConfiguration;
    gridSettings?: IGridSettings;
    urlTokens: IUrlTokens;
```



```

    operationId: string;
    features: IRequestContextFeatures;
    pageData: IGeneric<IAny>;
    headers: IRequestContextHeaders;
    cookies: ICookieContext;
}

```

## Test a module that has an authenticated signed-in state

Some modules might require that the state be **signed-in**. To test these modules, you can create a test page mock that has user authentication information.

To get started, follow these steps.

1. Load the e-Commerce webpage that you're working on, and sign in or create a new account.
2. Open your web browser's debugging tools. For example, if you're using Google Chrome, you can open the developer tools by pressing the F12 key.
3. Enter `___initialData___requestContext.user` in the console to get the user information. (User information is available in the `___initialData___requestContext.user` global JavaScript variable.)
4. Add the module that must be tested to a page mock.
5. In the `renderingContext` section of the page mock, add the following `userContext` section.

```

"userContext": {
  "token": "<TOKEN>",
  "isAuthenticated": true,
  "signInUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signin",
  "signOutUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signout",
  "signUpUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signup",
  "editProfileUrl": "https://dev.fabrikam.com/fedev/_msdyn365/editprofile",
  "signInName": "<User Name>",
  "firstName": "<User First Name>",
  "lastName": "<User Last Name>",
  "tenantId": "",
  "customerAccountNumber": "<User Account Number(HQ)>",
  "name": "<User Name>",
  "emailAddress": "<User Email Address>"
},

```

6. Update the user information from the web browser's debugging tools.

The user information can now be obtained in the React component from within the `this.props.context.request.user` object.

Here is an example of the page mock.

```

{
  "exception": null,
  "pageRoot": {
    "id": "core-root_0",
    "typeName": "core-root",
    "modules": {
      "body": [
        {
          "id": "default-page_0",
          "typeName": "default-page",
          "modules": {
            "primary": [
              {

```

```

        "id": "ProductFeature_0",
        "typeName": "product-feature",
        "config": {
            "imageAlignment": "left",
            "productTitle": "Retro Horn Rimmed Keyhole Nose Bridge Round
Sunglasses",
            "productDetails": "High-quality and pioneered with the perfect blend of
timeless classic and modern technology with hint of old school glamor.",
            "productImage": {
                "src" : "https://bit.ly/33cMGxr",
                "altText" : "Retro Horn Rimmed Keyhole Nose Bridge Round Sunglasses"
            },
            "buttonText": "Buy Now"
        }
    }
}
],
},
"renderingContext": {
    "gridSettings": {
        "xs": {
            "w":767
        },
        "sm": {
            "w":991
        },
        "md": {
            "w":1199
        },
        "lg": {
            "w":1599
        },
        "xl": {
            "w":1600
        }
    },
    "staticContext": {
        "staticCdnUrl": "/_scnr/"
    },
    "locale": "en-us",
    "userContext": {
        "token": "<TOKEN>",
        "isAuthenticated": true,
        "signInUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signin",
        "signOutUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signout",
        "signUpUrl": "https://dev.fabrikam.com/fedev/_msdyn365/signup",
        "editProfileUrl": "https://dev.fabrikam.com/fedev/_msdyn365/editprofile",
        "signInName": "<User Name>",
        "firstName": "<User First Name>",
        "lastName": "<User Last Name>",
        "tenantId": "",
        "customerAccountNumber": "<User Account Number(HQ)>",
        "name": "<User Name>",
        "emailAddress": "<User Email Address>"
    },
},
},
"statusCode": 200
}

```

## Additional resources

[App settings](#)

[Platform settings file](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Mock the signed-in state during local development](#)

[Configure module properties to be shown based on context](#)

[Globalize modules by using the CultureInfoFormatter class](#)

[Set up Azure Key Vault for secure key management](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# App settings

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers app settings in Microsoft Dynamics 365 Commerce.

## Overview

The `\src\settings\app.settings.json` file holds app settings for global configurations, routes, and themes. The following example shows a JavaScript Object Notation (JSON) file that includes a section for each type of setting.

```
{
  "config": {
    "logoUrl": {
      "type": "image",
      "friendlyName": "Logo Image",
      "description": "Logo Image"
    }
  },
  "routes": {
    "cart": {
      "friendlyName": "Cart Page Route",
      "description": "Cart Page Route",
      "default": "/cart2"
    },
    "checkout": {
      "friendlyName": "Checkout Page Route",
      "description": "Checkout Page Route",
      "default": "/checkout"
    }
  },
  "themes": {
    "default": {
      "friendlyName": "Default Theme"
    }
    "spring": {
      "friendlyName": "Spring Theme"
    }
  }
}
```

## Config section

The **config** section of the `app.settings.json` file supports global configuration fields. These fields can be set at the site level in the authoring tools, and they are shared across all modules by using the `this.props.context.app.config.*` application programming interface (API).

The global configuration fields are helpful when you have data that should be set only one time across the whole online site, but the data is required by multiple modules. For example, an API key for ratings and reviews has the same value across the online site, and that value is required by multiple modules. Modules themselves also support site-scoped configuration settings, but those settings can't be shared across different modules.

The schemas for these configurations follow the same schemas that are used for module configurations.

## Routes section

The **routes** section in the app.settings.json file is used to add URL routes in a module. For example, if a module must create a link to the cart page, a "cart" route is added to the JSON file. The route can then be retrieved from the module code by using the authoring tools.

Routes can be accessed in the module view file by using the **this.props.context.app.routes.\*** API.

## Themes section

The **themes** section in the app.settings.json file is used to expose themes to the authoring tools. The themes in this section should match the theme names in the \src\themes\ directory. They should provide localizable friendly names that the authoring tools can show when they are used to set a theme at the site level, or in the template, layout, or page editor.

### Localized app settings

Both the **friendlyName** property and the **description** property of each app setting should be localized for the site locale in the Dynamics 365 Commerce authoring tools. The src/resources/authoring/global.json file should be updated so that it includes a **settings** property, and all the settings that are related to resource strings should be listed under that property.

### Resource schema

```
{
  "settings": {
    "<setting_property>": {
      "friendlyName": {
        "value": "",
        "_value.comment": ""
      },
      "description": {
        "value": "",
        "_value.comment": ""
      },
    },
  },
},
}
```

Here is an example of a resource file.

```
{
  "settings": {
    "logoUrl": {
      "friendlyName": {
        "value": "Logo Image",
        "_value.comment": ""
      },
      "description": {
        "value": "Logo Image",
        "_value.comment": ""
      }
    },
    "cart": {
      "friendlyName": {
        "value": "Cart URL",
        "_value.comment": ""
      },
      "description": {
        "value": "Cart page route",
        "_value.comment": ""
      }
    }
  }
}
```

## Additional resources

[Request properties object](#)

[Platform settings file](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Mock the signed-in state during local development](#)

[Configure module properties to be shown based on context](#)

[Globalize modules by using the CultureInfoFormatter class](#)

[Set up Azure Key Vault for secure key management](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Platform settings file

2/18/2021 • 2 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic covers the various properties that are configured in the platform settings file in Microsoft Dynamics 365 Commerce.

The `platform.settings.json` file under the `\src\settings\` directory holds various platform property settings that are used by the Commerce e-commerce runtime. This file might not exist by default. If it doesn't exist, you can add it under the `\src\settings\` directory. The following example of a `platform.settings.json` file shows various supported property settings.

```
{
  "dataActionTimeoutInMs": 4000,
  "minClientChunkSize": 30000,
  "excludeModules": [ ],
  "namespaceExtensions" : [ ]
}
```

## dataActionTimeoutInMs

The `dataActionTimeoutInMs` property defines the maximum amount of time, in milliseconds, that data actions will wait for a response before they time out. The time-out represents a lower bound for page response, because the action framework will wait as long as the defined time-out before it times out and returns the page. The default value is 4,000 milliseconds (4 seconds).

## minClientChunkSize

The `minClientChunkSize` property defines the minimum size, in bytes, of webpack JavaScript chunks that will be sent to the browser. JavaScript chunks that are smaller than the minimum size will be grouped together to form chunks that are larger than the minimum size. The smaller the minimum size is, the more chunks will be generated. In this case, more code splitting will occur, and less unused JavaScript code will be included. However, many smaller chunks must be downloaded. By contrast, the larger the minimum size is, the fewer overall chunks will be generated. In this case, fewer JavaScript files must be downloaded, but some unused JavaScript code might be included. The default value is 30,000 bytes (30 KB).

## excludedModules

The `excludedModules` property defines a set of modules that will be excluded from webpack JavaScript chunks. Commerce modules are bundled into JavaScript chunks and sent to the browser on the client side. However, if modules aren't required on a site, they can be excluded to help reduce the size of JavaScript chunks and help increase the speed of page loads.

## namespaceExtensions

The `namespaceExtensions` property defines the supported namespaces that are used for module registration. By default, the only supported namespace is `@msdyn365-commerce-modules`. This namespace contains all the module library modules and the core set of modules. The module package name is defined in the following format: `<namespace>/<module_name>`. If modules are published that use a new namespace, the namespace can be added to the settings.

## Additional resources

[Request properties object](#)

[App settings](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Mock the signed-in state during local development](#)

[Globalize modules by using the `CultureInfoFormatter` class](#)

[Set up Azure Key Vault for secure key management](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Extend a module definition file

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to extend a module definition file. For example, you can create an extended module of another module to add new configuration fields.

## Overview

When you extend a property that is an object, you must extend the whole object. For example, to add a configuration property to your extended module, you first copy the existing configuration properties from the parent module to the child module. You then add the desired property.

## Examples

The following example of a module definition file shows how a core module can be extended by using a **\$ref** command to the core script injector module.

```
{
  "$ref": "@d365-commerce-modules/core-components/dist/lib/modules/script-injector/script-
injector.definition.json",
  "friendlyName": "Analytics Service",
  "name": "AnalyticsService"
  ...
}
```

The **\$ref** command can also include a relative path to another module in your `/src/modules/` directory.

```
{
  "$ref": "../productFeature/productFeature.definition.json",
  "friendlyName": "Extended Product Feature Module",
  "name": "extendedProductFeature",
  "config": {
    "extendedProductData": {
      "friendlyName": "Extended Product Data",
      "description": "Additional product data.",
      "type": "string"
    }
  }
}
```

After deployment, both the base module and the extended module appear in Microsoft Dynamics 365 Commerce.

## Additional resources

[Request properties object](#)

[App settings](#)

[Platform settings file](#)

[Cookie API overview](#)

[Interactive components overview](#)

Mock the signed-in state during local development

Configure module properties to be shown based on context

Globalize modules by using the `CultureInfoFormatter` class

Set up Azure Key Vault for secure key management

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Cookie API overview

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of the application programming interfaces (APIs) in the Microsoft Dynamics 365 Commerce e-Commerce software development kit (SDK) that are used to set and get cookie data.

## Cookie consent

Before cookies can be stored, the user must give consent. The Dynamics 365 Commerce online SDK provides utilities that help guarantee that the read/write operation of a cookie depends on user consent.

## Cookie APIs

The Dynamics 365 Commerce online SDK provides a set of APIs that access cookies from within the `props.context.request` API, as shown in the following example.

```
get<T>(cookieName: string, isEssential?: boolean): ICookieValue<T>;
set<T>(cookieName: string, cookieValue: T, options?: ICookieSetOptions): void;
remove(cookieName: string): void;
isConsentGiven(): boolean;
setConsentCookie(): void;
deleteConsentCookie(): void;
```

### Obtain user consent

The `setConsentCookie()` API is used to obtain user consent before cookies can be written.

### Determine whether user consent has been given

The `isConsentGiven()` API is used to determine whether user consent has been given.

### Set a cookie

The following example shows how to set a cookie.

```
this.props.context.request.cookies.set<string>('favoriteColor', 'blue');
```

If user consent isn't given before this API is called, the SDK maintains a queue of all `set` operations and sets a cookie only after the user gives consent.

### Get the value of a cookie

The following example shows how to get the value of a cookie.

```
const favColor = this.props.context.request.cookies.get<string>('favoriteColor');
```

## Additional resources

[Request properties object](#)

[App settings](#)

[Platform settings file](#)

[Extend a module definition file](#)

[Interactive components overview](#)

[Mock the signed-in state during local development](#)

[Configure module properties to be shown based on context](#)

[Globalize modules by using the CultureInfoFormatter class](#)

[Set up Azure Key Vault for secure key management](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Interactive components overview

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic covers interactive components that let site authors edit fields for text, rich text, links, images, and videos directly in visual page builder, the what-you-see-is-what-you-get (WYSIWYG) preview canvas in Microsoft Dynamics 365 Commerce site builder.

## Overview

In Commerce site builder, page or fragment authors can use interactive components to edit fields for text, rich text, links, images, and videos directly in visual page builder. Interactive components are released with the Commerce online software development kit (SDK) and include **Msdyn365.Text** for text, **Msdyn365.RichTextComponent** for rich text, **Msdyn365.Links** for links, **Msdyn365.Image** for images, and **Msdyn365.Video** for videos. After the interactive components are implemented, site builder will allow for inline editing of text, and it will open a picker window for links, images, and videos.

For the best authoring experience, module developers should use interactive components when they render configuration fields, to allow for inline editing. Any custom implementations of these components can be made interactive by wrapping the component in the **EditableField** higher-order component (HOC). Follow the guidelines in this topic to support interactive components inside a custom module.

## How interactive components work

An interactive component hooks an event handler to the component that will do the work of setting the specific configuration field that is required.

The following example shows how a text configuration field uses an interactive component to allow for inline editing of the text. A **productTitle** configuration value is specified in the **config** section of the module definition file.

```
...
  "config": {
    "productTitle": {
      "type": "string",
      "friendlyName": "Product Title",
      "description": "Product placement title"
    },
  },
}
```

To support an interactive canvas experience, the module uses the **Msdyn365.Text** interactive component. This interactive component specifies a **handleTextChange** event handler that will set the configuration value.

```
public handleTextChange = (event: Msdyn365.ContentEditableEvent) => this.props.config.productTitle!.text =
  event.target.value;

<Msdyn365.Text
  className={'product-title'}
  tag={'h2'}
  text={this.props.config.productTitle}
  editProps={{ onEdit: this.handleTextChange, requestContext: this.props.context.request }}
/>
```

# Interactive component reference

## Text component

After it's implemented, the text component lets site authors edit text inline, directly in visual page builder.

### Text component syntax

```
<Msdyn365.Text />
```

### Text component example

```
<Msdyn365.Text
  text={heading.text || ''}
  tag={'h2'}
  className={'display-4'}
  editProps={{ onEdit: this.handleTextChange, requestContext: this.props.context.request }}
/>

public handleTextChange = (event: Msdyn365.ContentEditableEvent) => this.props.config.heading!.text =
  event.target.value;
```

### Available properties

PROPERTY	DESCRIPTION	TYPE
text	The text to show.	String
tag	The HTML tag (for example, <b>DIV</b> or <b>H2</b> ) to use to render the text.	String
className	The Cascading Style Sheets (CSS) class name.	String
editProps	Properties that are required to enable interaction in site builder.	ITextEditProps

### ITextEditProps properties

PROPERTY	DESCRIPTION	TYPE
onEdit	The function handler that handles the text change event.	Function
requestContext	The Request context object.	IRequestContext

## Rich text component

After it's implemented, the rich text component lets site authors edit rich text inline, directly in visual page builder.

### Rich text component syntax

```
<Msdyn365.RichTextComponent />
```

### Rich text component example

```

<Msdyn365.RichTextComponent
  text={`<p className='d-none d-lg-block'>${paragraph}</p>`}
  editProps={{ onEdit: this.handleParagraphChange, requestContext: this.props.context.request }}
/>

```

#### Available properties

PROPERTY	DESCRIPTION	TYPE
text	The HTML text to show.	String
editProps	The properties that are required to enable interaction in site builder.	ITextEditProps

#### ITextEditProps properties

PROPERTY	DESCRIPTION	TYPE
onEdit	The function handler that handles the text change event.	Function
requestContext	The Request context object.	IRequestContext

### Links component

After it's implemented, the links component lets site authors edit an array of links in visual page builder.

#### Links component syntax

```

<Msdyn365.Links />

```

#### Links component example

```

<Msdyn365.Links
  links={config.links}
  editProps={{ onTextChange: this.handleLinkTextChange, requestContext: this.props.context.request }}
/>

```

#### Available properties

PROPERTY	DESCRIPTION	TYPE
link	The array of link data to render.	ILinksData[]
editProps	The properties that are required to enable interaction in site builder.	ITLinkEditProps

#### ILinksData properties

PROPERTY	DESCRIPTION	TYPE
linkText	The text to show as a link.	String
linkUrl	The URL to open.	String
openInNewTab	A flag that indicates whether the link should be opened on a new tab.	Boolean

PROPERTY	DESCRIPTION	TYPE
ariaLabel	The Accessible Rich Internet Applications (ARIA) label for accessibility.	String
className	The CSS class name.	String
key	The React key.	String
role	The role of the anchor tag (for example, <b>button</b> ).	String
additionalProperties	A dictionary of additional properties to add to the element.	[x: string]: string
linkTag	The HTML tag to use to render the link.	String
innerClassName	The CSS class name for the inner component in the anchor tag.	String
onClick	The click handler that handles the link click event.	Function

#### ILinkEditProps properties

PROPERTY	DESCRIPTION	TYPE
onTextChange	The function handler that handles the link text change event.	Function
requestContext	The Request context object.	IRequestContext

### Link component

After it's implemented, the link component lets site authors edit single links in visual page builder.

#### Link component syntax

```
<Msdyn365.Link />
```

#### Link component example

```
<Msdyn365.Link
  link={config.links[1]}
  editProps={{ onTextChange: this.handleLinkTextChange(1), requestContext: this.props.context.request }}
/>
```

#### Available properties

PROPERTY	DESCRIPTION	TYPE
links	The link data to render.	ILinksData
editProps	Properties that are required to enable interaction in site builder.	ILinkItemEditProps



## ILinksData properties

PROPERTY	DESCRIPTION	TYPE
linkText	The text to show as a link.	String
linkUrl	The URL to open.	String
openInNewTab	A flag that indicates whether the link should be opened on a new tab.	Boolean
ariaLabel	The ARIA label for accessibility.	String
className	The CSS class name.	String
key	The React key.	String
role	The role of the anchor tag (for example, <b>button</b> ).	String
additionalProperties	A dictionary of additional properties to add to the element.	[x: string]: string
linkTag	The HTML tag to use to render the link.	String
innerClassName	The CSS class name for the inner component in the anchor tag.	String
onClick	The click handler that handles the link click event.	Function

## ILinkItemEditProps properties

PROPERTY	DESCRIPTION	TYPE
onTextChange	The function handler that handles the link text change event.	Function
requestContext	The Request context object.	IRequestContext

## Image component

After it's implemented, the image component lets site authors edit images directly in visual page builder.

### Image component syntax

```
<Msdyn365.Image />
```

### Image component example

```
<Msdyn365.Image
  editProps={{ key: this.props.config.image || {}, requestContext: this.props.context.request }}
  {...config.image}
  {...imageProps}
/>
```

### Available properties

PROPERTY	DESCRIPTION	TYPE
src	The URL of the image source.	String
fallBackSrc	The fallback URL if the image can't be loaded.	String
altText	The alt text for the image.	String
caption	The caption string for the image.	String
title	The title for the image.	String
width	The width of the image.	number
height	The height of the image.	number
imageSettings	Settings for the image.	ImageSettings
additionalProperties	A dictionary of additional properties to add to the element.	[x: string]: string
editProps	Properties that are required to enable interaction in site builder.	IEditProps

#### IEditProps properties

PROPERTY	DESCRIPTION	TYPE
key	The path of the image property in the <b>config</b> section of the module definition file.	Object
requestContext	The Request context object.	IRequestContext

### Video component

After it's implemented, the video component lets site authors edit videos directly in visual page builder.

#### Video component syntax

```
<MsDyn365.Video></MsDyn365.Video>
```

#### Video component example

```
<MsDyn365.Video>
  <Player playerData={videoPlayerData} />
</MsDyn365.Video>
```

#### Available properties

PROPERTY	DESCRIPTION	TYPE
className	The CSS class name.	String

PROPERTY	DESCRIPTION	TYPE
editProps	Properties that are required to enable interaction in site builder.	IEditProps

#### IEditProps properties

PROPERTY	DESCRIPTION	TYPE
key	The path of the property in the <b>config</b> section of the module definition file.	Object
requestContext	The Request context object.	IRequestContext

## Generic editable HOC

Generic editable field components wrap any custom components and enable interactions in the context of site builder.

#### Generic editable HOC syntax

```
<EditableField />
```

#### Generic editable HOC example

```
<EditableField
  onChange={this.props.editProps ? this.props.editProps.onEdit : this.onEdit} // handle innerHTML change
  requestContext={this.props.editProps?.requestContext}
  fieldProps={{
    text: this.props.text,
    tag: this.props.tag,
    className: this.props.className
  }}
  type={FieldType.Text}
/>
```

#### Available properties

PROPERTY	DESCRIPTION	TYPE
fieldProps	Field properties.	[x: string]: string
type	The type of the field ( <b>text</b> , <b>image</b> , <b>video</b> , or <b>link</b> ).	FieldType
requestContext	The Request context object.	IRequestContext
onChange	The event handler for the <b>onChange</b> event.	Function
onMouseOver	The event handler for the <b>onMouseover</b> event.	Function
onMouseOut	The event handler for the <b>onMouseout</b> event.	Function

PROPERTY	DESCRIPTION	TYPE
onFocus	The event handler for the <b>onFocus</b> event.	Function
onBlur	The event handler for the <b>onBlur</b> event.	Function
placeholderText	The text to show when text is cleared from the field.	String
disabled	A flag that enables or disables editing of the file in site builder.	String

## Additional resources

[Request properties object](#)

[App settings](#)

[Platform settings file](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Mock the signed-in state during local development](#)

[Configure module properties to be shown based on context](#)

[Globalize modules by using the CultureInfoFormatter class](#)

[Set up Azure Key Vault for secure key management](#)

[Work with modules](#)

[Work with fragments](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Mock the signed-in state during local development

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to mock a signed-in user in a Dynamics 365 Commerce online local development environment.

Over the course of developing your e-Commerce online site, it may become necessary to develop and test scenarios for signed-in users. Rather than publish these pages and test against live pages, you can instead mock the signed-in state when running in developer mode.

## Configure your Azure AD B2C tenant

To take advantage of this feature, you must perform a one-time setup in your Azure Active Directory (Azure AD) business-to-consumer (B2C) tenant to allow you to mock the signed-in user status. To proceed with the following steps, you will need to be signed in as a user with global administrator privileges.

### Create a new resource owner password credentials (ROPC) flow

The first step is to create a new resource owner password credentials (ROPC) flow in your Azure AD B2C tenant.

To create a new ROPC flow, follow these steps.

1. Sign in to the [Azure portal](#) as the global administrator of your Azure AD B2C tenant, and then select the **Azure AD B2C** service.
2. Select **User flows** and **New user flow**.
3. Select **Sign in using resource owner password credentials (ROPC)**, and then select **Create**.
4. Enter a name for the user flow, for example "ROPC\_Auth". Copy and save the full name, as it will be used later as the `ropcUserFlowName` value in your `credentials.json` file.
5. Under **Application claims**, select **Show more**.
6. Select the following application claims:
  - **Display Name**
  - **Email Addresses**
  - **Given Name**
  - **Identity provider**
  - **SurName**
  - **User's object ID**
7. Select **OK**, and then select **Create**.
8. Select the new user flow, and then select **Run user flow**.

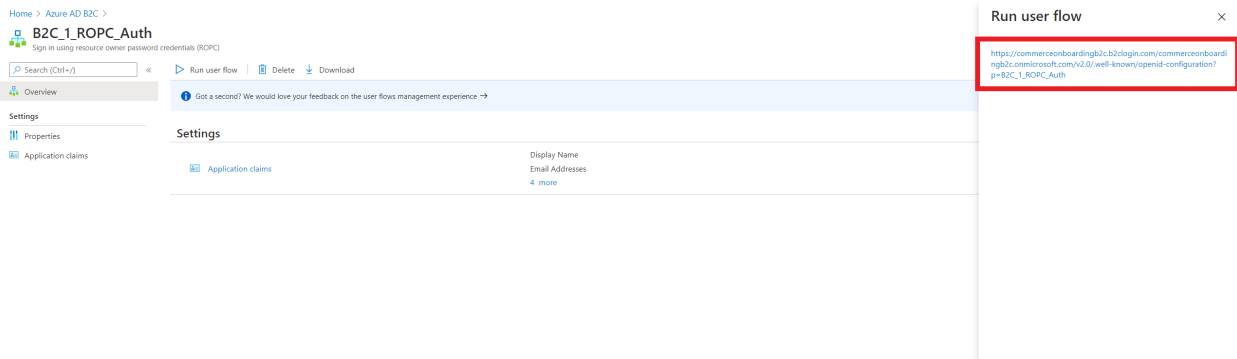
You have now created a new ROPC policy to enable local sign-in. Under **Run user flow** you should see an endpoint URL similar to

```
https://<LOGIN_DOMAIN>//<B2C_TENANT>.onmicrosoft.com/v2.0/.well-known/openid-configuration?p=B2C_1_ROPC_Auth
```

Take note of the `<LOGIN_DOMAIN>` and `<B2C_TENANT>` values from the URL, because this information will be used later in your `credentials.json` file.

In the following example image, the endpoint URL listed under **Run user flow** is

```
https://commerceonboardingb2c.b2clogin.com/commerceonboardingb2c.onmicrosoft.com/v2.0/.well-known/openid-configuration?p=B2C_1_ROPC_Auth
```



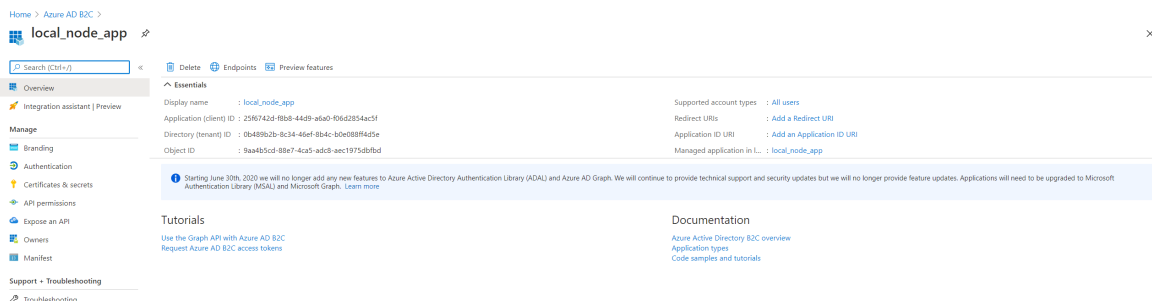
From the example above, you can obtain values for the `ropcUserFlowName`, `loginDomain`, and `b2cTenant` properties as follows:

PROPERTY NAME	EXAMPLE VALUE
<code>ropcFlowUserName</code>	B2C_1_ROPC_Auth
<code>loginDomain</code>	commerceonboardingb2c.b2clogin.com
<code>b2cTenant</code>	commerceonboardingb2c

### Create a native application

Next, you will create a native application meant to represent the Node application you will run during local development.

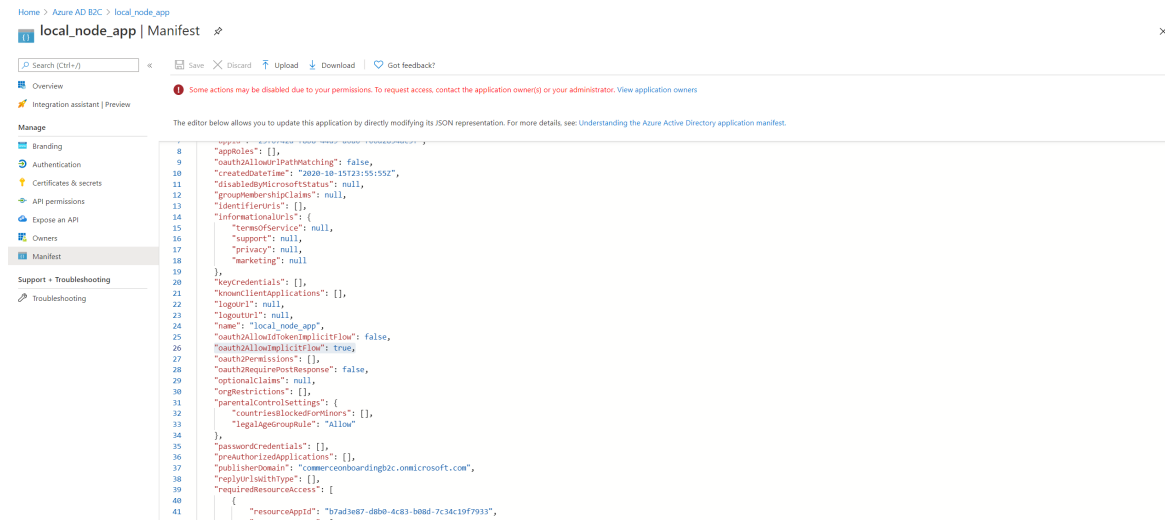
1. In the **Azure AD B2C** settings, select **App Registrations**, and then select **New registration**.
2. Enter a name for the application, for example "local\_node\_app".
3. For **Supported account types**, select **Accounts in any identity provider or organizational directory (for authenticating users with user flows)**.
4. For **Redirect URIs**, select **Public client/native (mobile & desktop)** from the drop-down list, and leave the URI as is.
5. Leave all other default values as is, and select **Register**.
6. Select the new application, and then copy and save the **Application (client) ID** value, as this ID will be used later as the `nativeApplicationId` property value in your `credentials.json` file.



7. In the left navigation pane under **Manage**, select **Authentication**.
8. Select **Try out the new experience** (if shown).
9. Under **Default client type**, select **Yes** for **Treat the application as a public client**. This setting is

required for the ROPC flow.

10. Select **Save**.
11. In the left navigation pane under **Manage**, select **Manifest** to open the manifest editor.
12. Set the `oauth2AllowImplicitFlow` attribute to `true`, and then select **Save**.



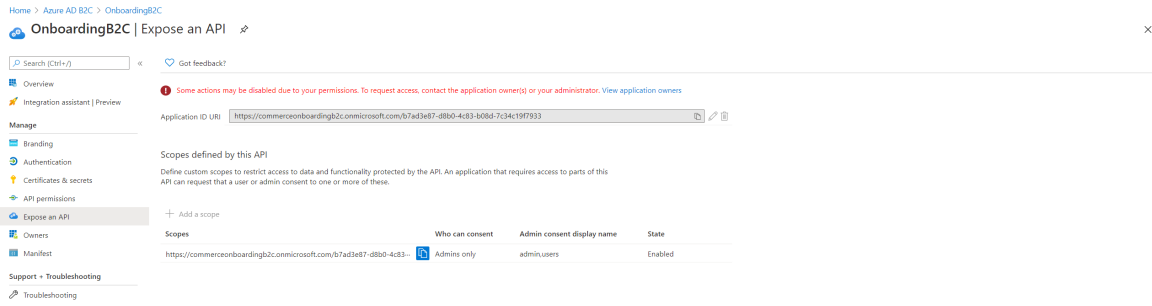
You have now created a new native application that will be used to represent your local Node application.

From the examples above, you have now obtained the following information:

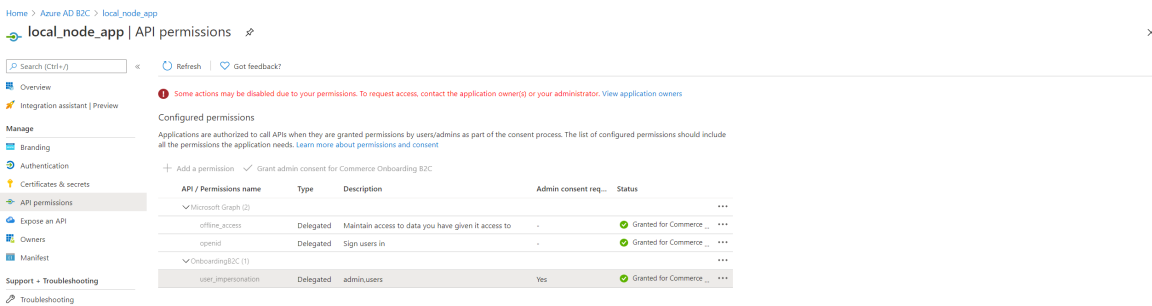
PROPERTY NAME	EXAMPLE VALUE
ropcFlowUserName	B2C_1_ROPC_Auth
loginDomain	commerceonboardingb2c.b2clogin.com
b2cTenant	commerceonboardingb2c
nativeApplicationId	25f6742d-f8b8-44d9-a6a0-f06d2854ac5f

### Configure scope and register the native application

1. In the Azure AD B2C settings, go to **App registrations**.
2. Open the application created above which is currently being used by the e-Commerce rendering application.
3. In the left navigation pane under **Manage**, select **Expose an API** and verify that a `user_impersonation` scope exists. If one does not exist, select **Add a scope** to create one. When prompted for an **Application ID URI**, leave the application ID URI as is and then add "user\_impersonation" for the **Scope name**. Then enter friendly values for **Admin consent display name** and **Admin consent description**.



4. Copy and save the full scope value, as this information will be used later as the `userImpersonationScopeURL` property value in your `credentials.json` file.
5. Return to the native application you just created, and in the left navigation pane under **Manage**, select **API permissions**.
6. Select **Add a permission**, and then select the **APIs my organization uses** tab.
7. Search for your e-Commerce rendering application that was created above, and then select it and add `user_impersonation` as a permission.



8. Select **Add permissions**.
9. Select **Grant admin consent for** (this name will contain your domain), and then select **Yes** to apply the consent. You should now see a green checkmark under **Status** for `user_impersonation`.

The Azure AD setup portion is now complete and you should now have your versions of all of the following example values.

PROPERTY NAME	EXAMPLE VALUE
ropcFlowUserName	B2C_1_ROPC_Auth
loginDomain	commerceonboardingb2c.b2clogin.com
b2cTenant	commerceonboardingb2c
nativeApplicationId	25f6742d-f8b8-44d9-a6a0-f06d2854ac5f
userImpersonationScopeURL	<code>https://commerceonboardingb2c.onmicrosoft.com/b7ad3e87-d8b0-4c83-b08d-7c34c19f7933/user_impersonation</code>

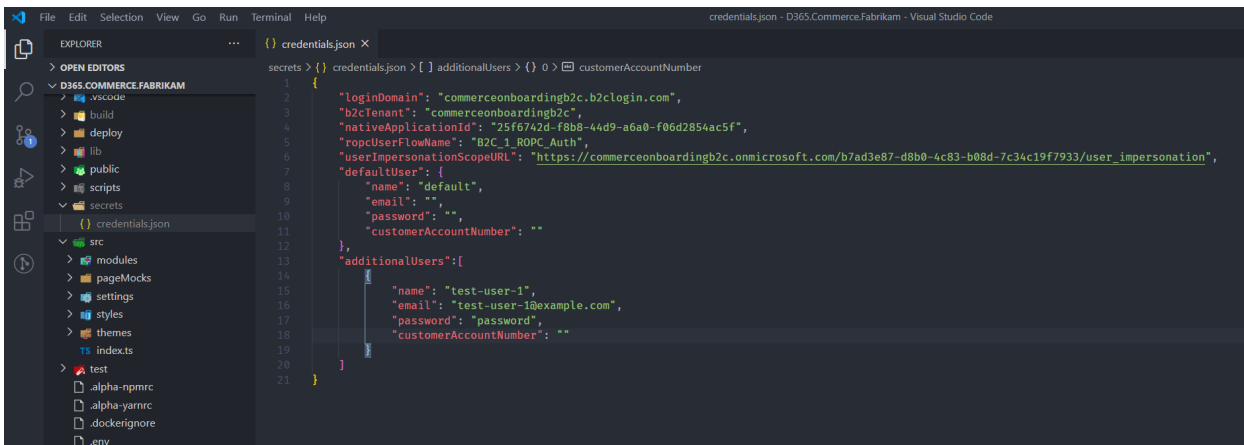


# Configure your Node application

After you have completed the steps to configure your Azure AD B2C tenant, you will need to create a credentials file in your online software development kit (SDK) Node application.

The credentials will live under the `secrets/` directory in your Node application. Create a `secrets/` directory in your application if you haven't already, and then create a new file named `credentials.json` that is similar to the following example that uses the data gathered above.

```
{
  "loginDomain": "commerceonboardingb2c.b2clogin.com",
  "b2cTenant": "commerceonboardingb2c",
  "nativeApplicationId": "25f6742d-f8b8-44d9-a6a0-f06d2854ac5f",
  "ropcUserFlowName": "B2C_1_ROPC_Auth",
  "userImpersonationScopeURL": "https://commerceonboardingb2c.onmicrosoft.com/b7ad3e87-d8b0-4c83-b08d-7c34c19f7933/user_impersonation",
  "defaultUser": {
    "name": "default",
    "email": "",
    "password": "",
    "customerAccountNumber": ""
  },
  "additionalUsers": [
    {
      "name": "test-user-1",
      "email": "test-user-1@example.com",
      "password": "password",
      "customerAccountNumber": ""
    }
  ]
}
```

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project structure with a 'secrets' directory containing 'credentials.json'. The main editor window displays the content of 'credentials.json', which matches the JSON code block shown above. The file path in the breadcrumb is 'secrets > {} credentials.json > {} additionalUsers > {} 0 > customerAccountNumber'. The code is syntax-highlighted and includes line numbers from 1 to 21.

## NOTE

Everything under the `secrets/` directory should be added to your `.gitignore` file to help prevent credentials from being leaked online.

After using the information collected in the Azure setup steps to populate your `credentials.json` file, you need to add test accounts that you want to use during local development. The accounts defined here should be valid accounts that have already been created in Dynamics 365 Commerce headquarters.

- **defaultUser:** The default user that will be used when the `mockUser` query parameter is set to `true`. The name value should be `default`.
- **additionalUsers:** An array of user objects that allows you to configure additional users to test with. Each entry in this array should be an object with a name, email address, password, and customer account number.

To sign in as one of these users, use the query parameter `mockUser= <name>`.

## Mock sign-in status

After all of the above configuration steps are complete, start your e-Commerce Node application in local dev mode using the `yarn start` command. Sign-in status is controlled using the `mockUser` query parameter and works to mock the signed-in state on mock pages as well as on published pages (for example,

`https://localhost:4000?mock=homepage&mockUser=true` OR `https://localhost:4000?mockUser=true`).

Use `mockUser= <true|false|name>` to control the signed-in behavior. The behavior of each of the query parameter values is described in the following table.

MOCKUSER VALUE	EXAMPLE	SIGN IN/SIGN OUT	DESCRIPTION
true	mockUser=true	Sign in	Signs in as the default user.
name	mockUser=test-user-1	Sign in	Signs in as the user specified in the query parameter.
false	mockUser=false	Sign out	Signs out the currently signed-in user.

You can use the `mockUser` query parameter to test pages as different users without signing out and signing back in again for each different user. For example, hitting `https://localhost:4000?mock=homepage&mockUser=true` and then `https://localhost:4000?mock=homepage&mockUser=test-user-1` would allow you to test the homepage mock as different signed-in users.

When you hit a page with `mockUser` turned on and successfully sign in, the signed-in state will persist across pages until you either sign in with a different user or sign out.

You can also make use of the sign-in and sign-out buttons on the webpage itself to mock signed-in user behavior. The sign-in button will sign you in as the default user while the sign-out button will sign out the currently signed-in user.

## Additional resources

[Request properties object](#)

[App settings](#)

[Platform settings file](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Configure module properties to be shown based on context](#)

[Globalize modules by using the CultureInfoFormatter class](#)

[Set up Azure Key Vault for secure key management](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure module properties to be shown based on context

2/18/2021 • 7 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic describes how to configure module properties so that they are shown or hidden based on the contextual values of other configuration properties.

Multiple module configuration properties can be defined in a module's definition file. However, there might be scenarios where the relevance of some property fields depends on the values that are set for other property fields of the module. Property fields that aren't relevant should be hidden to minimize the number of fields that are shown to a page editor who is configuring the module. This behavior helps reduce complexity and the possibility of confusion.

A module can use a conditional schema to define the rules that the module properties pane in Commerce site builder should follow to show or hide property fields based on the values of other property fields. For example, a module has a **layout** property that allows for two layouts, one of which has plain text, and the other of which has rich text and an image. In this case, the module designer might want to ensure that only property fields that are appropriate to the context (that is, the layout) are shown in site builder when a page editor configures the module.

The ability to show or hide property fields based on context is supported in module definition and module definition extension files that use the **dependentSchemas** property for conditional schemas. Two types of conditional schema are supported: *schema dependencies* and *property dependencies*.

## Schema dependencies

Schema dependencies can be used to declare that the schema should change when a specific value is selected for a configuration property. The **oneOf** property is used with the **dependentSchemas** property to declare the list of configuration properties that are applicable to a specific configuration value.

### Schema dependencies example

As the following example of a module definition file shows, when the **layout** property is set to **plainTextOnly**, the **featureText** property should be shown. Alternatively, when the **layout** property is set to **richTextWithImage**, the **featureRichText**, **featureImage**, and **imageAlignment** properties should be shown (but the **featureText** property should not be shown).

```
{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "config": {
    "layout": {
      "friendlyName": "Text Layout",
      "description": "Sets the desired text output to be plain text or rich text with images.",
      "type": "string",
      "oneOf": [
        {
          "friendlyName": "Plain Text",
          "description": "Sets the text output to be plain text.",
          "type": "string",
          "value": "plainTextOnly",
          "dependentSchemas": {
            "featureText": true
          }
        },
        {
          "friendlyName": "Rich Text with Image",
          "description": "Sets the text output to be rich text with images.",
          "type": "string",
          "value": "richTextWithImage",
          "dependentSchemas": {
            "featureRichText": true,
            "featureImage": true,
            "imageAlignment": true,
            "featureText": false
          }
        }
      ]
    }
  }
}
```

```

        "enum": {
            "plainTextOnly": "Plain Text Only",
            "richTextWithImage": "Rich Text With Image"
        },
        "default": "plainTextOnly",
        "scope": "module",
        "group": "Layout Properties"
    }
},
"dependentSchemas": {
    "oneOf": [
        {
            "properties": {
                "layout": {
                    "enum": {
                        "plainTextOnly": "plainTextOnly"
                    }
                },
                "featureText": {
                    "type": "string",
                    "friendlyName": "Feature Text",
                    "description": "Main text title to show in module."
                }
            }
        },
        {
            "properties": {
                "layout": {
                    "enum": {
                        "richTextWithImage": "richTextWithImage"
                    }
                },
                "featureRichText": {
                    "type": "richText",
                    "friendlyName": "Feature Text",
                    "description": "Main rich text to show in module."
                },
                "featureImage": {
                    "type": "image",
                    "friendlyName": "Feature Title",
                    "description": "Image to show in module."
                },
                "imageAlignment": {
                    "friendlyName": "Image Alignment",
                    "description": "Sets the desired alignment of the image, either left or right on the
text.",
                    "type": "string",
                    "enum": {
                        "left": "Left",
                        "right": "Right"
                    },
                    "default": "left"
                }
            }
        }
    ]
}
}
}

```

## Property dependencies

Property dependencies can be used to declare that specific configuration properties must be present if the value of another configuration property is present.

### Property dependencies example

In the following example, the `dependentSchemas` property specifies that whenever the `productTitle` value is

entered, the **subTitle** configuration property should be shown in site builder.

```
{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "config": {
    "productTitle": {
      "type": "string",
      "friendlyName": "Product Title",
      "description": "Product title."
    }
  },
  "dependentSchemas": {
    "productTitle": {
      "properties": {
        "subTitle" : {
          "type": "string",
          "friendlyName": "Product Sub Title",
          "description": "Product sub title."
        }
      },
      "required": ["productTitle"]
    }
  }
}
```

## Handling property override conflicts

Because the **dependentSchemas** property is supported in both module definition files and module definition extension files, there might be conflicts between the two types of files. By setting a Boolean **override** property to **true** in the module definition extension file, you can enable overrides of specific configuration properties.

The following examples show a module definition file and a module definition extension file that uses the **override** property.

### Module definition file example

```
{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "config": {
    "layout": {
      "friendlyName": "Text Layout",
      "description": "Sets the desired text output to be plain text or rich text with images.",
      "type": "string",
      "enum": {
        "plainTextOnly": "Plain Text Only",
        "richTextWithImage": "Rich Text With Image"
      },
      "default": "plainTextOnly",
      "scope": "module",
      "group": "Layout Properties"
    }
  }
}
```

### Module definition extension file example

```
{
```

```

    "$type": "definitionExtension",
    "config": {
      "layout": {
        "friendlyName": "Text Layout",
        "description": "Sets the desired text output to be plain text or rich text with images.",
        "type": "string",
        "enum": {
          "plainTextOnly": "Plain Text Only",
          "richTextOnly": "Rich Text Only",
          "richTextWithImage": "Rich Text With Image"
        },
        "default": "plainTextOnly",
        "override": true
      }
    },
    "dependentSchemas": {
      "oneOf": [
        {
          "properties": {
            "layout": {
              "enum": {
                "plainTextOnly": "plainTextOnly"
              }
            },
            "featureText": {
              "type": "string",
              "friendlyName": "Feature Text",
              "description": "Main text title to show in module.",
            }
          }
        },
        {
          "properties": {
            "layout": {
              "enum": {
                "richTextOnly": "richTextOnly"
              }
            },
            "featureRichText": {
              "type": "richText",
              "friendlyName": "Feature Text",
              "description": "Main rich text to show in module.",
            }
          }
        },
        {
          "properties": {
            "layout": {
              "enum": {
                "richTextWithImage": "richTextWithImage"
              }
            },
            "featureRichText": {
              "type": "richText",
              "friendlyName": "Feature Text",
              "description": "Main rich text to show in module.",
            },
            "featureImage": {
              "type": "image",
              "friendlyName": "Feature Title",
              "description": "Image to show in module.",
            },
            "imageAlignment": {
              "friendlyName": "Image Alignment",
              "description": "Sets the desired alignment of the image, either left or right on the
text.",
              "type": "string",
              "enum": {
                "left": "left",

```

```
    "left": "Left",
    "right": "Right"
  },
  "default": "left"
}
}
}
]
}
```

## Conflict resolution scenarios

The following tables list possible scenarios and expected outcomes when schema dependencies are used with module definition and module definition extension files.

### Regular scenarios

SCENARIO	EXPECTED OUTCOME
A schema dependency is used only in the module definition file. No conflicts exist between properties in the schema dependency and the module definition extension file.	The schema dependency is applied.
A schema dependency is used only in the module definition extension file. No conflicts exist between properties in the schema dependency and the module definition extension file.	The schema dependency is applied.
A schema dependency is used only in the module definition file. A conflict exists between properties in the schema dependency and the module definition extension file. For example, property A is declared both in the schema dependency of the module definition file and in the module definition extension file, which doesn't have a schema dependency.	A build error occurs.
A schema dependency on the same property is used both in the module definition file and in the module definition extension file.	The module definition file takes precedence.
The same property is defined both in the module definition file and in the module definition extension file.	The module definition file takes precedence.

### Override scenarios

SCENARIO	EXPECTED OUTCOME
The same property is defined both in the module definition file and in the module definition extension file. Either no <b>override</b> property is set for the property in the module definition extension file, or the <b>override</b> property is set to <b>false</b> .	The module definition file takes precedence.
A schema dependency on the same property is used both in the module definition file and in the module definition extension file. The <b>override</b> property is set to <b>true</b> for the property in the module definition extension file.	The module definition extension file takes precedence.



SCENARIO	EXPECTED OUTCOME
A schema dependency on the same property is used both in the module definition file and in the module definition extension file. Either no <b>override</b> property is set for the property in the module definition extension file, or the <b>override</b> property is set to <b>false</b> .	The module definition file takes precedence.
The same property is defined both in the module definition file and in the schema dependency of the module definition extension file. The <b>override</b> property is set to <b>true</b> for the property in the module definition extension file.	The module definition extension file takes precedence.
The same property is defined both in the module definition file and in the schema dependency of the module definition extension file. Either no <b>override</b> property is set for the property in the module definition extension file, or the <b>override</b> property is set to <b>false</b> .	The module definition file takes precedence.

## Additional resources

[Request properties object](#)

[App settings](#)

[Platform settings file](#)

[Module definition file](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Add module configuration fields](#)

[Extend a theme to add module extensions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Globalize modules by using the CultureInfoFormatter class

2/18/2021 • 10 minutes to read • [Edit Online](#)

This topic describes how to globalize modules by using the **CultureInfoFormatter** class.

## Overview

Globalization of an online site should include not only string localization, but also number, date, and currency formatting for the various languages and regions that your website serves.

The Microsoft Dynamics 365 Commerce Online Software Development Kit (SDK) provides a **CultureInfoFormatter** class that helps meet typical globalization requirements for the format of numbers, currencies as well as dates and times.

## Access the CultureInfoFormatter class in a module view file

An instance of the **CultureInfoFormatter** class is automatically created and can be accessed in the module view through the `this.props.context` object. The following examples shows how to use the **CultureInfoFormatter** methods to format currency.

```
public render(): JSX.Element | null {
  ...
  // this.props.context.cultureFormatter contains an initialized formatter
  // with locale of 'en-US' in this example
  const intlFormatter = this.props.context.cultureFormatter
  intlFormatter.formatCurrency(34.12);
  // expected output: $ 34.12
}
```

## Construct an instance of the CultureInfoFormatter class

The constructor of the **CultureInfoFormatter** class takes one argument, **lang-locale**.

The **lang-locale** argument must be a valid [BCP-47](#) language tag. A default value, 'en-US', is used if no language tag is specified. Language tags aren't case-sensitive. However, by convention, but the locale is usually capitalized.

### Example

```
import {CultureInfoFormatter} from '@msdyn365-commerce/core';

// Default constructor will use 'en-US' for locale
let intlFormatter = new CultureInfoFormatter();

// Constructs a new intl formatter using the French language as spoken in France
intlFormatter = new CultureInfoFormatter('fr-FR');

// Constructs a new intl formatter using the English language as spoken in Great Britain
intlFormatter = new CultureInfoFormatter('en-GB');
```

# CultureInfoFormatter class formatting functions

The `CultureInfoFormatter` class provides the following formatting functions:

- Currency formatting
- Date formatting
- Time formatting
- Number formatting

## Currency formatting

To format currency according to the conventions for a specific locale, use the `formatCurrency()` method as shown the following example.

```
/**
 * Returns a localized currency formatted version of a price.
 *
 * @param price Either a string or number representing the price that will be localized and formatted
 * @param currencyCode Optional argument. The three letter currency code that will be used for formatting
 the currency.
 * If the currency code is not provided the locale will be used to determine the best fit currency code.
 */
formatCurrency(price: string | number, currencyCode?: string): string;
```

The `currencyCode` argument is optional. If it's provided, it must be in [ISO 4217](#) format. If the `currencyCode` argument isn't provided, the formatter uses the locale to determine the best currency code to use.

## Currency formatting examples

```
import {CultureInfoFormatter} from '@msdyn365-commerce/core';

// Set locale to fr-FR
let cultureInfoFormatter = new CultureInfoFormatter('fr-FR');

// Using a string argument for the price
cultureInfoFormatter.formatCurrency('34.12', 'eur');
// expected output: "34,12 €"

// Using a number argument for the price and letting the formatter find
// the best currency code to use from the locale given
cultureInfoFormatter.formatCurrency(34.12);
// expected output: "34,12 €"

cultureInfoFormatter = new CultureInfoFormatter('en-IN');
cultureInfoFormatter.formatCurrency(34.12, 'inr');
// expected output: ₹ 34.12
```

## Date formatting

To format a date according to the conventions for a specific locale, use the `formatDate()` method as shown the following examples.

### SDK version 1.27.7 and greater

```
/**
 * Returns a localized formatted version of a date
 *
 * @param date Date object or valid date string representing the date that will be localized and formatted
 * @param options An optional argument that controls the formatting.
 */
public formatDate = (date: Date | string, options?: IDateFormatOptions): string
```

### SDK versions earlier than 1.27.7

```
/**
 * Returns a localized formatted version of a date
 *
 * @param date Date object representing the date that will be localized and formatted
 * @param options An optional argument that controls the formatting.
 */
formatDate(date: Date, options?: IDateFormatOptions): string;
```

The **options** argument is optional. It lets you control the localization and formatting. For more information about date formatting properties, see [IDateFormatOptions](#).

### Date formatting examples

```
import {CultureInfoFormatter} from '@msdyn365-commerce/core';
const testDate = new Date(2012, 11, 20, 3, 0, 0); // 12/20/2012 (in US format-mm/dd/yyyy)
let cultureInfoFormatter = new CultureInfoFormatter('en-US');

// Basic format with no options
cultureInfoFormatter.formatDate(testDate);
// expected output: "12/20/2012"

// Set lang-locale to English as spoken in Great Britain
cultureInfoFormatter = new CultureInfoFormatter('en-GB');
cultureInfoFormatter.formatDate(testDate);
// expected output: "20/12/2012"

// Set lang-locale to German as spoken in Germany
cultureInfoFormatter = new CultureInfoFormatter('de-DE');

let options: IDateFormatOptions = <IDateFormatOptions>{};
options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };

cultureInfoFormatter.formatDate(testDate);
// expected output: "Donnerstag, 20. Dezember 2012"
```

### Time formatting

To format a time according to the conventions for a specific locale, use the **formatTime()** method as shown the following examples.

### SDK version 1.27.7 and greater

```
/**
 * Returns a localized formatted version of a time
 *
 * @param time Date object or valid date string representing the time that will be localized and formatted
 * @param options An optional argument that controls the formatting
 */
public formatTime = (time: Date | string, options?: ITimeFormatOptions): string;
```

### SDK versions earlier than 1.27.7

```
/**
 * Returns a localized formatted version of a time
 *
 * @param time Date object representing the time that will be localized and formatted
 * @param options An optional argument that controls the formatting
 */
formatTime(time: Date, options?: ITimeFormatOptions): string;
```

The **options** argument is optional. It lets you control the localization and formatting. For more information

about time formatting properties, see [ITimeFormatOptions](#).

### Time formatting examples

```
import {CultureInfoFormatter} from '@msdyn365-commerce/core';
const testDate = new Date(2012, 11, 20, 13, 34, 23); // 1:34:23 PM in en-US Format
let cultureInfoFormatter = new CultureInfoFormatter('en-US');
let options: ITimeFormatOptions = <ITimeFormatOptions>{};
options.hour12 = false;
// Format time with 24 hour time
cultureInfoFormatter.formatTime(testDate, options);

cultureInfoFormatter = new CultureInfoFormatter('fr-FR');
options = <ITimeFormatOptions>{};
options.second = 'numeric';
cultureInfoFormatter.formatTime(testDate, options);
// expected output: "13:34:23"
```

### Number formatting

To format a number according to the conventions for a specific locale, use the `formatNumber()` method as shown in the following example.

```
/**
 * Returns a localized formatted version of a number
 *
 * @param value The number that will be localized and formatted
 * @param options An optional argument that controls the formatting.
 */
formatNumber(value: number, options?: INumberFormatOptions): string;
```

The `options` argument is optional. It lets you control the localization and formatting. For more information about number formatting properties, see [INumberFormatOptions](#).

### Number formatting examples

```
import {CultureInfoFormatter} from '@msdyn365-commerce/core';

let cultureInfoFormatter = new CultureInfoFormatter('en-US');
cultureInfoFormatter.formatNumber(123456789);
// expected output: "123,456,789"
cultureInfoFormatter.formatNumber(1234567.89);
// expected output: "1,234,567.89"

// German language uses comma as decimal separator and period for thousands
cultureInfoFormatter = new CultureInfoFormatter('de-DE');
cultureInfoFormatter.formatNumber(1234567.89);
// expected output: "1.234.567,89"

const options: INumberFormatOptions = <INumberFormatOptions>{};
options.style = 'percent';
(cultureInfoFormatter.formatNumber(0.7842, options));
// expected output: "78,42 %"

// Setting the language to Arabic formats numbers using Arabic numerals
cultureInfoFormatter = new CultureInfoFormatter('ar-EG');
cultureInfoFormatter.formatNumber(1234567.89);
// expected output: "١,٢٣٤,٥٦٧,٨٩"
```

## Formatting options

This section covers the formatting options and property details for `ITimeFormatOptions`,

`IDateFormatOptions`, and `INumberFormatOptions` interfaces.

## `ITimeFormatOptions`

```
interface ITimeFormatOptions {
  localeMatcher?: 'best fit' | 'lookup';
  formatMatcher?: 'basic' | 'best fit';
  hourCycle?: 'h11' | 'h12' | 'h23' | 'h24';
  timeZone?: string;
  hour12?: boolean;
  hour?: 'numeric' | '2-digit';
  minute?: 'numeric' | '2-digit';
  second?: 'numeric' | '2-digit';
  timeZoneName?: 'short' | 'long';
}
```

### Property details

NAME	TYPE	ALLOWED VALUES	DESCRIPTION
<code>localeMatcher</code>	enum	'best fit' or 'lookup'	Specify the algorithm that is used to match and find the locale. The <b>'lookup'</b> matcher follows the lookup algorithm that is specified in <a href="#">BCP-47</a> . The <b>'best fit'</b> matcher lets the runtime provide a locale that is at least as well-suited to the request as the result of the lookup algorithm, although it might be better suited than that result.
<code>formatMatcher</code>	enum	'basic' or 'best-fit'	Specify the format matching algorithm that is used, and when it's used. The default value is <b>'best fit'</b> .
<code>hourCycle</code>	enum	'h11', 'h12', 'h23', or 'h24'	Specify the hour cycle that is used.
<code>timeZone</code>	string	Time zone names, as defined by the <a href="#">IANA time zone database</a>	Specify the time zone that is used. The default value is the runtime's default time zone. If the environment lacks native internationalization support and internationalization is polyfilled, this option won't be supported (because internationalization polyfill doesn't support time zones).

NAME	TYPE	ALLOWED VALUES	DESCRIPTION
hour12	boolean	'true' or 'false'	Specify whether 12-hour time is used instead of 24-hour time. This option overrides the <b>hourCycle</b> property, and its default value depends on the locale.
hour	enum	'numeric' or '2-digit'	Specify the representation of the hour. The <b>'2-digit'</b> value forces hours to be shown as two digits.
minute	enum	'numeric' or '2-digit'	Specify the representation of the minute. The <b>'2-digit'</b> value forces minutes to be shown as two digits.
second	enum	'numeric' or '2-digit'	Specify the representation of the second. The <b>'2-digit'</b> value forces seconds to be shown as two digits.
timeZoneName	enum	'short' or 'long'	Specify the representation of the time zone name. The <b>'short'</b> value shows the three-character abbreviation for the time zone. The <b>'long'</b> value shows the full name.

## IDateFormatOptions

```
interface IDateFormatOptions extends ITimeFormatOptions {
    weekday?: 'narrow' | 'short' | 'long';
    year: 'numeric' | '2-digit';
    month?: 'numeric' | '2-digit' | 'narrow' | 'short' | 'long';
    day?: 'numeric' | '2-digit';
}
```

### NOTE

All properties provided in `ITimeFormatOptions` interface can be used in `IDateFormatOptions` interface since the time object is a subcomponent of the date object.

### Property details

NAME	TYPE	ALLOWED VALUES	DESCRIPTION
------	------	----------------	-------------

NAME	TYPE	ALLOWED VALUES	DESCRIPTION
weekday	enum	'narrow', 'short', or 'long'	Specify the representation of the day. The <b>'narrow'</b> value shows the one-character or two-character representation of the day. The <b>'short'</b> value shows the three-character representation. The <b>'long'</b> value shows the full name.
year	enum	'numeric' or '2-digit'	Specify the representation of the year. The <b>'2-digit'</b> value shows the two most significant digits of the year (for example, the year 2018 is shown as <b>18</b> ). The <b>'numeric'</b> value shows the whole year.
month	enum	'numeric', '2-digit', 'narrow', 'short', or 'long'	Specify the representation of the month. The <b>'numeric'</b> value shows the numeric representation of the month (for example, April is shown as <b>4</b> ). The <b>'2-digit'</b> value shows the two-digit numeric representation (for example, April is shown as <b>04</b> ). The <b>'narrow'</b> value shows the two-character representation (for example, April is shown as <b>AP</b> ). The <b>'short'</b> value shows the three-character representation (for example, April is shown as <b>Apr</b> ). The <b>'long'</b> value shows the full name.
day	enum	'numeric' or '2-digit'	Specify the representation of the day. The <b>'2-digit'</b> value forces days to be shown as two digits.

No default value is defined for each date/time component property. However, if all component properties are undefined, the **'numeric'** value is assumed for the **year**, **month**, and **day** property.

### **INumberFormatOptions**



```

export interface INumberFormatOptions {
  localeMatcher?: 'best fit' | 'lookup';
  style?: 'decimal' | 'percent' | 'currency';
  currency?: string;
  minimumIntegerDigits?: number;
  minimumFractionDigits?: number;
  maximumFractionDigits?: number;
  minimumSignificantDigits?: number;
  maximumSignificantDigits?: number;
}

```

### Property details

NAME	TYPE	ALLOWED VALUES	DESCRIPTION
localeMatcher	enum	'best fit' or 'lookup'	Specify the algorithm that is used to match and find the locale. The <b>'lookup'</b> matcher follows the lookup algorithm that is specified in <a href="#">BCP-47</a> . The <b>'best fit'</b> matcher lets the runtime provide a locale that is at least as well-suited to the request as the result of the lookup algorithm, although it might better suited than that result.
style	enum	'decimal', 'percent', or 'currency'	Specify the formatting style that is used. Use the <b>'decimal'</b> value for plain number formatting, the <b>'currency'</b> value for currency formatting, and the <b>'percent'</b> value for percentage formatting. The default value is <b>'decimal'</b> .
currency	string	Three-character <a href="#">ISO 4217</a> currency codes	Specify the currency that is used in currency formatting. There is no default value. If the <b>style</b> property is set to <b>'currency'</b> , the <b>currency</b> property must be provided.
minimumIntegerDigits	number	Any integer value between 1 and 21	Specify the minimum number of integer digits that is used. The default value is 1.
minimumFractionDigits	number	Any integer value between 0 and 20	Specify the minimum number of fraction digits that is used. The default value for number and percentage formatting is 0 (zero). The default value for currency formatting is provided by the <a href="#">ISO 4217</a> standard.

NAME	TYPE	ALLOWED VALUES	DESCRIPTION
maximumFractionDigits	number	Any integer value between 0 and 20	Specify the maximum number of fraction digits that is used.
minimumSignificantDigits	number	Any integer value between 1 and 21	Specify the minimum number of significant digits that is used. The default value is 1.
maximumSignificantDigits	number	Any integer value between 1 and 21	Specify the maximum number of significant digits that is used. The default value is 21.

## Additional resources

[Request properties object](#)

[App settings](#)

[Platform settings file](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Mock the signed-in state during local development](#)

[Configure module properties to be shown based on context](#)

[Set up Azure Key Vault for secure key management](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up Azure Key Vault for secure key management

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic describes how to set up Azure Key Vault to provide secure key management in Dynamics 365 Commerce.

## Overview

Some Dynamics 365 Commerce e-Commerce development scenarios require business-sensitive data such as credentials or access tokens that must be stored securely. [Azure Key Vault](#) provides the capability to import, store, and manage cryptographic keys and certificates that can be securely accessed as needed.

This topic shows how to do the following:

- Create a Key Vault to securely store sensitive information.
- Configure your e-Commerce site to securely communicate with Retail Server.
- Set up Retail Server to securely communicate with your Key Vault.
- Access secret values from within your e-Commerce components.

## Create a Key Vault to store application secrets

The first step is to create a new Key Vault to store application secrets. You will need an Azure account to access Azure Key Vault.

To create a new Key Vault, follow these steps.

1. Navigate to your [Azure Portal homepage](#).
2. Select **Create a resource**.
3. Search for **Key Vault**, and then select **Create**.
4. Select the subscription and resource group you would like this Key Vault to be a part of.
5. Enter a name, region, and pricing tier for your Key Vault.
6. Select **Next**, and then create access policies and assign permissions to users. Leave all other settings as is.
7. Select **Review + create**.
8. After you have reviewed the configuration, select **Create** and then wait for the deployment to complete.
9. After the Key Vault has successfully been deployed, you can add any secrets under **Secrets**.

## Configure server-to-server authentication between the e-Commerce Node application and Retail Server

Next, the e-Commerce Node application needs to be configured to securely communicate with Retail Server.

For the following steps, you will need to have the tenant ID of the Azure App Service hosting your Node application, as well as the client ID of the managed identity tied to your Azure App Service. If you do not have access to Azure App Service, work with your service integrator or support team to obtain the required information.

### Find and copy your tenant ID

To find your tenant ID, follow these steps.

1. Navigate to your [Azure Portal homepage](#).

2. Go to the directory containing the Azure App Service hosting your Node application.
3. Go to **Azure Active Directory**.
4. In the left navigation pane, select **Properties**.
5. Find and copy the **Tenant ID**.

### Find and copy the client ID of the managed identity for your Node application

To find the client ID of the managed identity for your Node application, follow these steps.

1. Navigate to your [Azure Portal homepage](#).
2. Go to the directory containing the Azure App Service hosting your Node application.
3. In the left navigation pane, select **Identity**.
4. Select the **User assigned** tab.
5. Select the managed identity resource for your Node application.
6. In the left navigation pane, select **Overview**, and then copy the **Client ID** value.

The following example image highlights the **Client ID** value on the Azure Portal **Overview** page.



### Add your Node application details into Retail Server's authentication allow list

To add your Node application details into Retail Server's authentication allow list, follow these steps.

1. In Commerce headquarters, go to **Commerce Shared Parameters**.
2. Select the **Identity Providers** tab.
3. In the first section named **IDENTITY PROVIDERS**, select **Add**.
4. For **Issuer Value**, enter `https://sts.windows.net/<TENANT_ID>/`, where **<TENANT\_ID>** is your tenant ID.
5. For **Name**, enter "Azure AD".
6. For **Type**, enter "Active Directory".
7. In the second section named **RELYING PARTIES**, add an entry with the client ID of your managed identity.
8. For **Type**, select **Confidential**.
9. For **UserType**, select **Application**.
10. Provide a name for your Node application.
11. In the third section named **SERVER RESOURCE IDS**, add an entry with the server resource ID of `https://commerce.dynamics.com`, and then select **Save**.

You should now have a configuration similar to the one in the following example.

## Set up Identity Providers to be used by Retail Server

Identity Provider	Provider Type	Provider Name
https://localhost:44336/core	Commerce Identity Provider	Open ID Connect
https://sts.windows.net/[redacted]	Azure AD	Azure Active Directory

**RELYING PARTIES**

[Add](#) [Remove](#)

ClientId	Type	UserType
93b1ebe5-969a-4cb5-9d48-7b928ff2655e	Confidential	Application
d5527362-3bc8-4e63-b5b3-606dc14747e9	Public	Worker
d6b5a0bd-bf3f-4a8c-b370-619fb3d0e1cc	Public	Worker

**SERVER RESOURCE IDS**

[Add](#) [Remove](#)

Server Resource Id

https://commerce.dynamics.com

To synchronize the changes into the channel database, in Commerce headquarters go to **Distribution schedule** and execute job 1110 (Global configuration), as shown in the following image.

Finance and Operations | Distribution schedule

1110 Global configuration

Name	Description	Current batch job	Current batch status
1110	Global configuration		N/A

**General**

Direction of data flow: Download  Active  Yes

**Channel database groups**

+ Add - Remove

✓ Channel database group

Default

Your Node application will now be able to securely communicate and request Key Vault secrets from Retail Server.

## Add Key Vault details to Retail Server

Next, you will configure Retail Server to securely communicate with your Key Vault.

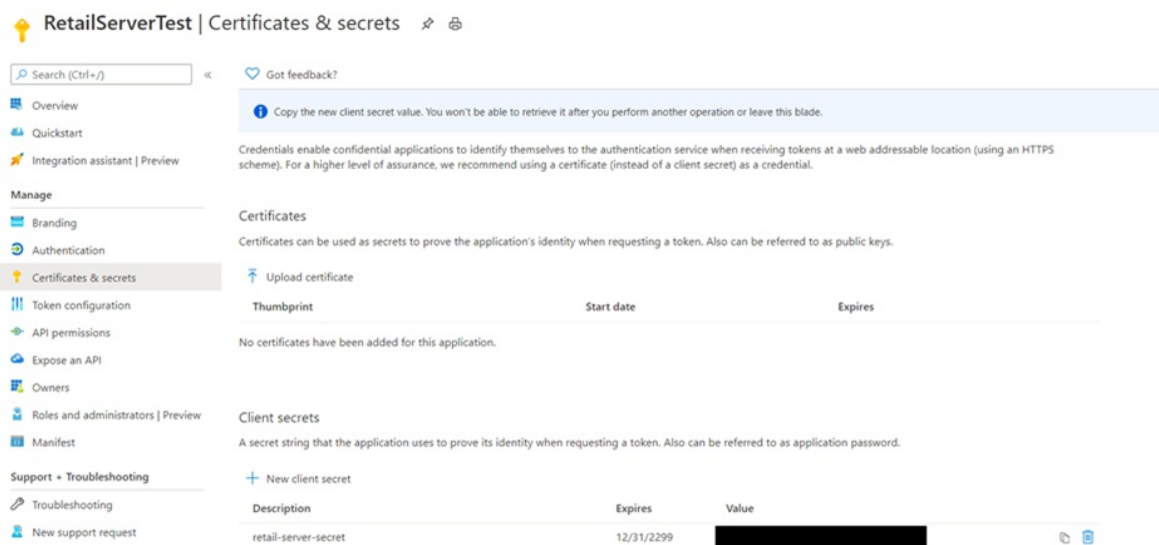
### Create a new app registration and add a client secret

You will first need to create a new app registration under Azure AD to represent your Retail Server so that you can register your Key Vault with Retail Server.

To create a new app registration, follow these steps.

1. Navigate to your [Azure Portal homepage](#).
2. Go to the directory containing the Azure App Service hosting your Node application.
3. Go to **Azure Active Directory**.

- In the left navigation pane, select **App registrations**.
- Select **New registration**, and then enter a name (for example, "RetailServer").
- In the overview panel, copy and save the **Application (client) ID** value for later use.
- Select **Certificates & secrets**, and then under **Client secrets** select **New client secret**.
- In the **Add a client secret** dialog box, enter a description under **Description** (for example, "retail-server-secret").
- Under **Expires**, select **Never**, and then select **OK**.
- On the **Certificates & secrets** page, copy the secret value In the **Value** box of your new client secret and store it in a safe place. This secret value is what will enable communication between Retail Server and your Key Vault.



#### NOTE

You will only have one opportunity to copy the secret value so it's important to do so now.

### Add an access policy in your Key Vault

Next, to add an access policy in your Key Vault, follow these steps.

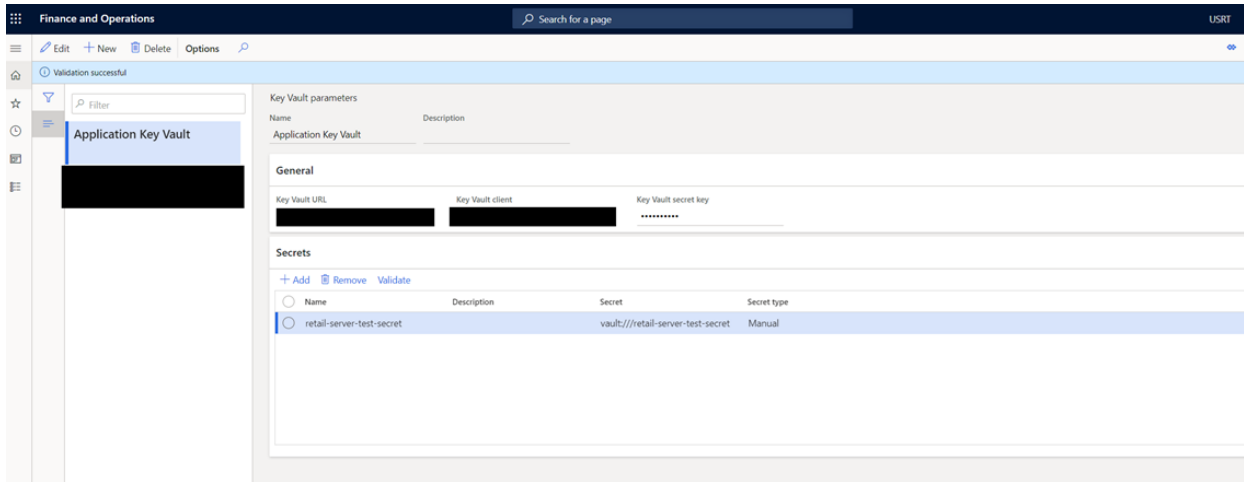
- Go to your Key Vault.
- Select **Access policies**, and then select **Add Access Policy**.
- For the first option, select **Key, Secret, and Certificate Management**.
- For **Select principal**, search for and select the app you registered earlier (for example, "RetailServer").
- Leave **Authorized application** blank.
- Select **Save**.

### Add the Key Vault details in Retail Server

Next, to add the Key Vault details in Retail Server, follow these steps.

- In Commerce headquarters, go to **Key Vault parameters**.
- From the store selector on the top right, select **USRT**.
- Select **New**, and then enter a name to represent your Key Vault.

4. Enter the **Key Vault URL**. This is the **Vault URI** value listed on your Key Vault's **Overview** page.
5. Enter the **Key Vault client**. This is the application ID of the registered app.
6. Enter the **Key Vault secret key**. This is the secret value saved from the app registration process.
7. Add the secrets you want to access from Retail Server. For example, if the secret name is "retail-server-test-secret", add "retail-server-test-secret" as the **Name**, and "vault:///retail-server-test-secret" as the **Secret**. The **Secret type** should be set to **Manual**.
8. Select **Validate** to test your configuration. If everything was configured correctly, you will see a message that says, "Validation successful."



### IMPORTANT

The Key Vault must be configured in the USRT store.

## Access secret values within your e-Commerce Node application

After the configuration steps above are complete, you will be able to access the secret values from within your e-Commerce Node application using the `SecretManager` class. This class is initialized on the global `msdyn365Commerce` object and implements the interface shown in the following example. Along with the `secretKey`, the `baseURL` for your Retail Server needs to be passed in as a second argument. This base URL can be found in the `RequestContext` API under the `requestContext.apiSettings.baseUr1` API, which is accessed through the action context inside of data actions or the `props.context` context object in modules.

```
export interface ISecretManager {
  getSecretValue(secretKey: string, baseRetailServerUr1: string): Promise<ISecretValue>;
}

export interface ISecretValue {
  value: string; // secret value
  id?: string; // secret ID (if applicable)
  error?: Error; // Error details (set if request to fetch secret value failed)
  expiresOn: number; // Unix timestamp (in seconds) when the secret value will expire
}
```

To import this into your code, add `msdyn365Commerce` to the import statement for the '@msdyn365-commerce/core' library as shown in the following example.

```
import msdyn365Commerce, { IRequestContext, ... } from '@msdyn365-commerce/core';
```

The following example shows how to use the `SecretManager` class to access the secret value:

```
const secretValue: ISecretValue | undefined = await
msdyn365Commerce.secretManager?.getSecretValue('secretKey', requestContext.apiSettings.baseUr1);
```

Note that `secretManager` is a nullable property on `msdyn365Commerce` because the `SecretManager` class can only fetch secrets when running server-side. This is to prevent leaking the secret value to your browser. If the `secretManager` property is undefined, it means that the code is running in the context of a browser (client-side).

As a developer, you should only use the `SecretManager` class on code that is ensured to run server-side (for example, data actions that will run only on server-side), because you will otherwise be unable to access the secret value. If you include this property in code that will run in both contexts (server-side and client-side), it is important to include a fallback option if the `secretManager` property is undefined.

If the request to fetch the secret value fails, the error property will be set. You can use this to debug any issues you may encounter when trying to fetch secret values.

## Local development

Your e-Commerce Node application is only able to communicate with Retail Server and request tokens need to securely communicate in a deployed App Service environment. This means that when developing locally, the `SecretManager` class will be unable to retrieve secrets from your Key Vault. Instead you can create a secrets directory in your Node application and add a `secrets.json` file, where you can configure `secretKeys` and `secretValues` that will only be used when developing locally.

### NOTE

Everything under the `secrets/` directory should be added to your `.gitignore` file to help prevent secrets from being leaked online.

The following example shows the contents of a "secrets/secrets.json" file.

```
{
  "secretKey": "secretValue!"
}
```

If you do not configure this file, during local development you may encounter errors as the `SecretManager` class attempts to communicate with your Key Vault but fails to do so.

## Additional resources

[Request properties object](#)

[App settings](#)

[Platform settings file](#)

[Extend a module definition file](#)

[Cookie API overview](#)

[Interactive components overview](#)

[Mock the signed-in state during local development](#)

[Configure module properties to be shown based on context](#)

[Globalize modules by using the CultureInfoFormatter class](#)



**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Data actions

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic covers data actions in Microsoft Dynamics 365 Commerce.

## Overview

Data actions are JavaScript functions that are used in the Dynamics 365 Commerce architecture to help fetch and map data that is required by modules across applications.

Data actions offer improved performance through the following features:

- Integrated application-level and request-level caches enable state sharing scenarios.
- Built-in utilities support batching to minimize the number of external requests that your application requires.
- Automatic deduplication helps guarantee that multiple data action calls aren't duplicated.

The Dynamics 365 Commerce platform includes a set of core data actions that can be called from modules to do typical data retrieval. For example, core data actions can return product details. You can also create custom data actions to fetch and process data that is required by modules.

## Anatomy of a data action

Here is an example of a template TypeScript file that is created for a new data action.

```

import * as Msdyn365 from '@msdyn365-commerce/core';

/**
 * GetProductReviews Input Action
 */

export class GetProductReviewsInput extends Msdyn365.CommerceEntityInput implements Msdyn365.IActionInput {

    // TODO: Determine if the results of this get action should cache the results and if so provide
    // a cache object type and an appropriate cache key
    constructor() {
        super({shouldCacheOutput: true, cacheObjectType: 'TODO', cacheKey: 'TODO'});
    }

    public getCacheKey = () => `TODO`;
    public getCacheObjectType = () => 'TODO';
    public dataCacheType = (): Msdyn365.CacheType => 'application';
}

// TODO: Create a data model here or import one to capture the response of the action
export interface IGetProductReviewsData {
    text: string;
}

/**
 * TODO: Use this function to create the input required to make the action call
 */
const createInput = (args: Msdyn365.ICreateActionContext): Msdyn365.IActionInput => {
    return new GetProductReviewsInput();
};

/**
 * TODO: Use this function to call your action and process the results as needed
 */
async function action(input: GetProductReviewsInput, ctx:
Msdyn365.IActionContext): Promise<IGetProductReviewsData> {
    // const apiSettings = Msdyn365.msdyn365Commerce.apiSettings;

    // TODO: Uncomment the below line to get the value from a service
    // const response = await Msdyn365.sendRequest<IGetProductReviewsData[]>('/get/example/id/1', 'get');
    return {text: 'Static data from action'};
}

export const IGetProductReviewsAction = Msdyn365.createObservableDataAction({
    action: <Msdyn365.IAction<IGetProductReviewsData>>action,
    input: createInput
});

```

### Key parts of a data action

- **Action function** – The main function that contains the logic that is run when the action is called. This function might involve making application programming interface (API) calls, reading cookies, or transforming data that was passed in.

```

async function action(input:GetProductReviewsInput, ctx:
Msdyn365.IActionContext):Promise<IGetProductReviewsData> {
    // const apiSettings = Msdyn365.msdyn365Commerce.apiSettings;

    // TODO: Uncomment the below line to get the value from a service
    // const response = await Msdyn365.sendRequest<IGetProductReviewsData[]>('/get/example/id/1',
'get');
    return {text: 'Static data from action'};
}

// TODO: Create a data model here or import one to capture the response of the action
export interface IGetProductReviewsData {
    text: string;
}

```

- **Action input class** – The class that is used to pass data into the action function. The "cacheObjectType" and "cacheKey" values indicate where in the cache the class should put the result of the action.

```

export class GetProductReviewsInput extends Msdyn365.CommerceEntityInput implements
Msdyn365.IActionInput {

    // TODO: Determine if the results of this get action should cache the results and if so provide
    // a cache object type and an appropriate cache key
    constructor() {
        super({shouldCacheOutput: true, cacheObjectType: 'TODO', cacheKey: 'TODO'});
    }

    public getCacheKey = () => `TODO`;
    public getCacheObjectType = () => 'TODO';
    public dataCacheType = (): Msdyn365.CacheType => 'application';
}

```

- **createInput method** – This optional method can be used to build an instance of an action input class that is used to load data when a page is first populated.

```

const createInput = (args: Msdyn365.ICreateActionContext): Msdyn365.IActionInput => {
    return new GetProductReviewsInput();
};

```

### Create a new custom data action

To create a new custom data action, follow this step.

- At a command prompt, go to your root software development kit (SDK) folder, and run the **yarn msdyn365 add-data-action DATA\_ACTION\_NAME** command-line interface (CLI) command to create a data action, as shown in the following example.

```

c:\repos\Msdyn365.Commerce.Online>yarn msdyn365 add-data-action get-product-reviews

```

TypeScript files for new custom data actions are created under the `\src\actions\` directory. The name of each file uses the name of the data action. For the preceding example, the path and name of the file for the new custom data action are `\src\actions\getProductReviews.ts`.

## Additional resources

[Data action cache options](#)

[Test data actions with mocks](#)

[Page load data actions](#)

[Event-based data actions](#)

[Core data actions](#)

[Call Retail Server APIs](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Data action cache options

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic provides an overview of supported data action cache options in Dynamics 365 Commerce.

The Dynamics 365 Commerce online software development kit (SDK) supports caching entities at the application level, which enables caching of data action responses to improve rendering performance and reduce server load. For more information, see [Data actions overview](#).

## Caching a data action

A data action can save data in either the request or application cache, based on its action input. The **dataCacheType** property of `IActionInput` enables the underlying action runtime to make this decision on the action's behalf. By default, the action responses go into the request cache.

```
export class FullProductInput implements IActionInput {
  public productId: number;
  public channelId: number;
  public ProductDetailsCriteria: ProductDetailsCriteria;

  constructor(productId: number | string, channelId: number, criteria: ProductDetailsCriteria) {
    this.productId = typeof productId === 'string' ? parseInt(productId, 10) : productId;
    this.channelId = channelId;
    this.ProductDetailsCriteria = criteria;
  }

  public getCacheKey = () => `${this.productId.toString()}-${this.channelId.toString()}-${this.ProductDetailsCriteria}`;
  public getCacheObjectType = () => 'FullProduct';
  public dataCacheType = (): CacheType => 'request';
}
```

## Supported cache types

The following cache types are supported and can be set on the **dataCacheType** property:

- **request**: Action input caches the entity for the life cycle of the request. All of the subsequent data action inputs with the same cachekey (within the same request) will be served from the request cache.
- **application**: Action input caches the entity for the life cycle of the application (subject to time to refresh (TTR) and time to live (TTL) values as defined in cache settings. All of the subsequent data action inputs with the same cachekey will be served from the application cache.
- **instance**: Instance is a special cache type setting primarily used for aggregator data actions that do not make a request and extract information from other data actions, for example a categories hierarchy. Such data actions are run on server and client independently. If the instance is not specified, then such aggregator data actions would contain duplicate data.
- **none**: Used to skip or bypass the request cache. An action with cache type "none" skips the caching layer completely.

## Cache settings

Cache settings can be configured using the `\src\settings\cache-settings.json` file. The following example shows the format of a `cache-settings.json` file.

```
{
  "checkPeriodInSeconds": 120,
  "ttlInSeconds": {
    "entity-with-ttl": 4,
    "category": 60
  },
  "ttrInSeconds": {
    "entity-with-ttr": 2
  }
}
```

For more information, see [Data action cache settings](#).

#### **checkPeriodInSeconds**

This setting represents the time interval in seconds, at which the SDK will do a pass on the application cache to nudge TTL expired items from the cache. The default value is 600 seconds.

#### **ttlInSeconds - TTL (Time to Live)**

This setting indicates the duration an entity can stay in the application cache. This value is configurable at the application level and also per entity. The default value is 600 seconds.

#### **ttrInSeconds - TTR (Time to Refresh)**

TTR tells the action runtime to keep the data fresh, in the sense that the cache value is retained in the application cache but a fresh call is made to update the cache data. In the event of an unexpected error, you can still choose to return the data in the cache. The default value is 60 seconds.

#### **defaultTTLInSeconds**

This setting indicates the default time to live (in seconds) for all the cache items. The default value is 600 seconds.

#### **defaultTTRInSeconds**

This setting indicates the default time to refresh (in seconds) for all the cache items. The default value is 60 seconds.

## Additional resources

[Data actions overview](#)

[Test data actions with mocks](#)

[Page load data actions](#)

[Event-based data actions](#)

[Core data actions](#)

[Call Retail Server APIs](#)

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Test data actions with mocks

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to test data actions with mock data.

## Overview

By mocking data actions in Microsoft Dynamics 365 Commerce, you can replace the output of a data action with the data that is specified in the `actionmock.json` file that has been loaded. An action mock is useful if you want to test your module without invoking the actual action. You will have to use this approach if you haven't configured your Commerce server (`MSDyn365Commerce_BASEURL` property) in the `.env` file. For more information about `.env` files, see [Configure a development environment \(.env\) file](#).

## Action mock structure

To create a data action mock, create a new file under the `/src/module/<MODULE_NAME>/mocks/` directory for your module. The name of the new file should be in the format `<MODULE MOCK_NAME>.actionmock.json`, as in the following example.

```
/src/modules/product-feature/mocks/myModuleMock.actionmock.json
```

After the file is created, you can simulate the data action return object inside the file by specifying the `CacheObjectType` and `CacheKey` values that are defined in the data action. The `CacheKey` value can be set to `"*"` to accept any cache key. For more information, see [Data actions](#).

The following example shows how the `<MODULE MOCK_NAME>.actionmock.json` file should be structured.

```
{
  "CacheObjectType": "MyCacheObjectType",
  "CacheKey": "MyCacheKey",
  "Mock": {
    "foo": "bar"
  }
}
```

If no `CacheKey` value is specified, or if `"*"` is specified, all actions that have the corresponding `CacheObjectType` value will receive the mock output.

## Example

The following example shows a module definition file that uses a data action, and the corresponding data action mock that returns product data.

Example module definition file:



```

{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "categories": [
    "storytelling"
  ],
  "tags": [
    ""
  ],
  "dataActions": {
    "products": {
      "path": "@msdyn365-commerce-modules/retail-actions/dist/lib/get-simple-products",
      "runOn": "server"
    }
  },
  "config": {
    "imageAlignment": {
      "friendlyName": "Image Alignment",
      "description": "Sets the desired alignment of the image, either left or right on the text.",
      "type": "string",
      "enum": {
        "left": "Left",
        "right": "Right"
      },
      "default": "left",
      "scope": "module",
      "group": "Layout Properties"
    }
  },
  "resources": {
    "resourceKey": {
      "comment": "resource description",
      "value": "resource value"
    }
  }
}

```

Example module mock file:

```
[
  {
    "CacheObjectType": "SimpleProduct",
    "CacheKey": "*",
    "Mock": {
      "RecordId": 22565423455,
      "ItemId": "2101",
      "Name": "Retro Horn-Rimmed Keyhole Sunglasses",
      "Description": "High-quality with the perfect blend of timeless classic and modern technology
with hint of old school glamor.",
      "ProductTypeValue": 3,
      "DefaultUnitOfMeasure": "Ea",
      "BasePrice": 15,
      "Price": 15,
      "AdjustedPrice": 14,
      "MasterProductId": null,
      "Components": null,
      "Dimensions": null,
      "Behavior": null,
      "LinkedProducts": null,
      "PrimaryImageUrl": "https://bit.ly/33cMGxr",
      "ExtensionProperties": null
    }
  }
]
```

#### NOTE

The **CacheObjectType** and **CacheKey** values should match the values in the data action that you're mocking. If no **CacheKey** value is specified, all actions that have the corresponding **CacheObjectType** value receive the mock output.

## Use an action mock in a preview

To use an action mock in your module preview, include the query string parameter for the action mock **actionMock=MODULE\_NAME:MOCK\_FILE\_NAME**, as shown in the following example.

```
https://localhost:4000/modules?type=product-feature&actionMock=product-feature:myModuleMock
```

Here is the syntax of the query string parameter.

```
{module-name}:{action-mock-file-name}
```

If no action mock file name is specified, the package name is used to search for the mock.

## Additional resources

[Data actions overview](#)

[Data action cache options](#)

[Page load data actions](#)

[Event-based data actions](#)

[Core data actions](#)

[Call Retail Server APIs](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Page load data actions

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic covers page load data actions in Microsoft Dynamics 365 Commerce.

## Overview

Every page that is rendered, whether it's a product details page, a department page, or a home page, requires data. Page load data actions are used to obtain that data.

## The createInput method

When a page is loaded, a data action can be called by using the `createInput` method. The following example shows the sample code that is created in the TypeScript file for a data action by using the `yarn d365 add-data-action DATA_ACTION_NAME` command-line interface (CLI) command.

```
const createInput = (args: Msdyn365.ICreateActionContext): Msdyn365.IActionInput => {
    return new GetProductReviewsInput();
};
```

The `ICreateActionContext` type represents an object that is passed to every `createInput` method. It contains three pieces of information:

- The request context (`inputData.requestContext`)
- The module's configuration (`inputData.config`)
- The module's data (`inputData.data`)

In the following example, notice that the `createInput` method in the TypeScript template file has been added to the `createDataAction` call. Therefore, the Dynamics 365 Commerce online software development kit (SDK) can determine that this data action can be run on page load.

```
export default Msdyn365.createObservableDataAction({
    action: <Msdyn365.IAction<IGetMyDataData>>action,
    input: createInput
});
```

### Example of the createInput method

For the following example of the `createInput` method, the module that calls the data action has a configuration string property that is named `productId`. The value of this property is a product ID that the data action will use as input.

```
/**
 * Creates a ProductInput using module configuration data
 */
const createInput = (inputData: ICreateActionContext<IGeneric<IAny>>): IActionInput => {
    // Ensure module has a config property 'productId'
    if(inputData.config && inputData.config.productId) {
        // Create and return an input for the data action using the module configuration data.
        return new ProductInput(inputData.config.productId);
    }
};
```

## Register the data action on a module

Before a module can call a data action on page load, the data action must be registered in the module definition file.

The following example shows a module that uses the data action earlier in this topic to get product information so that it can be shown inside the module.

```
{
  "$type": "moduleDefinition",
  "name": "product-module",
  "friendlyName": "Product module",
  "description": "Product module",
  "categories": ["Product"],
  "dataActions": {
    "product": {
      "path": "../../actions/get-product",
      "runOn": "server"
    }
  },
  ...
}
```

Here, the **dataActions** node indicates the data actions that should be run when a module is loaded on a page. The name of every key in the data actions object refers to the name of the property that the result will be assigned to in the module's data properties (in this case, the **this.props.data.product** property). Additionally, the **path** property of every data action points to a file that exports the data action that should be run. Generally, the data action is the default export of the file it was written in.

To help ensure that you have the correct typing for the module when you develop your module view, you must update the module data file with corresponding types, based on the data actions that you've registered. Here is an example.

```
import { AsyncResult } from '@msdyn365-commerce/retail-proxy';

// product-module.data.ts
export interface IProductModuleData {
  product: AsyncResult<SimpleProduct>;
}
```

Now, when you develop this module, you will have access to the product information via the **this.props.data.product** property.

## Client-side rendering

Dynamics 365 Commerce renders pages server side and [hydrates](#) the page on the client after the initial load. Data actions defined in a module definition are executed on the server by default, which causes modules to render server side. If a module explicitly needs to render on client, the **runOn** parameter value needs to be set to 'client' in the module definition file as shown below.

```
{
  "$type": "moduleDefinition",
  "name": "product-module",
  "friendlyName": "Product module",
  "description": "Product module",
  "categories": ["Product"],
  "dataActions": {
    "product": {
      "path": "../../actions/get-product",
      "runOn": "client"
    }
  },
  ...
}
```

#### NOTE

If a data action is being used in multiple modules with both server/client "runOn" configurations, the data action will run only on the server side. Therefore, it is important to avoid conflicting the **runOn** configurations if the module needs to be rendered explicitly on the client side.

## Register a core data action

The Dynamics 365 Commerce online SDK contains a set of core data actions that can be used to perform typical Commerce data tasks. Interfaces for core data actions can be found under the `\node_modules\@msdyn365-commerce-modules\retail-actions\dist\lib` directory. To register a core data action so that you can use it inside your module, use the following format in the **dataActions** node of your `MODULE_definition.json` file.

```

{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "categories": [
    "storytelling"
  ],
  "tags": [
    ""
  ],
  "dataActions": {
    "products": {
      "path": "@msdyn365-commerce-modules/retail-actions/dist/lib/get-simple-products",
      "runOn": "server"
    }
  },
  "config": {
    "productIds": {
      "friendlyName": "Product ID",
      "description": "Provide a Product Id that the module will display",
      "type": "string",
      "scope": "module",
      "group": "Content Properties"
    }
  },
  "resources": {
    "resourceKey": {
      "comment": "resource description",
      "value": "resource value"
    }
  }
}

```

Notice that this example calls a core data action that is named **get-simple-products**. This data action uses the **productIds** config property to get the list of product IDs and returns an array of **SimpleProduct** results. The following example shows the module `data.ts` file that defines the return value. The interface for the **SimpleProduct** return type is defined in the `\node_modules\@msdyn365-commerce\commerce-entities\dist\types\commerce-entities` directory that is imported at the top of the example.

```

import { AsyncResult, SimpleProduct } from '@msdyn365-commerce/retail-proxy';

export interface IProductFeatureData {
  products: AsyncResult<SimpleProduct>[];
}

```

## Additional resources

[Data actions overview](#)

[Data action cache options](#)

[Test data actions with mocks](#)

[Event-based data actions](#)

[Core data actions](#)

[Call Retail Server APIs](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Use event-based data actions

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to use event-based data actions.

## Overview

In some scenarios, you don't want a data action to run when a page is first loaded. Instead, you want it to run dynamically in response to some event on the client. For example, a product can be added to a customer's cart in response to a button click, search results can be shown in response to a change in the text input, or banner text can be updated in response to a time-based event.

## Example

In the following example, a very basic module loads product information when a user clicks a button.

The following code shows a **react** component that contains a button that users can click.

```
// product-button.tsx
import * as React from 'react';

/**
 * ProductAddToCart component
 */
class ProductButton extends React.Component {
  public render(): JSX.Element {
    return (
      <div>
        <button onClick={_getProductInfo}>Get Product Info!</button>
      </div>
    );
  }

  // On-Click handler function
  private _getProductInfo = () => {
    console.log('We need to get the product!');
  }
}

export default ProductButton;
```

Currently, this component just logs a message to the console when the button is clicked. To replace that behavior with the data action, you must do three things:

1. Import the data action and its input class.
2. Create an input for the data action.
3. Invoke the data action.

The following code shows an updated **react** component that makes a call for a product when the button is clicked.

```
// product-button.tsx
import * as React from 'react';

// Import our data action and input class
// NOTE: Generally the data action is the DEFAULT export of the file it was written in.
import getProductDataAction, { ProductInput } from '../actions/get-product';

/**
 * ProductAddToCart component
 */
class ProductButton extends React.Component {
  public render(): JSX.Element {
    return (
      <div>
        <button onClick={_getProductInfo}>Get Product Info!</button>
      </div>
    );
  }

  // On-click handler
  private _getProductInfo = async() => {
    // Create the input for our data-action
    const actionInput = new ProductInput('12345');

    // Run and await the result of the data action
    const product = await getProductDataAction(actionInput, {callerContext:
this.props.context.actionContext});

    // Log the result to the console
    console.log(product);
  }
}

export default ProductButton;
```

#### NOTE

This example uses a hard-coded **productId** value. However, it can be updated so that the **productId** value is read from text input or the module's configuration properties.

## Additional resources

[Data actions overview](#)

[Data action cache options](#)

[Test data actions with mocks](#)

[Page load data actions](#)

[Core data actions](#)

[Call Retail Server APIs](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Core data actions

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic covers the list of core data actions that are included with the Microsoft Dynamics 365 Commerce e-Commerce software development kit (SDK). You can find all core data actions in the `\node_modules\@msdyn365-commerce-modules\retail-actions\dist\lib` directory.

## NOTE

All core data actions are observable data actions that are wrapped in an `AsyncResult` class.

DATA ACTION	DESCRIPTION
add-address	Add a new address to an existing customer.
checkout	Initiate checkout for an order.
delete-address	Delete an existing customer address.
generic-data-action	A sample data action.
get-active-cart	Get the active cart, or create an active cart if none exists.
get-address	Return a list of customer addresses.
get-card-payment-accept-point	Return card payment information.
get-categories	Return a list of product categories.
get-categories-hierarchy	Return a list of product categories as a hierarchy.
get-checkout-cart	Return information about the current checkout cart.
get-current-category	Return a category hierarchy for a given category ID.
get-customer	Return customer information for a given customer account number.
get-customer-loyalty-cards	Return a list of loyalty cards for a given customer.
get-dimensions-for-selected-variant	Return a list of available product dimensions for a given product variant.
get-full-products	Get detailed product information, including prices and ratings, for a product.
get-full-products-by-category	Get detailed product information for the products in a specific category.

DATA ACTION	DESCRIPTION
get-full-products-by-refine-search-category	Get detailed product information for a category when refiners are applied.
get-full-products-by-refine-search-text	Get detailed product information, based on a given search term, when refiners are applied.
get-full-products-search-by-text	Get detailed product information, based on a given search term.
get-list	Return a list of products.
get-order-history	Return a list of past orders for a customer.
get-org-unit-configuration	Return organizational unit information.
get-products-by-category	Return a list of products for a specific category.
get-recommendations	Return a list of product recommendations.
get-refiners-by-category	Return a list of applicable refiners for a given category.
get-refiners-by-text	Return a list of applicable refiners for a given text search term.
get-related-products	Return a list of related products.
get-selected-variant	Return a specific variant for a given product.
get-simple-products	Return a list of products together with basic information for a list of product IDs.
get-wishlist-by-customer-id	Return a customer's wish list.
refine-search-by-category	Return a list of products, based on the selected category.
search-org-unit-locations	Return a list of store locations.
update-address	Update an existing customer address.
update-line-delivery-specifications	Update delivery information for line items in the cart.
update-primary-address	Update a customer's primary address.

## Additional resources

[Data actions overview](#)

[Data action cache options](#)

[Test data actions with mocks](#)

[Page load data actions](#)

## Event-based data actions

### Call Retail Server APIs

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Call Retail Server APIs

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic explains how to call application programming interfaces (APIs) for Microsoft Dynamics 365 Retail Server from a data action or directly from module code.

## Overview

To call Retail Server APIs, you must use the Retail Server proxy library that Retail Server provides. This proxy library is also known as TypeScriptProxy or TSProxy. It allows for streamlined communication with Retail Server from JavaScript-based or TypeScript-based environments.

## Install the Retail Server proxy

The Retail Server proxy is available for download via the Dynamics 365 npm feed and should be added by default. If it isn't there, you can get it by adding a reference in the packages.json file.

To install the proxy in your software development kit (SDK) development environment, follow these steps.

1. Determine your current active version of Retail Server. This version will be the version of the Retail Server NuGet package that you use for back-end extensibility.
2. In the **package.json** file, add the following entry in the **dependencies** section. (This entry might already be present and have up-to-date version information.)

```
"@msdyn365-commerce/retail-proxy": "{RETAIL_SERVER_VERSION}"
```

3. Run **yarn install**.

You should now have access to the correct Retail Server proxy for your project. You might see that a reference is already included as part of the module library.

## Retail Server proxy data action managers

The Retail Server proxy contains a set of APIs that communicate internally with Retail Server via HTTP. These APIs are all available through a set of data action managers. To import the code for these data action managers, you can use the following import paths.

```
// Generic example
import {} from '@msdyn365-commerce/retail-proxy/dist/DataActions/{DATA_ACTION_MANAGER_NAME}.g';

// Specific example
import { getByIdsAsync } from '@msdyn365-commerce/retail-proxy/dist/DataActions/ProductsDataActions.g';
```

The following data action managers are available:

- CartsDataActions
- CatalogsDataActions
- CategoriesDataActions
- CommerceListsDataActions
- CustomersDataActions

- EmployeesDataActions
- OrgUnitsDataActions
- PickingListsDataActions
- ProductsDataActions
- PurchaseOrdersDataActions
- RecommendationsDataActions
- SalesOrdersDataActions
- ScanResultsDataActions
- ShiftsDataActions
- StockCountJournalsDataActions
- StoreOperationsDataActions
- SuspendedCartsDataActions
- TransferOrdersDataActions
- WarehousesDataActions

For a list of all the available Retail Server APIs in each data action manager, see [Retail Server Customer and Consumer APIs](#).

## Retail Server proxy data methods

The Retail Server proxy is closely linked to the [Data Action Framework](#). Therefore, for every Retail Server API, there are two exposed Retail Server proxy methods:

- **The createInput method** – This method creates an `IActionInput` class that can be used either to run a [page load data action](#), or to do a direct state update or fetch via the `actionContext.update()` or `actionContext.get()` methods. This method is always named `create{RETAIL_SERVER_API_NAME}Input`.
- **The action method** – This method can be invoked on its own as an [event-based data action](#), or it can be added inside another action method to create a [data action chain](#). This method is always named `{RETAIL_SERVER_API_NAME}Async`.

## Create a page load Retail Server proxy data action

When you want to attach a Retail Server proxy API call to a module so that it will run when a page is loaded, you must create a new data action. The process resembles the process for creating a standard page load data action.

In this example, you will create a module that uses the Retail Server proxy to get all the categories that are available for the configured channel when a page is loaded. To start, you must identify the correct Retail Server proxy API to use. For this example, this API is the `GetCategories` API that is provided by the `CategoriesDataActions` data action manager. You can then construct a data action that can be used in a module definition. In general, to complete this task, you must do two things:

- Provide a **createInput** method that calls the Retail Server proxy **createInput** method for the desired API and passes it any contextual data that you want (for example, the channel ID).
- Have the action method of the new data action be the **retailAction** method that is provided by the Retail Server proxy. The **retailAction** method is designed to parse the input that is passed to it and call the corresponding Retail Server proxy API.

Under the `src\actions` directory, create a file for a new data action. For this example, the file is named `get-category-list.ts`, and it contains the following code.

```

import { createObservableDataAction, IAction, ICreateActionContext } from '@msdyn365-commerce/core';
import { Category, retailAction } from '@msdyn365-commerce/retail-proxy';
import { createGetCategoriesInput } from '@msdyn365-commerce/retail-proxy/dist/DataActions/CategoriesDataActions.g';

/**
 * Get Org Unit Configuration Data Action
 */
export default createObservableDataAction({
  action: <IAction<Category[]>>retailAction,
  input: (context: ICreateActionContext) => {
    return createGetCategoriesInput({ Paging: { Top: 0 } },
    context.requestContext.apiSettings.channelId);
  }
});

```

The `get-category-list.ts` file exports a data action that can be registered on a module to get the list of all categories from whatever channel has been configured for the project. Because this approach requires much less custom code to make the HTTP call than manual communication with Retail Server, we recommend that you always call Retail Server by using the Retail Server proxy.

The following example shows how the `get-category-list` data action can be registered in the `"module" > "dataActions"` node in the module definition file.

```

{
  "$type": "contentModule",
  "friendlyName": "Product Feature",
  "name": "product-feature",
  "description": "Feature module used to highlight a product.",
  "categories": [
    "storytelling"
  ],
  "tags": [
    ""
  ],
  "dataActions": {
    "products": {
      "path": "@msdyn365-commerce-modules/retail-actions/dist/lib/get-simple-products",
      "runOn": "server"
    },
    "productReviews": {
      "path": "../../actions/getproductreviews",
      "runOn": "server"
    },
    "categories": {
      "path": "../../actions/get-category-list",
      "runOn": "server"
    }
  },
  "config": {
    "imageAlignment": {
      "friendlyName": "Image Alignment",
      "description": "Sets the desired alignment of the image, either left or right on the text.",
      "type": "string",
      "enum": {
        "left": "Left",
        "right": "Right"
      },
      "default": "left",
      "scope": "module",
      "group": "Layout Properties"
    },
    "productTitle": {
      "type": "string",
      "friendlyName": "Product Title".
    }
  }
}

```



```

    "friendlyName": "Product Placement",
    "description": "Product placement title"
  },
  "productDetails": {
    "type": "richText",
    "friendlyName": "Product Details",
    "description": "Rich text representing the featured product details"
  },
  "productImage": {
    "type": "image",
    "friendlyName": "Product Image",
    "description": "Image representing the featured product"
  },
  "buttonText": {
    "type": "string",
    "friendlyName": "Button Text",
    "description": "Text to show on the call to action button"
  },
  "productIds": {
    "friendlyName": "Product ID",
    "description": "Provide a Product Id that the module will display",
    "type": "string",
    "scope": "module",
    "group": "Content Properties"
  }
},
"resources": {
  "resourceKey": {
    "comment": "resource description",
    "value": "resource value"
  }
}
}

```

The module `data.ts` file also requires an entry for the return type of the data action. The following example shows a sample module `data.ts` file. After it's implemented, the property can be accessed from the module's view file by using the `this.props.data` object.

```

import { AsyncResult, Category, SimpleProduct } from '@msdyn365-commerce/retail-proxy';
export interface IProductFeatureData {
  products: AsyncResult<SimpleProduct>[];
  categories: AsyncResult<Category>[];
}

```

## Call a Retail Server proxy API directly in module code

The following example shows how to call the `getCategories` Commerce API by using the Commerce proxy `getCategoriesAsync` wrapper API. This API call returns a list of all categories, and the sample code logs them to the console.

```

import * as React from 'react';

import { getCategoriesAsync } from '@msdyn365-commerce/retail-
proxy/dist/DataActions/CategoriesDataActions.g';
import { IProductFeatureData } from './product-feature.data';
import { imageAlignment, IProductFeatureProps } from './product-feature.props.autogenerated';

export interface IProductFeatureViewProps extends IProductFeatureProps<{}> {
  productName: string;
}

/**
 *
 * ProductFeature component
 * @extends {React.PureComponent<IProductFeatureProps<IProductFeatureData>>}
 */
class ProductFeature extends React.PureComponent<IProductFeatureProps<IProductFeatureData>> {
  public render(): JSX.Element | null {
    const {
      config,
    } = this.props;

    // set default product info values from config values if available
    let ProductName = config.productTitle ? config.productTitle : 'No product name defined';

    getCategoriesAsync({ callerContext: this.props.context.actionContext },
this.props.context.request.apiSettings.channelId)
      .then(categoryList2 => {
        let categories2 = '';
        for (let i = 0; i < categoryList2.length; i++) {
          categories2 += `${categoryList2[i].Name}, `;
        }
        console.log(categories2);
      });

    const ProductFeatureViewProps = {
      ...this.props,
      productName: ProductName,
    };

    return this.props.renderView(ProductFeatureViewProps) as React.ReactElement;
  }
}

export default ProductFeature;

```

## Additional resources

[Data actions overview](#)

[Data action cache options](#)

[Test data actions with mocks](#)

[Page load data actions](#)

[Event-based data actions](#)

[Core data actions](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Call Retail Server extension APIs

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to call Microsoft Dynamics 365 Retail Server extension application programming interfaces (APIs) from data actions or directly from module code.

## Overview

Dynamics 365 Retail Server extension APIs can be called from either the point of sale (POS) system or from e-Commerce modules and data actions. To call the APIs from e-Commerce modules and data actions, you must create proxy module TypeScript (.ts) files by using a tool that is provided as part of the Dynamics 365 Retail software development kit (SDK). You can then include these files in your e-Commerce configurations and use them to call the Retail Server extension APIs from e-Commerce modules and data actions.

### NOTE

This topic doesn't explain how to create Retail Server extensions. For more information, see [Create a new Retail Server extension](#).

It's assumed that the following prerequisites are in place:

- A Retail Server extension has already been deployed.
- You have access to the Retail Server extension dynamic-link libraries (DLLs) that are available.

In addition, make sure that the .env file **MSDyn365Commerce\_BASEURL** has a value that points to the environment that has the deployed Retail Server extension, so that you can test against it. If this option isn't available, you can use a mock-up instead.

If you're developing on the local Tier 1 Retail Server virtual machine (VM), you must point **MSDyn365Commerce\_BASEURL** to the local URL (`https://usnconeboxax1ret.cloud.onebox.dynamics.com/`), and make sure that **MSDyn365Commerce\_CHANNELID** and **MSDyn365Commerce\_OUN** are set to the appropriate online channel that you're using. Here is an example of an .env file.

```
...
MSDyn365Commerce_BASEURL=https://usnconeboxax1ret.cloud.onebox.dynamics.com/
MSDyn365Commerce_CHANNELID=68719478279
MSDyn365Commerce_CATALOGID=0
MSDyn365Commerce_OUN=128
...
```

For more information, see [Configure a development environment \(.env\) file](#).

You might also have to change the following setting in the Retail Server web.config file, so that the call can go through in the local development environment.

```
<add key="AllowedOrigins" value="*" />
<!-- <add key="AllowedOrigins"
value="https://usnconeboxax1pos.cloud.onebox.dynamics.com;https://usnconeboxax1ecom.cloud.onebox.dynamics.com" /> -->
```

# Create proxy files

You will need the Retail SDK to generate proxy files. If you are using a Tier 1 development VM environment, make sure that you have the latest Retail SDK installed. For more information, see [Migrate the Retail SDK from Visual Studio 2015 to Visual Studio 2017](#).

For information about how Retail extensions can be called from Retail POS, see [Typescript and C# proxies for Retail point of sale \(POS\)](#). That topic explains how to create a proxy file by using a command that resembles the following command.

```
K:\RetailSDK\References\microsoft.dynamics.commerce.tools.coreproxygenerator\10.14.20128.1\tools>CommerceProxyGenerator.exe
..\..\..\microsoft.dynamics.commerce.tools.extensionsproxygenerator\9.22.20167.4\tools\Microsoft.Dynamics.Retail.RetailServerLibrary.dll k:\WarrantySample\Contoso.RetailServer.WarrantySample.dll
/application:typescriptextensions
```

## NOTE

The SDK version that is referenced in the command might differ, depending on the version of Retail Server that you're running.

The process for creating proxy files for e-Commerce is similar, but the final `/application:typescriptextensions` option is replaced by `/application:typescriptmoduleextensions`, as shown in the following example.

```
K:\RetailSDK\References\microsoft.dynamics.commerce.tools.coreproxygenerator\10.14.20128.1\tools>CommerceProxyGenerator.exe
..\..\..\microsoft.dynamics.commerce.tools.extensionsproxygenerator\9.22.20167.4\tools\Microsoft.Dynamics.Retail.RetailServerLibrary.dll k:\WarrantySample\Contoso.RetailServer.WarrantySample.dll
/application:typescriptmoduleextensions
```

After you run the preceding command, two new files are generated: `DataActionExtension.g.ts` and `DataServiceEntities.g.ts`.

## Proxy data methods

The proxy is closely linked to the data action framework. For every Retail Server extension API, there are two exposed proxy methods:

- **createInput** – This method creates an `IActionInput` class that can be used either to run a page load data action, or to do a direct state update or fetch via the `actionContext.update()` or `actionContext.get()` methods. This method is always named `create{COMMERCE_SCALE_UNIT_EXTENSION_API_NAME}Input`.
- **action** – This method can be invoked on its own as an event-based data action, or it can be added inside another action method to create a data action chain. This method is always named `{RETAIL_SERVER_API_NAME}Async`.

## Call a proxy API with a data action

From the Dynamics 365 Commerce online SDK, include the two new files from the previous section in the `/src/actions/` directory. Then create a new data action .ts file, and paste in code that resembles the following example to call the Retail Server extension APIs from the data action.

The following example uses a file that is named `get-warranty-info.ts`. Notice that it imports the `DataActionExtension.g` and `DataServiceEntities.g` files that were generated earlier, and it calls the

createGetWarrantyByProductIdInput Retail Server extension API as defined in the proxy file.

```
import { createObservableDataAction, IAction, ICreateActionContext } from '@msdyn365-commerce/core';
import { retailAction } from '@msdyn365-commerce/retail-proxy';
import { createGetWarrantyByProductIdInput } from './DataActionExtension.g';
import { IProductWarranty } from './DataServiceEntities.g';
/**
 * Get Warranty Info Data Action
 */
export default createObservableDataAction({
  action: <IAction<IProductWarranty[]>>retailAction,
  input: (context: ICreateActionContext) => {
    return createGetWarrantyByProductIdInput({Paging:{Top:250}}, '12345');
  }
});
```

You can now call the data action just as you might call any other data action, by declaring it in the **dataActions** node of your `module.definition.json` file, as shown in the following example. The data action will be called at page load time.

```
{
  "$type": "contentModule",
  "friendlyName": "Warranty List",
  "name": "warrantylist",
  "description": "Provides a list of available warranty services available for purchase.",
  "categories": [
    "warranty"
  ],
  "tags": [
    ""
  ],
  "dataActions": {
    "productWarranties": {
      "path": "../../actions/get-warranty-info",
      "runOn": "server"
    }
  },
  "resources": {
    "warrantyName": {
      "value": "Warranty Name",
      "comment": "Text for the warranty name header"
    },
    "warrantyDescription": {
      "value": "Description",
      "comment": "Text for the warranty description header"
    },
    "warrantyPrice": {
      "value": "Price",
      "comment": "Text for the warranty price header"
    },
    "warrantyBuyButton": {
      "value": "Buy now",
      "comment": "Text for the warranty action button"
    }
  }
}
```

You must also add the return data type to the module's `data.ts` file, as shown in the following example.

```
import { AsyncResult } from '@msdyn365-commerce/retail-proxy';
import { IProductWarranty } from '../actions/DataServiceEntities.g';

export interface IWarrantylistData {
  productWarranties: AsyncResult<IProductWarranty[]>;
}

```

Finally, you can access the data from your module React component by using the **this.props.data.productWarranties** property. The following example shows a sample module view.tsx file that renders data coming back from the Retail Server extension API.

```
import * as React from 'react';
import { IWarrantylistsampleViewProps } from './warrantylistsample';
export default (props: IWarrantylistsampleViewProps) => {
  const {
    data: { productWarranties }
  } = props;
  if (productWarranties && productWarranties.result) {
    return (
      <table className='table'>
        <thead>
          <tr>
            <th>Warranty</th>
            <th>Price</th>
          </tr>
        </thead>
        <tbody>
          {productWarranties.result.map(warranty => (
            <tr>
              <td>{warranty.Description}</td>
              <td>${warranty.Price}</td>
              <button type='button' className='btn btn-primary'>Buy Now</button>
            </tr>
          ))}
        </tbody>
      </table>
    );
  } else {
    return <div>No warranty info returned</div>;
  }
};

```

## Call proxy APIs directly

The following example shows a Retail Server API being called directly inside of a module's typescript React view code without using a data action.

```
import * as React from 'react';
import { getWarrantyByProductIdAsync } from '../actions/DataActionExtension.g';
import { IWarrantyListViewProps } from './warrantylist';
export default (props: IWarrantyListViewProps) => {
  const _getWarrantyOnClick = async() => {
    getWarrantyByProductIdAsync({ callerContext: props.context.actionContext }, '12345').then( result =>
    {
      ...
    });
  }

  return (
    <div>
      <button onClick={_getWarrantyOnClick}>Get Warranty Info</button>
    </div>
  );
};
```

## Additional resources

[Data actions overview](#)

[Data action cache options](#)

[Test data actions with mocks](#)

[Page load data actions](#)

[Event-based data actions](#)

[Core data actions](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Chain data actions

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to chain data actions.

## Overview

To create a maintainable and compact codebase, you will often need a suite of composable data actions that can easily use each another to create more complex code flows. The Microsoft Dynamics 365 Commerce Online Software Development Kit (SDK) lets you seamlessly chain data actions but still provide all the out-of-box benefits of the data action architecture (caching, batching, and deduplicating).

## Examples

The following examples show data action chaining. First, a product information call is made. This call is followed by an inventory call.

The first example shows a data action that gets product information.

```

// get-product.ts
import { IAction, IActionInput } from '@msdyn365-commerce/action';
import { SimpleProduct } from '@msdyn365-commerce/commerce-entities';
import { CommerceEntityInput, createObservableDataAction, IAny, ICreateActionContext, IActionContext,
IGeneric, IHTTPError, IHTTPResponse, sendCommerceRequest } from '@msdyn365-commerce/core';
import { ProductInput } from './inputs/product-input';

/**
 * Product Action Input Class
 * This class is used by our Action Function, so that it has access to a productId
 */
export class ProductInput implements IActionInput {
  public channelId: number;
  public productId: number;
  constructor(productId: number, channelId: number) {
    this.channelId = channelId;
    this.productId = productId;
  }
  public getCacheKey = () => `${this.channelId}-${this.productId}`;
  public getCacheObjectType = () => 'Product';
  public dataCacheType = (): Msdyn365.CacheType => 'application';
}

/**
 * Action Function which calls the Retail API and returns the product based on the passed ProductInputs
productId
 */
async function getSimpleProductAction(input: ProductInput, ctx: IActionContext): Promise<SimpleProduct> {
  const { apiSettings } = ctx.requestContext;

  // Construct our HTTP request using information available in actionContext (ctx), and our Action Input
(input)
  const requestUrl = `${apiSettings.baseUrl}/Products/${input.productId}`;

  // Make the HTTP Request
  return sendCommerceRequest<SimpleProduct>(requestUrl, 'get')
    .then((response: IHTTPResponse) => {
      if(response.data) {
        return response.data;
      }
      ctx.trace('[getSimpleProductAction] Invalid response from server');
      return <SimpleProduct>[];
    })
    .catch((error: IHTTPError) => {
      ctx.trace(error.message);
      ctx.trace(error.stack || '');
      ctx.trace(`Unable to Fetch Product.`);
      return <SimpleProduct>{};
    });
}

/**
 * This exports the action Data Action, which the result of passing your action method and createInput
method (if used)
 * to the createDataAction method.
 */
export default createObservableDataAction({
  action: <IAction<SimpleProduct>>getSimpleProductAction
});

```

The second example shows a data action that gets inventory information.

```

// check-inventory.ts
import { IAction, IActionInput } from '@msdyn365-commerce/action';
import { ProductAvailableQuantity } from '@msdyn365-commerce/commerce-entities';
import { createObservableDataAction, IAny, IActionContext, IGeneric, IHTTPError, IHTTPResponse,
sendCommerceRequest } from '@msdyn365-commerce/core';

// Because this API also only needs a productId, we can re-use the ProductInput we created earlier here.
import { ProductInput } from './get-product.ts';

/**
 * Calls the Retail API and returns the Availability information for the passed Product
 */
async function getProductAvailabilityAction(input: ProductInput, ctx: IActionContext):
Promise<ProductAvailableQuantity> {
    const { apiSettings } = ctx.requestContext;

    // Construct our HTTP request using information available in actionContext (ctx), and our Action Input
    (input)
    const requestUrl = `${apiSettings.baseUrl}/Products/GetProductAvailabilities`;
    const requestBody = {
        productId: input.productId;
    }

    // Make the HTTP Call and handle the response
    return sendCommerceRequest<ProductAvailableQuantity>(requestUrl, 'post', requestBody)
        .then((response: IHTTPResponse) => {
            if(response.data) {
                return response.data;
            }
            ctx.trace('[getProductAvailabilityAction] Invalid response from server');
            return <ProductAvailableQuantity>[];
        })
        .catch((error: IHTTPError) => {
            ctx.trace(error.message);
            ctx.trace(error.stack || '');
            ctx.trace(`Unable to Fetch Product Availability.`);
            return <ProductAvailableQuantity>{};
        });
}

export default createObservableDataAction({
    action: getProductAvailabilityAction
})

```

Now that you have actions for getting the product data and the **ProductAvailableQuantity** value of a product, you can create a new chain data action. Chain data actions are just data actions that invoke other data actions as part of their execution.

The following example shows the **getProductWithAvailability** chain data action, which uses both the preceding actions.

```

import getProduct, { ProductInput } from './get-product';
import getProductAvailability from './check-inventory';

/**
 * This interface will be the return type of our chain data action.
 * This contains both the basic product information in addition to the product's availability information.
 */
export interface SimpleProductWithAvailability extends SimpleProduct {
  availability: ProductAvailableQuantity
}

/**
 * Calls the Retail API and returns the Availability information for the passed Product
 */
async function getProductWithAvailabilityAction(input: ProductInput, ctx: IActionContext):
Promise<SimpleProductWithAvailability> {
  // First we get the product
  const product: SimpleProductWithAvailability = await <SimpleProductWithAvailability>getProduct(input,
ctx);

  // If we successfully get the product, then we try to get its availability information.
  if(product) {
    // Get the availability information
    product.availability = await getProductAvailability(input, ctx)
  } else {
    return <SimpleProductWithAvailability>{};
  }
}

export default createObservableDataAction({
  action: getProductWithAvailabilityAction
})

```

You can now use the new chain data action wherever you need both the basic product information and the product's current inventory status. In addition, the calls that exist in the chain data action are still run through the cache. They are also batched and deduplicated together with other actions that are run on the same page.

## Additional resources

[Batch data actions](#)

[Create an observable data action](#)

[Share state across modules](#)

[Data action cache settings](#)

[Data action overrides](#)

[Data action hooks](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Batch data actions

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to batch data actions.

## Overview

Often you'll have an application that requires many calls to the same application programming interface (API) during the load of a single page. An example is a product feature page that showcases information about many products instead of just one product. In a typical approach, multiple calls are made to the data action to get products. However, because this approach uses many individual HTTP requests to get the product information, it might not be efficient. To solve this issue, the data action architecture supports batchable data actions.

## Examples

The main difference between a batch data action and a standard data action is the batch data action's support for an array of action inputs. In the following example of a standard data action, notice that the `getSimpleProductAction` action method accepts only one `ProductInput` class.

```
async function getSimpleProductAction(input: ProductInput, ctx: IActionContext): Promise<SimpleProduct>
```

To change this data action to a batch data action, modify the method signature so that the method can accept an array of `ProductInput` classes and return an array of `SimpleProduct` objects. The following example shows how the data action method must be updated so that it can process an array of inputs and return an array.

```
async function getSimpleProductsAction(inputs: ProductInput[], ctx: IActionContext):  
Promise<SimpleProduct[]> {  
    const { apiSettings } = ctx.requestContext;  
  
    // Construct our HTTP request using information available in actionContext (ctx), and our Action Input  
    (input)  
    const requestUrl = `${apiSettings.baseUrl}/Products`;  
  
    // Construct our request context from all the passed inputs  
    const requestBody = {  
        productIds: inputs.map(input => input.productId);  
    }  
  
    // Get the SimpleProducts  
    return sendCommerceRequest<SimpleProduct[]>(requestUrl, 'post', requestBody)  
        .then((response: IHTTPResponse) => {  
            if(response.data) {  
                return response.data;  
            }  
            ctx.trace('[getSimpleProductsAction] Invalid response from server');  
            return <SimpleProduct[]>[];  
        })  
        .catch((error: IHTTPError) => {  
            ctx.trace(error.message);  
            ctx.trace(error.stack || '');  
            ctx.trace(`Unable to Fetch Products.`);  
            return <SimpleProduct[]>[];  
        });  
}
```

Now that the data action method has been updated so that it can handle an array of inputs, the `isBatched` property in the action creation call must be set to `true`.

```
export default createObservableDataAction({
  action: <IAction<SimpleProduct[]>>getSimpleProductsAction,
  input: createInput,
  isBatched: true
});
```

Because this action now supports batching, if the action is called in multiple places during a page load, the data action framework automatically groups the requests together. That's why this approach helps minimize the number of HTTP requests that are required and helps maximize performance.

#### NOTE

Some APIs might not support batching on their side. So when you create a batch data action, you should confirm that the service that you're using can support the action.

## Additional resources

[Chain data actions](#)

[Create an observable data action](#)

[Share state across modules](#)

[Data action cache settings](#)

[Data action overrides](#)

[Data action hooks](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create an observable data action

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to create an observable data action in Microsoft Dynamics 365 Commerce.

## Overview

Observable data actions are used to track the status of a data action as it's running. This capability is helpful if you must run logic or render a user interface (UI) in response to the current status of a data action. Observable data actions use a special promise-like class that is named **AsyncResult**. This class adds "observer" functionality to a standard promise.

The following example shows how to take advantage of an **AsyncResult** class. This data action waits three seconds before it returns a string.

```
// test-action.ts
/**
 * Test action method for TestAsync action
 * @param input The action input
 * @param context The action context
 */
const testAction = async (input: TestAsyncActionInput, context: IActionContext): Promise<string> => {
  await new Promise(resolve => {
    setTimeout(
      () => {
        resolve();
      }, 3000);
  });

  // Flag-based error scenario
  if (input.shouldError) {
    throw new Error('Oops');
  }
  return 'foo';
};

/**
 * Test Action Input
 */
export class TestAsyncActionInput implements IActionInput {
  public shouldError: Boolean = false;
  constructor(shouldError?: Boolean) {
    this.shouldError = shouldError || this.shouldError;
  }
  public getCacheKey = () => `test`;
  public getCacheObjectType = () => 'test';
  public dataCacheType = (): Msdyn365.CacheType => 'none';
}

/**
 * Test createInput method for TestAsync Action
 */
const createInput = () => {
  return new TestAsyncActionInput();
};
```

In this example, the data action simulates an outgoing application programming interface (API) call. It also includes a flag that can be set to allow a failure simulation. In some scenarios, you might want to create a

module that can send updates about the status of a data action. You can't use a regular data action for that purpose.

## Create the observable data action

To create the observable data action, use the new `createObservableDataAction` utility method for data action creation.

```
export default createObservableDataAction({
  input: createInput,
  action: <IAction<string>>testAction
});
```

The `createObservableDataAction` method is equivalent to the `createDataAction` method, but returns an `IObservableAction` interface instead of an `IAction` interface. The `IObservableAction` interface returns an `AsyncResult` class instead of the `Promise` class that a standard `IAction` interface returns. The `AsyncResult` class provides additional data including the `status` and `error` properties of the data action.

A mock can be created to test the data action, as shown in the following example.



```

// test-module.tsx
import { observer } from 'mobx-react';
import * as React from 'react';
import { IAsyncTestModuleData } from './async-test-module.data';
import { IAsyncTestModuleProps } from './async-test-module.props.autogenerated';
import testAction, { TestAsyncActionInput } from './actions/async-test-action';

/**
 * TestModule Component
 * @extends {React.PureComponent<IAsyncTestModuleProps<IAsyncTestModuleData>>}
 */
@observer
class AsyncTestModule extends React.PureComponent<IAsyncTestModuleProps<IAsyncTestModuleData>> {
  constructor(props: IAsyncTestModuleProps<IAsyncTestModuleData>) {
    super(props);
  }
  public render(): JSX.Element {
    return (
      <div className='row'>
        <div className='col'>
          <h1>Status: { this.props.data.testResult.status }</h1>
          <h1>Result: { this.props.data.testResult.result }</h1>
          { this.props.data.testResult.error &&
            <h1>Error: { this.props.data.testResult.error.message }</h1>
          }
          <button
            // tslint:disable:jsx-no-lambda
            // tslint:disable-next-line:react-this-binding-issue
            onClick={(e) => this._clientCall(e)}
          > Run on client
          </button>
          <button
            // tslint:disable:jsx-no-lambda
            // tslint:disable-next-line:react-this-binding-issue
            onClick={(e) => this._clientCall(e, true)}
          > Error on client
          </button>
        </div>
      </div>
    );
  }
}
export default AsyncTestModule;

```

The following example shows a sample module definition that registers the sample observable data action that was created earlier.

```

// test-module.definition.json
{
  "$type": "contentModule",
  "friendlyName": "test-module",
  "name": "test-module",
  "description": "Module for testing observable data actions",
  "categories": ["test-module"],
  "tags": ["samples"],
  "dataActions": {
    "testResult": {
      "path": "./actions/test-action",
      "runOn": "client"
    }
  }
}

```

When adding a data action inside the module data.ts file, ensure that every data action that returns an

**observableDataAction** object is wrapped by an **AsyncResult** class. This will guarantee the correct typings when a module is written.

```
// test-module.data.ts
import { AsyncResult } from '@msdyn365-commerce/retail-proxy';
export interface IAsyncTestModuleData {
  testResult: AsyncResult<string>;
}
```

When a data action is wrapped in an **AsyncResult** class (as in the above example), the module will now have access to the **status**, **result**, and **error** properties. The **status** property contains the current state of the data action, which can be one of: **'Success'**, **'Loading'** or **'Failed'**. The **result** property contains the data that is returned by the action if it succeeds. If the data action throws an error, the **result** property won't be filled in. Instead, the **error** property can be used to see the error details.

By taking advantage of the **status**, **result**, and **error** properties that are provided by observable data actions, complicated scenarios can be better handled in a module. Examples include showing a loading screen while a data action call runs and providing contextual error messages in response to a failed data action.

## Additional resources

[Chain data actions](#)

[Batch data actions](#)

[Share state across modules](#)

[Data action cache settings](#)

[Data action overrides](#)

[Data action hooks](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Share state across modules

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic describes how to share state across multiple modules by using data actions in Dynamics 365 Commerce.

## Overview

Data actions perform the important role of state management in situations where you must share state across multiple modules on the same page. In general, state is shared within the application state of the running Node application.

## Examples

In this example, two modules share basic interaction. One module (**sample-button**) has a button, and the other module (**sample-message**) shows a message when that button is selected.

First, you must have a data action that returns an object that contains the number of times that the button has been selected. Here is what the code looks like.

```

// sample-state.ts
import { CacheType, createObservableDataAction, IAction, IActionContext, IActionInput, IAny,
ICreateActionContext, IGeneric } from '@msdyn365-commerce/core';
export interface ISampleState {
  clickCount: number;
}

/**
 * SampleState - action input
 */
export class SampleStateInput implements IActionInput {
  public getCacheKey = () => `SampleState`;
  public getCacheObjectType = () => 'SampleState';
  public dataCacheType = (): CacheType => 'request';
}

/**
 * SampleState - action
 */
export async function sampleStateAction(input: SampleStateInput, ctx: IActionContext): Promise<ISampleState>
{
  return { clickCount: 0 };
}

/**
 * SampleState - create new input for create action
 */
const createInput = (inputData: ICreateActionContext<IGeneric<IAny>>): IActionInput => {
  return new SampleStateInput();
};

/**
 * SampleState - create action
 */
export default createObservableDataAction<ISampleState>({
  action: <IAction<ISampleState>>sampleStateAction,
  input: createInput
});

```

In its current state, this data action has no implementation. It just creates a place to store an object in the cache. Because the two modules in this example must communicate with each other, it's helpful if they both observe this object. To give modules access to this object, you must make sure that the modules register the data action that you created earlier as a page load data action.

Here is the code for the **sample-button** module.

```

// sample-button.definition.json
{
  "$type": "contentModule",
  "friendlyName": "Sample Button",
  "name": "sample-button",
  "description": "Sample Button",
  "categories": ["sample-button"],
  "tags": ["samples"],
  "module": {
    "view": "./sample-button",
    "dataActions": {
      "sampleState": {
        "path": "../..actions/sample-state/sample-state"
      }
    }
  }
}

```

Here is the code for the **sample-message** module.

```
// sample-message.definition.json
{
  "$type": "contentModule",
  "friendlyName": "Sample Message",
  "name": "sample-message",
  "description": "Sample Message",
  "categories": ["sample-message"],
  "tags": ["samples"],
  "dataActions": {
    "sampleState": {
      "path": "../actions/sample-state/sample-state"
    }
  }
}
```

Both modules are now registered to the data action. Therefore, they both observe the same object in the application state. The next step is to update the application state when the **sample-button** module has a user click event. All modules that observe the application state should then be automatically updated. Here is the code for the **sample-message** module.

```
// sample-message.data.ts
import { AsyncResult } from '@msdyn365-commerce/retail-proxy';
import { ISampleState } from '../actions/sample-state/sample-state';
export interface ISampleMessageData {
  sampleState: AsyncResult<ISampleState>;
}
```

```
// sample-message.tsx
import * as React from 'react';
import { ISampleMessageData } from './sample-message.data';
import { ISampleMessageProps } from './sample-message.props.autogenerated';

/**
 * SampleMessage Module used for showcasing cross-module communication
 * @extends {React.Component<ISampleMessageProps<ISampleMessageData>>}
 */
export default class SampleMessage extends React.Component<ISampleMessageProps<ISampleMessageData>> {
  constructor(props: ISampleMessageProps<ISampleMessageData>) {
    super(props);
  }
  public render(): JSX.Element {
    if(this.props.data.sampleState.result) {
      return (<h3>The Button has been clicked {this.props.data.sampleState.result.clickCount} times.
</h3>>);
    }
    return (<h3>Error: No Sample State Detected</h3>>);
  }
}
```

The **sample-message** module is very straightforward. It asks for the **ISampleState** value by using a page load data action. Then, based on the data that is returned, it renders a simple message. Because the application state is internally powered by **MobX**, this module can automatically react when the data that it's observing changes.

Finally, here is the code for the **sample-button** module that updates the application state in response to on a user click event.

```

// sample-button.data.ts
import { AsyncResult } from '@msdyn365-commerce/retail-proxy';
import { ISampleState } from '../actions/sample-state/sample-state';
export interface ISampleButtonData {
  sampleState: AsyncResult<ISampleState>;
}

```

```

// sample-button.tsx
import * as React from 'react';
import { ISampleButtonData } from './sample-button.data';
import { ISampleButtonProps } from './sample-button.props.autogenerated';
import { SampleStateInput } from '../actions/sample-state/sample-state';

/**
 * SampleButton component used for showcasing cross-module communication
 * @extends {React.Component<ISampleButtonProps<ISampleButtonData>>}
 */
export default class SampleButton extends React.Component<ISampleButtonProps<ISampleButtonData>> {
  constructor(props: ISampleButtonProps<ISampleButtonData>) {
    super(props);
    this._onClick.bind(this);
  }
  public render(): JSX.Element {
    return (
      <button onClick={this._onClick}>
        Click Me!
      </button>
    );
  }

  // onClick Handler should update application state
  private _onClick = (e: React.MouseEvent): void => {
    if (this.props.data.sampleState.result) {
      // This will directly update our application state, which should trigger all modules observing
      the state to update
      this.props.context.actionContext.update(new SampleStateInput(), { clickCount:
this.props.data.sampleState.result.clickCount + 1 });
    }
  }
}

```

As you can see, the `onClick` handler makes a call to the `actionContext.update()`. This method lets you directly change the application state. When the state is changed, MobX takes over and re-renders all the modules that are observing the state that includes the `sample-message` module.

## Additional resources

[Chain data actions](#)

[Batch data actions](#)

[Create an observable data action](#)

[Data action cache settings](#)

[Data action overrides](#)

[Data action hooks](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Data action cache settings

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic covers cache settings for data actions in Microsoft Dynamics 365 Commerce.

The software development kit (SDK) \src\settings\cache.settings.json file holds cache settings for entities that are returned from data actions calls. After these entities are cached, they are returned from the cache for all data action requests that have the same cache key, provided that their time to refresh (TTR) and time to live (TTL) values are valid.

For core data actions that are provided in the SDK, the cache key is implicitly set within the data action itself. (For information about how to determine the cache key from the SDK, see the [Determine cache keys for data actions](#) section later in this topic.) For custom data actions, the cache key can be defined in the action input by overriding the `getCacheKey()` method.

For all entities, the default TTR is 60 seconds, and the default TTL is 600 seconds.

The following example shows a JavaScript Object Notation (JSON) file that includes a section for TTR and TTL settings. Each section contains name of the entity for the data action's cache key followed by the time in seconds.

```
{
  "ttlInSeconds": {
    "AttributeValue": 1800,
    "Category": 1800,
    "CategoryHierarchy": 1800,
    "ChannelConfiguration": 1800,
    "FullProduct": 1800,
    "ProductRating": 1800,
    "OrgUnit": 1800,
    "ProductCatalog": 1800,
    "SimpleProduct": 1800
  },
  "ttrInSeconds": {
    "AttributeValue": 900,
    "Category": 900,
    "CategoryHierarchy": 1800,
    "ChannelConfiguration": 1800,
    "FullProduct": 900,
    "OrgUnit": 1800,
    "Product": 900,
    "ProductCatalog": 300,
    "ProductDimensionValue": 900,
    "ProductRating": 900,
    "SimpleProduct": 900
  }
}
```

## Determine cache keys for data actions

You can find the data cache key for a specific core data action by looking at the source code under the SDK \node\_modules\@msdyn365-commerce-modules\retail-actions\dist\lib directory.

For example, to find the code for the "get-simple-products" data action, look in the get-simple-products.js file. By examining the source code of this file in the following example, you can find the entity name **SimpleProduct**. You can then add this entity to the JSON file for cache settings in the previous example.



```

...
export class ProductInput {
  constructor(productId, apiSettings, channelId) {
    this.getCacheKey = () => buildCacheKey(`RecordId-${this.productId.toString()}`, this.apiSettings);
    this.getCacheObjectType = () => 'SimpleProduct';
    this.dataCacheType = () => 'application';
    this.apiSettings = apiSettings;
    this.productId = +productId;
    this.channelId = channelId || apiSettings.channelId;
  }
}
...

```

## Cache settings for specific entities

The following table provides cache setting descriptions for specific data action cache key entities.

ENTITY	DESCRIPTION
AttributeValue	Contains the metadata of the product attributes information shown on a product details page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the product attributes shown. For information about the APIs returning this entity, see <a href="#">Products controller</a> .
Category	Contains the metadata of the category information shown in a navigation module. For better performance, and to reduce round trips to Retail Server, it is recommended to cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the category information shown. For information about the APIs returning this entity, see <a href="#">Categories controller</a> .
CategoryHierarchy	Contains the metadata of the hierarchy of available categories shown in a navigation module. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the categories hierarchy information shown. For information about the APIs returning this entity, see <a href="#">Categories controller</a> .
OrgUnit	Contains metadata of the organization unit used in "buy online, pick up in store" (BOPIS) scenarios and the store locator module. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. For information about the APIs returning this entity, see <a href="#">Org units controller</a> .
ProductDimensionValue	Contains the metadata of the product dimensions information shown on a product details page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the product dimensions shown. For information about the APIs returning this entity, see <a href="#">Products controller</a> .

ENTITY	DESCRIPTION
ProductList-Recommendations	Contains the list of recommended products on a homepage or product details page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the product list recommendations.
ProductList-RelatedProducts	Contains the list of related products, shown on product details page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the related products.
ProductRating	Contains the metadata of the product rating information shown on a product details page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the product rating information shown. For information about the APIs returning this entity, see <a href="#">Products controller</a> .
ProductRefiner	Contains the metadata of the product refiners shown on a product list page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the product refiners shown. For information about the APIs returning this entity, see <a href="#">Products controller</a> .
ProductSearchResult	Contains the metadata of the product search results shown on a product list page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the product list in search results. For information about the APIs returning this entity, see <a href="#">Products controller</a> .
SimpleProduct	Contains the metadata of the product shown on a product details page. For better performance, and to reduce round trips to Retail Server, we recommended that you cache the entity with optimum TTR and TTL values. Longer TTR affects the freshness of the product information shown. For information about the APIs returning this entity, see <a href="#">Products controller</a> .

## Inline cache options

The Dynamics 365 Commerce online SDK's data action layer provides flexibility for controlling how a data action response should be cached and what its scope should be (application or request). For all custom data actions, the cache type can be defined as part of the [action input](#). A common scenario is for data actions to invoke a Retail Server proxy data action, or module business logic that will invoke a Retail Server proxy data action. An example of such a scenario would be when a user selects a "find store" button. For these scenarios, a module can set the "bypassCache" option on the action context when invoking the Retail Server proxy data action. This setting tells the SDK to honor the module's given cache preferences. For more information about Retail Server proxy data actions, see [Call Retail Server APIs](#).

Supported values for bypassCache are:

- **get** - Ignores cache while performing read and fetches the latest information from Retail Server.
- **none** - Action response is cached per request. If `byPassCache` is not specified runtime defaults to this value.

In general, for custom data actions and Retail Server proxy actions that read data from Retail Server (calling APIs with names that start with "get," "search," or "read"), the value of "bypassCache" is set to "none" by default. For all other Retail Server proxy actions configured to fetch the latest data from Retail Server, the value of "bypassCache" is set to "get" by default. These values can be overridden if needed.

The following is an example of usage.

```
if (checkoutCartId) {
  try {
    checkoutCart = await readAsync({ callerContext: ctx, bypassCache: 'get' }, checkoutCartId);
  } catch {
    ctx.telemetry.error('Error getting checkout cart based on saved checkout cart ID');
    ctx.telemetry.exception(error);
  }
}
```

## Additional resources

[Data actions overview](#)

[Core data actions](#)

[Data action overrides](#)

[Commerce Scale Unit customer and consumer APIs](#)

[Call Retail Server APIs](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Data action overrides

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to override default data actions with custom data actions in Dynamics 365 Commerce.

## Overview

The modules included with the Dynamics 365 Commerce software development kit (SDK) module library use pre-existing data actions to fetch data for the modules to use. There may be times when you want to use a custom data action to alter the business logic, but do not want to clone the module because you won't be changing the UX rendered by the module. With data action overrides, you can override the registered action by creating a new action with the same action ID. Overriding a data action will cause all previous uses of the data action, either through import or inclusion in the module definition.json file, to use your new data action.

## Data action ID naming convention

Each data action has a data action ID that's used when the `createObservableDataAction` method is called through the "id" property. The data action ID should follow the naming convention of `[package name]/[module name]/[action name]`, as in the following example.

```
export default createObservableDataAction({
  id: '@msdyn365-commerce-modules/retail-actions/get-address',
  action: <IAction<Address[]>>getAddressAction,
  input: <(args: ICreateActionContext) => IActionInput>createGetAddressInput
});
```

## Data action override example

To override the get-address data action, you need to create a new data action that uses the get-address ID in the `/src/actions` directory.

The following command creates a new data action in the `\src\actions` directory.

```
yarn msdyn365 add-data-action custom-get-address
```

Next, you replace the default template code with your code. In the example below, the data action ID is the same as used in the `...\Msdyn365.Commerce.Online\node_modules@msdyn365-commerce-modules\retail-actions\dist\lib\get-address.js` data action `@msdyn365-commerce-modules/retail-actions/get-address`.

```

import { CacheType, IAction, IActionContext, IActionInput, ICommerceApiSettings } from '@msdyn365-commerce/core';
import { createObservableDataAction, ICreateActionContext } from '@msdyn365-commerce/core';
import { Address } from '@msdyn365-commerce/retail-proxy';

import { readAsync } from '@msdyn365-commerce/retail-proxy/dist/DataActions/CustomersDataActions.g';

/**
 * Input class for the getAddress data action
 */
export class AddressInput implements IActionInput {
    public userAccountNumber: string;
    private apiSettings: ICommerceApiSettings;

    constructor(userAccountNumber: string, apiSettings: ICommerceApiSettings) {
        this.userAccountNumber = userAccountNumber;
        this.apiSettings = apiSettings;
    }

    public getCacheKey = () => this.userAccountNumber
    public getCacheObjectType = () => 'GetAddress';
    public dataCacheType = (): CacheType => 'request';
}

/**
 * createInput method for the getAddress method
 * @param inputData The input data passed to the createInput method
 */
export const createGetAddressInput = (inputData: ICreateActionContext): IActionInput => {
    const { requestContext } = inputData;
    if (!requestContext.user.isAuthenticated || !requestContext.user.customerAccountNumber) {
        throw new Error('Unable to create address input. User is not authenticated.');
```

# Additional resources

[Chain data actions](#)

[Batch data actions](#)

[Create an observable data action](#)

[Share state across modules](#)

[Data action cache settings](#)

[Data action hooks](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Data action hooks

2/18/2021 • 2 minutes to read • [Edit Online](#)

## IMPORTANT

The data action hooks feature has been deprecated. To avoid performance issues, we strongly recommend that you use [data action overrides](#) to modify business logic in the data action layer.

This topic describes how to hook into pre- and post- data action events to further process data if needed.

## Overview

The modules included with the Dynamics 365 Commerce software development kit (SDK) use pre-existing actions to fetch data for the modules to use. You may have scenarios where you want to change some business logic in the data action layer. In addition to support for data action overrides, the Commerce platform also has the ability to hook into pre- and post- data action events.

## Supported data action hook events

The following data action hook events are supported:

- **preHook**: Runs before the action starts to modify the data action input. This event is only applied to uncached data actions.
- **preReadOnlyHook**: Runs before the action starts but cannot modify the data action input. This event can be applied to both cached and uncached data actions.
- **postHook**: Runs after the data action completes and can modify the data action output. This event is only applied to uncached data actions.
- **postReadOnlyHook**: Runs after the data action completes but cannot modify the data action output. This event is supported on both cached and uncached data actions.

## Use data action hooks

To use a data action hook, you can leverage the **add-data-action** command-line interface (CLI) command to create the file. The following example creates a data action file named "get-address-hook.action.ts" under the "\\src\\actions" directory.

```
yarn msdyn365 add-data-action get-address-hook
```

Next, you replace the code in the data action file with the following template code and then add appropriate code to the event you are interested in.

```
import { createDataActionHook, IActionInput } from '@msdyn365-commerce/core';
// import { Cart } from '@msdyn365-commerce/retail-proxy';

const beforeCart = async (inputs: IActionInput | IActionInput[]) => {
  // tslint:disable-next-line: no-console
  console.info('Pre Hook');
};

const afterCart = async (_inputs: IActionInput | IActionInput[], cartName: string | string[]) => {
  // tslint:disable-next-line: no-console
  console.info('Post Hook');
};

const preReadOnlyCart = async (inputs: IActionInput | IActionInput[]) => {
  // tslint:disable-next-line: no-console
  console.info('Pre ReadOnly Hook');
};

const postReadOnlyCart = async (_inputs: IActionInput | IActionInput[], cartName: string | string[]) => {
  // tslint:disable-next-line: no-console
  console.info('Post ReadOnly Hook');
};

createDataActionHook({
  actionId: 'Action_ID',
  postHook: afterCart,
  preHook: beforeCart,
  preReadOnlyHook: preReadOnlyCart,
  postReadOnlyHook: postReadOnlyCart
});
```

#### NOTE

You'll need to change the **Action\_ID** to the data action ID you are adding event hooks to. The module library list of data actions can be found in the ...\\Msdyn365.Commerce.Online\\node\_modules@msdyn365-commerce-modules\\retail-actions\\dist\\lib directory of the SDK.

## Additional resources

[Chain data actions](#)

[Batch data actions](#)

[Create an observable data action](#)

[Share state across modules](#)

[Data action overrides](#)

[Data action cache settings](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Theming overview

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic presents an overview of online site theming in Microsoft Dynamics 365 Commerce.

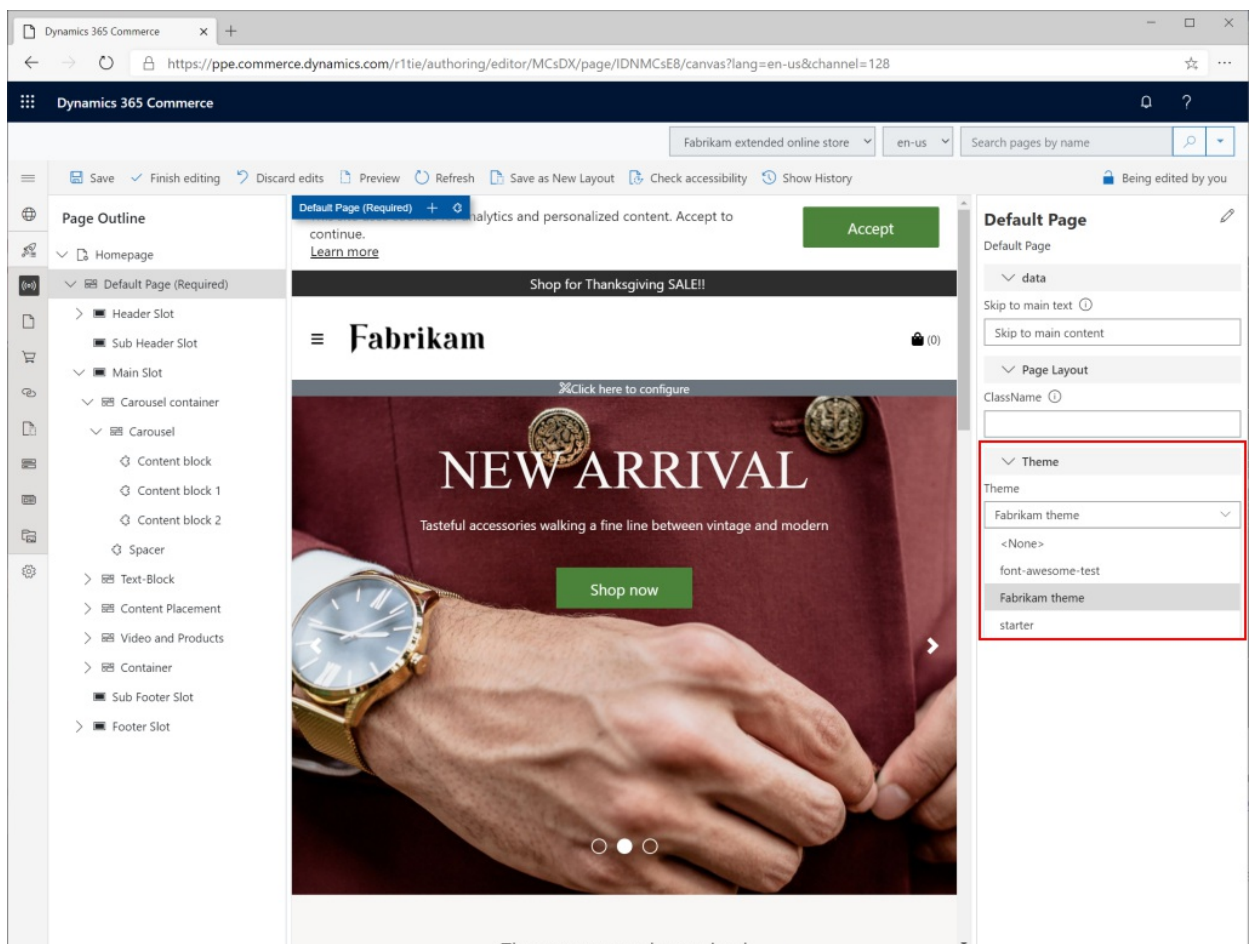
## Overview

Dynamics 365 Commerce lets you apply a theme to your whole online site, individual templates, or individual pages. For example, you might have a default theme that is set for the whole online site and a campaign theme that is applied to just a subset of the pages on the site.

Themes include Sassy Cascading Style Sheets (SCSS) files that you can use to format your site pages and modules. They can optionally also contain module view and definition extensions, so that modules can render different views, depending on the theme that is selected.

After a theme is created and uploaded to your production site, you can use the Commerce site builder tool to set the theme for the site. You can set the site's theme in a template, in a layout, or on a single page. When an online page is rendered, the appropriate theme is applied, so that all the modules on that page have a consistent look and feel. The site builder tool also lets you upload additional Cascading Style Sheets (CSS) overrides. In that way, you can make changes on top of the selected theme.

The following illustration shows how a theme is selected for a page in Dynamics 365 Commerce. Notice that the page container (**Default page**) is selected, and the **Theme** field for the page appears in the properties pane on the right.



The screenshot shows the Dynamics 365 Commerce site builder interface. The main content area displays a page titled "Default Page (Required)" with a promotional banner for "NEW ARRIVAL" featuring a watch. The banner includes the text "Tasteful accessories walking a fine line between vintage and modern" and a "Shop now" button. The left sidebar shows the "Page Outline" with various slots and containers. The right sidebar shows the "Default Page" properties pane, where the "Theme" field is highlighted with a red box, showing "Fabrikam theme" selected from a dropdown menu. Other options in the dropdown include "<None>", "font-awesome-test", and "starter".

A theme can be set on the master page in a similar manner. In this case, the theme is applied to all pages that are

derived from the master page. Note that if the **locked** property is turned off, individual pages can override the theme.

## General guidelines for creating a custom theme

- Create a new theme by using the `yarn msdyn365 add-theme NEW_THEME_NAME` command-line interface (CLI) command. This command will create a theme in the `/src/themes/` folder.
- Under the `styles` directory, you will find the SCSS entry point file for the theme. This file uses the naming pattern `THEME_NAME.theme.scss`.
- Themes are created as special modules. They contain definition files that include the theme's friendly name and description, and also a template React component.
- There is no limit to the number of `.scss` files that your theme can contain.
- Your theme entry point can import other `.scss` files by using relative paths.

## Best practices

- Module library modules are built by using Bootstrap 4 classes. Therefore, we recommend that every theme include either Bootstrap 4 or Bootstrap 4 RTL as the SCSS framework.
- If you want to take advantage of module library modules that are built by using Font Awesome glyph icons, you should include **font-awesome** in the SCSS file. The following example shows how to include **Bootstrap** and **font-awesome** in a SCSS file.

```
$fa-font-path: 'https://use.fontawesome.com/releases/v5.2.0/webfonts' !default;  
@import "bootstrap/scss/bootstrap";  
...
```

## Recommended structure for a custom theme

This section shows the recommended structure for any custom theme.

Import or define the following items:

- Fonts and glyph icons
- Mix-ins and functions:
  - **Bootstrap**: Dependencies, excluding components and utilities
  - **Shared components**: Dependencies, excluding components and utilities
  - Custom theme mix-ins and functions
- Theme variables:
  - Custom theme variables
  - **Bootstrap**: Default theme variables (fallbacks)
- SCSS for components and modules:
  - **Shared components**: Components
  - Custom components and modules
- Utilities:
  - Bootstrap, shared component, and custom utilities

## Hooks for module theming

For every module, a class name is defined that matches the module name. In this way, any theme can target the module. This class name should be the first class name that is applied to the outermost element that is rendered by the React component. To allow for more granular theme options, developers can provide additional class names on elements or features of a module. In that way, custom themes can target those elements or features.

## Custom themes

Custom themes can be created by using the Dynamics 365 Commerce online SDK. They can then be stored in the `/src/themes/` folder. For more information, see [Create a theme](#).

## RTL and LTR support within a theme

You may have requirements to support both right-to-left (RTL) and left-to-right (LTR) languages on your e-Commerce site. Themes support the ability to specify different RTL and LTR SCSS files.

### NOTE

RTL and LTR support within a theme is available in Dynamics 365 Commerce release 10.0.15.

Each theme has a `styles\THEME_NAME.theme.scss` file that is created using the `yarn msdyn365 add-theme` command-line interface (CLI) command. For example, using the command `yarn msdyn365 add-theme spring` to create a new theme called "spring" will create the file `"\src\themes\spring\styles\spring.theme.scss"`, which contains the SCSS code for the theme. SCSS files are compiled into CSS files when using the `yarn start` or `yarn pack` commands, and are then used to render site pages.

To support specific RTL or LTR versions of a SCSS file, you can provide additional files using the following file naming convention: `THEME_NAME.rtl.theme.scss` for RTL support and `THEME_NAME.ltr.theme.scss` for LTR support. When a page renders, the appropriate CSS file will be referenced according to the browser language setting. If you only need support for a single language, use the default `THEME_NAME.theme.scss` file.

## RTL and LTR best practices

Because the CSS code used in RTL and LTR layouts is generally the same except for a few properties, those differing properties can be specified in their respective "theme" files and a "base-style" file can be created to be shared (and imported) by both the RTL and LTR SCSS files.

- `THEME_NAME-rtl.theme.scss` - Contains specific properties for the RTL layout.
- `THEME_NAME-ltr.theme.scss` - Contains specific properties for the LTR layout.
- `base-style.scss` - Contains shared styles and is imported in the two `THEME_NAME` files.

## Additional resources

[Create a new theme](#)

[Configure theme settings](#)

[Configure theme style presets](#)

[Extend a theme to add module extensions](#)

[Override a module library component in a theme](#)

[Extend a theme from a base theme](#)

[Add custom resources to your customization code](#)

[CLI command reference](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a new theme

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to create a new theme for a Microsoft Dynamics 365 Commerce online site.

## Overview

The Dynamics 365 Commerce online software development kit (SDK) lets you create custom themes that can be applied to your online site. They can also be shared across the sites in a single tenant if desired.

After a theme is created and uploaded to your sandbox or production environment, you can use Commerce site builder to set the theme on your site (or sites). Then, when an online page is rendered, the appropriate theme is applied. In this way, modules will have a consistent look and feel across site pages.

Themes contain Sassy Cascading Style Sheet (SCSS) files that you can use to style your online site and individual modules. They can also optionally contain the following extensions:

- Module view extensions that provide new layout views on a module
- Definition extensions to change a module's configurations

## Create a new theme

The online software development kit (SDK) provides the **add-theme** command-line interface (CLI) command. To create a new theme in Commerce, you run the command as shown in the following example, replacing **THEME\_NAME** with the name that you want to give to the new module.

```
yarn msdyn365 add-theme THEME_NAME
```

The following example shows how to create a theme that is named **spring**.

```
yarn msdyn365 add-theme spring
```

The new theme will be created in a new directory under the `...\src\themes` directory. For example, the `spring-time` theme is created under the `...\src\themes\spring` directory.

### Theme naming conventions

Theme names are case-insensitive. The theme's friendly name and description can be added to the theme definition file in the new theme directory.

## Theme definition file

A theme is created as a special module. Each theme has a theme definition file in JavaScript Object Notation (JSON) format that contains the theme's friendly name and description. The **\$type** property will be set to **"themeModule"**.

The following example shows a theme definition file.

```
{
  "$type": "themeModule",
  "friendlyName": "Spring theme",
  "name": "spring",
  "description": "This is default theme."
}
```

## Theme styles

SCSS files are stored under the `...\src\themes\THEME_NAME\styles` directory. By default, this directory includes a `THEME_NAME.theme.scss` file. This file is the entry point SCSS file. You can add other SCSS files and directories to the directory as you require.

For example, the file name and path of the `spring` theme might be `...\src\themes\spring\styles\spring.theme.scss`.

## Test a theme

You can easily test a theme in your development environment by using the `?theme=THEME_NAME` query string parameter.

To test your theme, follow these steps.

1. At a command prompt, under the directory of your local code repository, run `yarn start`.
2. In a web browser, load a module test page, and add the `?theme=THEME_NAME` query string parameter, as shown in the following example.

```
https://localhost:4000/modules?type=product-feature&theme=spring
```

If your `.env` file's `MSDyn365_HOST` entry points to your production site, you can also use the following URL to preview what your site looks like when a custom theme is applied.

```
https://localhost:4000?theme=spring
```

### Mock new configuration values in a theme

If new configuration fields are added to a module in a theme, mock data can be added to the module's mock file. For example, if you modify a module library module's view and definition files, you can add new configuration mocks to the module library mock files that are under the `...\node_modules\@msdyn365-commerce-modules` directory in the module library module.

In a similar way, if you're mocking data for a custom module, you can add new mock data in the module's mock JSON file. Alternatively, you can create new mock files under the same module mock directory.

## Additional resources

[Theming overview](#)

[Configure theme settings](#)

[Configure theme style presets](#)

[Extend a theme to add module extensions](#)

[Override a module library component in a theme](#)

[Extend a theme from a base theme](#)

[Add custom resources to your customization code](#)

## Configure a development .env file

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure theme settings

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to configure theme settings in Microsoft Dynamics 365 Commerce.

## Overview

The Dynamics 365 Commerce e-Commerce extensibility software development kit (SDK) lets theme designers specify various layouts for each module. Theme designers can then control specific layout options for images. The layouts are specified in the theme.settings.json file in /src/settings/src/settings.

## Example theme.settings.json file

The following example shows how module layouts that have specific size values for image settings can be added to the theme.settings.json file. Each layout will be exposed in the authoring tool when the module is configured. The **productFeature** module is configured so that it has two layouts: **layout1** and **layout2**. The **layout1** layout specifies height and width image settings for the **sm** and **lg** viewport sizes.



```

{
  "modules": {
    "productFeature": {
      "properties": {
        "layout1": {
          "friendlyName": "Default Layout",
          "description": "Default Hero Layout",
          "type": "layout",
          "properties": {
            "image": {
              "friendlyName": "Feature Image",
              "description": "Feature Image Settings",
              "type": "imageSizes",
              "properties": {
                "sm": {
                  "width": 767,
                  "height": 431
                },
                "lg": {
                  "width": 1259,
                  "height": 472
                }
              }
            }
          }
        },
        "layout2": {
          "friendlyName": "Banner Layout",
          "description": "Banner Layout without image",
          "type": "layout"
        }
      }
    }
  },
  "gridSettings": {
    "xs": 576,
    "sm": 576,
    "md": 768,
    "lg": 992,
    "xl": 1200
  }
}

```

Default grid breakpoint settings can also be set in the `theme.settings.json` file (see `gridSettings` in the preceding example).

## Additional resources

[Theming overview](#)

[Create a new theme](#)

[Configure theme style presets](#)

[Extend a theme to add module extensions](#)

[Override a module library component in a theme](#)

[Extend a theme from a base theme](#)

[Add custom resources to your customization code](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure theme style presets

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic explains how to add style variables that are known as style presets to custom themes in Microsoft Dynamics 365 Commerce site builder.

## Overview

A style preset is a stored set of all authorable style values across a site theme. Style presets can be used to immediately change the look of a site from within site builder. Style presets let site builder authors quickly change, preview, and activate a set of style values across their site, without having to use Cascading Style Sheets (CSS) or deploy new themes. Font styles, button styles, and site colors are typical examples of style variables that can be managed through style presets.

For more information about how style presets work in site builder, see [Work with style presets](#).

## Default style presets

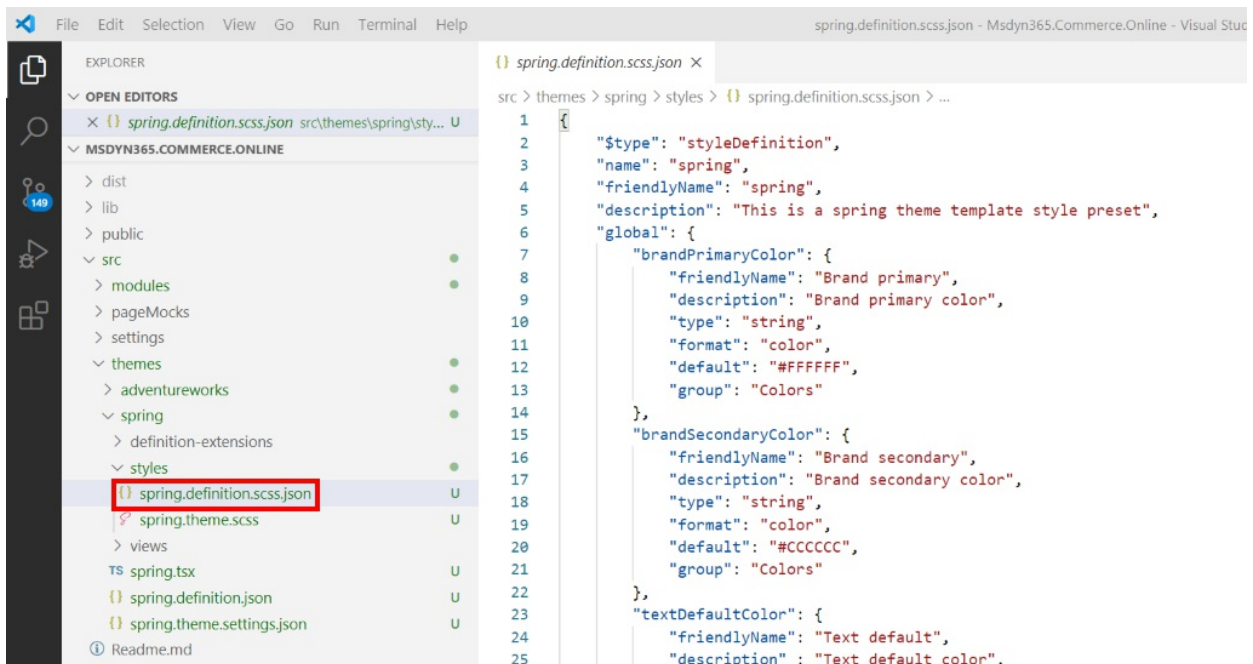
Themes that have authorable style presets must have a default style preset. They can also have additional optional preset settings that are known as preset instances. For example, a theme that has a default "modern light" style might also have optional preset instances such as "modern dark" and "vintage dark."

### Style preset definition files

Each theme contains a style presets definition file that provides metadata for site builder, such as the friendly name and description of the style preset. This definition file also includes the global and module-specific styles that will be available for customization in site builder.

When the **add-theme** [command line interface \(CLI\)](#) command is used to create a theme, a style preset definition file is automatically created under the theme's **styles** directory. The naming convention for the definition file is **THEME\_NAME.definition.scss.json**.

The following illustration shows an example (in Visual Studio Code) of a theme style definition file that was created by using the **add-theme** CLI command.

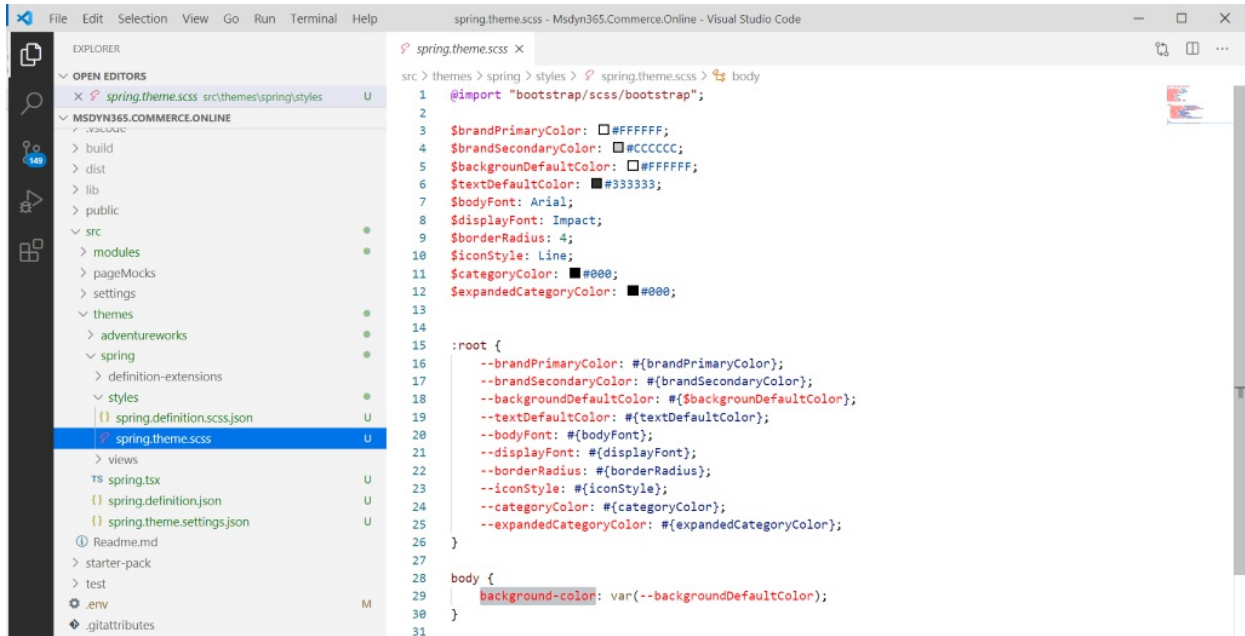


The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project structure with a 'styles' directory containing a file named 'spring.definition.scss.json', which is highlighted with a red box. The main editor area displays the content of this file, which is a JSON object defining a style preset. The file content is as follows:

```
1 {
2   "$type": "styleDefinition",
3   "name": "spring",
4   "friendlyName": "spring",
5   "description": "This is a spring theme template style preset",
6   "global": {
7     "brandPrimaryColor": {
8       "friendlyName": "Brand primary",
9       "description": "Brand primary color",
10      "type": "string",
11      "format": "color",
12      "default": "#FFFFFF",
13      "group": "Colors"
14    },
15    "brandSecondaryColor": {
16      "friendlyName": "Brand secondary",
17      "description": "Brand secondary color",
18      "type": "string",
19      "format": "color",
20      "default": "#CCCCCC",
21      "group": "Colors"
22    },
23    "textDefaultColor": {
24      "friendlyName": "Text default",
25      "description": "Text default color",
```

Each style that is defined under the **global** and **module** sections should also be defined in the theme's Sassy CSS (SCSS) file. The naming convention for that file is **THEME\_NAME.scss**

In the example in the following illustration, SCSS variables have been defined for **brandPrimaryColor** in the **theme.scss** file. The default color is **#FFFFFF**. When the style preset is turned on, the color value is replaced with the default color value that is defined in the style preset definition file. Although both colors happen to be the same in this example, a site builder author can choose to override this property with any color. Modules that use this global SCSS variable will then automatically pick up the color change when the user applies the setting from within site builder.



```
1 @import "bootstrap/scss/bootstrap";
2
3 $brandPrimaryColor: #FFFFFF;
4 $brandSecondaryColor: #CCCCCC;
5 $backgroundDefaultColor: #FFFFFF;
6 $textDefaultColor: #333333;
7 $bodyFont: Arial;
8 $displayFont: Impact;
9 $borderRadius: 4;
10 $iconStyle: Line;
11 $categoryColor: #000;
12 $expandedCategoryColor: #000;
13
14
15 :root {
16   --brandPrimaryColor: #{brandPrimaryColor};
17   --brandSecondaryColor: #{brandSecondaryColor};
18   --backgroundDefaultColor: #{backgroundDefaultColor};
19   --textDefaultColor: #{textDefaultColor};
20   --bodyFont: #{bodyFont};
21   --displayFont: #{displayFont};
22   --borderRadius: #{borderRadius};
23   --iconStyle: #{iconStyle};
24   --categoryColor: #{categoryColor};
25   --expandedCategoryColor: #{expandedCategoryColor};
26 }
27
28 body {
29   background-color: var(--backgroundDefaultColor);
30 }
31
```

### Style preset definition schema

The following style preset definition schema is used for top-level section properties in the style presets definition file:

- **\$type** - The type of definition file. The value of this property must be **"styleDefinition"**.
- **name** – The name of the theme. This property must match the theme name in the theme definition file.
- **friendlyName** – The friendly name of the style preset. This name is shown in site builder when style presets are configured. The minimum length is three characters.
- **description** – The description of the style preset. The description provides a friendly string that is shown in site builder when style presets are configured.
- **global** – The **global** section is used to add style presets that are globally scoped across the theme. CSS property names are used as children of this node to define styles for the CSS style.
- **module** – The **module** section is used to add style presets for specific modules. Module names are used as children of this node to define module-specific style presets.

### Style preset property schema

The following style preset property schema is used for each style property that is defined in the **global** and **module** sections of the style presets definition file:

- **friendlyName** – The friendly name of the individual style preset property. This name is shown in site builder when style presets are configured. The minimum length is three characters.
- **description** – The description of the style preset property. The description provides a friendly string that is shown in site builder when style presets are configured.
- **type** – This property is used as metadata for site builder. The only supported value is **"string"**.
- **format** – This property provides extra metadata to site builder, so that it can present specific user experience (UX) configuration scenarios. This property is optional and currently supports only the value **"color"**, which

is used to open the color picker in site builder.

- **default** – The default CSS style value that is set for this property.
- **group** – This property is used to group similar properties in site builder.
- **enum** – This property is optional and provides a set of hard-coded values that site builder authors can select among.

## Style preset instances

Together with the default style preset settings, a theme can contain one or more optional style preset instances. To create a preset instance file, you must manually create additional style preset definition files under the **styles** directory. The contents of these style preset definition files resemble the contents of the default style preset definition file. However, the default values for the properties in the **global** and **module** sections typically differ. The naming convention for the preset instance file is **PRESET\_INSTANCE\_NAME.scss.json**.

The following example shows a style preset instance file for a dark theme. The file name is **modern-dark.scss.json**.

### NOTE

The value of the **name** property must be unique. It must differ from the name of the default theme and other style preset instances.

```

{
  "$type": "styleDefinition",
  "name": "modern-dark",
  "friendlyName": "modern dark",
  "description": "This is a spring modern light theme template style preset",
  "global": {
    "brandPrimaryColor": {
      "friendlyName": "Brand primary",
      "description": "Brand primary color",
      "type": "string",
      "format": "color",
      "default": "#AAAAAA",
      "group": "Colors"
    },
    "brandSecondaryColor": {
      "friendlyName": "Brand secondary",
      "description": "Brand secondary color",
      "type": "string",
      "format": "color",
      "default": "#CCCCCC",
      "group": "Colors"
    },
    "textDefaultColor": {
      "friendlyName": "Text default",
      "description": "Text default color",
      "type": "string",
      "format": "color",
      "default": "#555555",
      "group": "Colors"
    },
    "backgroundDefaultColor": {
      "friendlyName": "Background default",
      "description": "Background default color",
      "type": "string",
      "format": "color",
      "default": "#000000",
      "group": "Colors"
    },
    ...
  }
}

```

## Localize style preset names and descriptions

Property names and descriptions for the style preset and style preset instance files can be localized by using the **themes** node in the global.json file. The process resembles the process for localizing the module config properties. For more information, see [Localize a module](#).

The following example shows a global.json file that sets various localized style preset properties.

```

{
  "settings" : {...},
  "modules" : {...},
  "themes" : {
    "spring": {
      "friendlyName": {
        "value": "Spring",
        "_value.comment": "Spring theme name"
      },
      "description": {
        "value": "This is the spring theme.",
        "_value.comment": "Spring theme description"
      },
      "styles": {
        "definition": {
          "description": {

```

```

        "value": "This is the Spring theme style preset",
        "_value.comment": ""
    },
    "global": {
        "brandPrimaryColor": {
            "friendlyName": {
                "value": "Primary brand color",
                "_value.comment": ""
            },
            "description": {
                "value": "The primary brand color used across the site.",
                "_value.comment": ""
            },
            "group": {
                "value": "Colors",
                "_value.comment": ""
            }
        },
        "HeaderFontSize": {
            "friendlyName": "Header text size",
            "description": "This is the text default size for header elements",
            "type": "string",
            "enum": {
                "5px": "Small",
                "10px": "Medium",
                "15px": "Large"
            },
            "group": "Typography"
        }
    },
    "modules"
        "header": {
            "categoryColor" : {
                "friendlyName": {
                    "value": "Header background",
                    "_value.comment": ""
                },
                "description": {
                    "value": "This is the background color for the module",
                    "_value.comment": ""
                },
                "group": {
                    "value": "Color",
                    "_value.comment": ""
                }
            }
        }
    },
    "presets": {
        "dark-theme": {
            "description": {
                "value": "This is the dark theme style preset",
                "_value.comment": ""
            }
        },
        "light-theme": {
            "description": {
                "value": "This is the dark theme style preset",
                "_value.comment": ""
            }
        }
    }
}

```

# Additional resources

[Theming overview](#)

[Create a new theme](#)

[Configure theme settings](#)

[Extend a theme to add module extensions](#)

[Override a module library component in a theme](#)

[Extend a theme from a base theme](#)

[Add custom resources to your customization code](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Extend a theme to add module extensions

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic describes how to extend a theme to add module extensions in Microsoft Dynamics 365 Commerce.

## Overview

Dynamics 365 Commerce e-Commerce themes can optionally contain the following module extensions to either the set of modules in the Commerce module library or custom modules:

- Module view extensions that provide new layout views on a module
- Definition extensions that change a module's configurations
- Data action extensions that call additional data actions

## Theme module view extensions

Themes let you include customized module view extensions, which are generally used to change the default layout of a module for a selected theme. These customized module view extensions are supported for both module library modules and custom modules. For example, you might want to add a new button to a module library module to support additional features. By creating a view extension, you can avoid having to use the `clone` command-line interface (CLI) command to create a full copy of the module library module. In some cases, you might want to extend the module definition, and also add more configuration properties, slots, or resources. For more information about how to create module view extensions, see the [Create a module view extension](#) section of this topic.

View extensions are stored under the `...\src\themes\THEME_NAME\views` directory and follow a naming pattern that resembles the naming pattern for module views (`MODULE_NAME.view.tsx`). For example, a view extension might be named `product-feature.view.tsx`. If a view extension exists in the selected theme, the React component will call it instead of the default view file. Therefore, view extensions can be written exactly like a module view that is used for a module.

In general, you might want to examine the existing view file for one of the module library modules before you create a new view on it. You might also want to copy and paste additional code into the existing view file. To view the source code of a module library module view, open the `...\node_modules\@msdyn365-commerce-modules` directory, and look for the module that you're interested in. You might have to fix the file path references for relative path imports.

### Create a module view extension

The online software development kit (SDK) provides the `add-view-extension` CLI command. To create a new module view extension in Commerce, you run the command `yarn msdyn365 add-view-extension THEME_NAME MODULE_NAME`, replacing `THEME_NAME` with the name of the theme that you want to add the view extension to and `MODULE_NAME` with the name of the module that you're extending.

For example, run the following command to add a new file that is named `product-feature.view.ts` under the `spring-theme` theme's view directory.

```
yarn msdyn365 add-view-extension spring product-feature
```

The following example shows a view file extension that was created by using the preceding command.

```
import * as React from 'react';
import { ISampleModuleViewProps } from '../../modules/product-feature/./product-feature';

export default (props: IProductFeatureViewProps) => {
  return (
    <div className='row'>
      <h2>Config Value: {props.config.showText}</h2>
      <h2>Resource Value: {props.resources.resourceKey}</h2>
    </div>
  );
};
```

## Theme definition extensions

You might have scenarios where you must extend the **config**, **slots**, **dataActions**, or **resources** section of a module definition, and then access those sections from the module view extension. Although you can add new configurations, slots, and resources, you can't modify existing ones. However, by using a **disableConfigProperties** section, you can disable the inheritance of some configurations.

Definition extensions are stored under the **definition-extensions** folder. They follow the naming pattern **MODULE\_NAME.definition.ext.json**, where **MODULE\_NAME** is the name of the module that you're extending.

### Create a module definition extension

To create a new definition extension, create a new file under the **/src/themes/THEME\_NAME/definition-extensions** folder that matches the module that you're extending. For example, the path of a definition extension might be **/src/themes/spring-theme/definition-extensions/product-feature.definition.ext.json**.

The following example shows a theme definition extension file where several configurations, slots, and resources have been added, in addition to a new data action. Notice that the **\$type** property must be set to **"definitionExtension"**. In the definition file, you can declare new properties under the **dataActions**, **config**, **slots**, and **resources** nodes, as shown in this example.

```

{
  "$type": "definitionExtension",
  "dataActions": {
    "products": {
      "path": "@msdyn365-commerce-modules/retail-actions/dist/lib/get-simple-products",
      "runOn": "server"
    },
    "cartNameExtension": {
      "path": "../../actions/cart-extension.action",
      "runOn": "server"
    }
  },
  "config": {
    "subTitle": {
      "type": "string",
      "friendlyName": "Sub Title",
      "description": "Sub Title for additional information"
    },
    "favoriteIcon": {
      "type": "image",
      "friendlyName": "Favorite icon",
      "description": "Favorite icon"
    },
    "favoriteIconSettings": {
      "friendlyName": "Favorite icon Settings",
      "description": "Image settings for the favorite icon property",
      "type": "imageSettings"
    }
  },
  "slots": {
    "content": {
      "friendlyName": "Content Slot",
      "description": "This is the content slot",
      "allowedCategories": ["container"],
      "max": 10,
      "min": 0
    }
  },
  "resources": {
    "recommendedLocations": {
      "value": "Recommended Locations"
    }
  },
  "disableConfigProperties": ["subTitle", "imageAlignment"]
}

```

The `disableConfigProperties` section can be used to define configuration fields that should be disabled. Configuration fields that have been disabled don't appear as configurable options in Commerce site builder for the selected module when the theme is set.

When the `yarn start` command is run, a new props file is automatically generated and appears in the definition extension directory. This file includes the merge of the parent module and the extended module, based on the properties and rules that are defined in the definition extension file.

If you're using a definition extension together with a module view extension, you must add the reference to the new automatically generated file in your view file to take advantage of the new definition changes.

The following example shows the addition of the new automatically generated file. This example also imports the module data file if it's required.

```

import * as React from 'react';
import { IProductFeatureViewProps } from '../../modules/product-feature/./product-feature';
import { IProductFeatureProps } from '../../definition-extensions/product-feature.ext.props.autogenerated';

export default (props: IProductFeatureViewProps & IProductFeatureProps<{}>) => {
  return (
    <div className='row'>
      <h2>Config Value: {props.config.showText}</h2>
      <h2>New Config SubTitle Value: {props.config.subTitle}</h2>
      <h2>Resource Value: {props.resources.resourceKey}</h2>
    </div>
  );
};

```

## Data action extensions

After a theme is selected, any additional data actions that are added to the module theme definition extension file are triggered when pages that use the module are loaded. Any data actions that are added to a module theme extension are called before data actions that are defined on the original module.

The return data from a data action call must be declared in a **THEME\_NAME.data.ts** file under the theme's **/views** directory. The following example shows the file structure of a theme that calls a data action that is named **cart-extension.action**. Notice that the new data action is included in the **actions** folder. The example definition extension file that was shown earlier includes an additional data action, **cart-extension**, that is called by using a relative path from within the **\*.definition.ext.json** file.

```

src
├── __actions
│   └── __cart-extension.action.ts
├── __modules
├── __themes
│   └── __spring
│       └── __definition-extensions
│           └── __product-feature.definition.ext.json
│               └── __styles
│                   └── __view
│                       └── __product-feature.data.ts
│                           └── __product-feature.view.tsx

```

Data that is returned from a data action must be assigned to a variable that is declared in the **MODULE\_NAME.data.ts** file under the **views** directory. The name of the variable must match the name that is provided in the **dataAction** section of the module definition extension file. In the following example, notice that the variable name **cartNameExtension** matches the variable name that is provided in the example definition extension file that was shown earlier.

```

import { AsyncResult } from '@msdyn365-commerce/retail-proxy';

export interface ICartExtensionData {
  cartNameExtension: AsyncResult<string>;
  products: AsyncResult<SimpleProduct>[];
}

```

To consume the data that is returned from the new data action, the module view file can now import the new interface, as shown in the following example for **product-feature.view.tsx**.

```
import * as React from 'react';
import { ICartExtensionData } from './product-feature.data';
import { IProductFeatureViewProps } from '../../modules/product-feature./product-feature';
import { IProductFeatureProps } from '../definition-extensions/product-feature.ext.props.autogenerated';

export default (props: IProductFeatureViewProps & IProductFeatureProps<ICartExtensionData>) => {
  return (
    <div className='row'>
      <h2>Config Value: {props.config.showText}</h2>
      <h2>New Config SubTitle Value: {props.config.subTitle}</h2>
      <h2>Resource Value: {props.resources.resourceKey}</h2>
      <h2>Cart Extension Value: {props.data.cartName}</h2>
    </div>
  );
};
```

## Additional resources

[Theming overview](#)

[Create a theme](#)

[Configure theme settings](#)

[Configure theme style presets](#)

[Override a module library component in a theme](#)

[Extend a theme from a base theme](#)

[Add custom resources to your customization code](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Override a module library component in a theme

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to override module library components to allow for customizations in a theme.

## Overview

The Microsoft Dynamics 365 Commerce module library contains a set of TypeScript components that various module library modules use, and that can be overridden. These components consist of helper APIs that contain business logic and other logic to help render the HTML for the module HTML, handle events, and make server-side calls.

If you must change any logic in a component, you can use the command-line interface (CLI) [add-component-override](#) command to override the component for a selected theme. For example, if you want to change the logic for strikethrough pricing, you can override the Price component. Other changes, such as [module view overrides](#), can also be used for visual changes. In rare cases, you might have to [clone a module](#) to make all the changes that you require.

## Component list

To get a list of components, you can use the following CLI command.

```
yarn msdyn365 add-component-override --list-components
```

Component source code can be found under `\node_modules\@msdyn365-commerce\components\src\` inside the directory for the online software development kit (SDK) that is installed. You might want to browse the source code to see what options are available in each component.

The following table lists the components that included as part of the module library. The components might vary, depending on the version of the module library that you use.

COMPONENT	DESCRIPTION
AddToCart	This component is used to add a product to the cart. It works in combination with the cart state that is present in the global state package. This component is rendered as a button on a product details page (PDP). When the button is selected, it validates the inventory check for the product and calls the Retail Server addcartline API to add the product to the cart.
AddToWishlist	This component is used to add a product to, or remove a product from, the wishlist. It works in combination with the wishlist state that is present in the global state package. This component renders the user interface (UI) for the add/remove wishlist button. It also handles the onclick event. The onclick event will ensure that the user is signed in.

COMPONENT	DESCRIPTION
CartIcon	This component is used to render the HTML to show the cart icon in the header module. When the cart icon is selected, this component opens the cart page. The module library also has a CartIcon module that is separate from this component. If the CartIcon module isn't configured in a header module, this component is used as a fallback.
CartLineItem	This component is used to render the HTML to show a single line item in the cart. It consists of a product image, the product details that are selected in the buy box module on the PDP, the product quantity, and the product price with discounts.
Price	This component is used to render the HTML to show the product price across the site in the correct format. It doesn't make any API calls.
Product	This component is used to render the HTML to show the products in a card format across the site. It consists of a product image, title, description, price, and rating.
PromoCode	This component is used to apply promotional codes or discount codes to the cart. It works in combination with the cart state that is present in the global state package to apply or remove a promotion in the cart.
Rating	This component is used to render the HTML to show the product rating across the site. It doesn't make any API calls. This component also handles the click event to update the rating.
SocialMedia	This component is used to render the social media icons for sharing a product from the buy box module. It currently supports Facebook, LinkedIn, Mail, Pinterest, and Twitter.
WishlistIcon	This component is used to render the HTML to show the wishlist icon in the header module. It also handles the click event to open the wishlist page.

## Override a component in a theme

To override a component in a theme, you can run the following CLI command: `yarn msdyn365 add-component-override [themeName] [componentName] [--list-components]`. After this command is run, you should find a new TypeScript file under the theme's `\view\components` directory. You can then modify the file in that directory as you require.

### Example

```
yarn msdyn365 add-component-override spring add-to-cart
```

## Additional resources

[Theming overview](#)

[Create a new theme](#)

[Configure theme settings](#)

[Configure theme style presets](#)

[Extend a theme to add module extensions](#)

[Extend a theme from a base theme](#)

[Add custom resources to your customization code](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Extend a theme from a base theme

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to extend a theme from a base theme for a Microsoft Dynamics 365 Commerce online site.

## Overview

By using the Dynamics 365 Commerce online store extensibility software development kit (SDK), you can create either themes that are standalone themes or themes that are extended from a base theme. For example, you can have a base theme that defines Cascading Style Sheets (CSS) styles for modules, module view extensions, and module definition extensions. You can then have a different theme, or even a set of themes, that adds changes on top of the base theme. This capability is helpful when a single Dynamics 365 environment has multiple online sites that use different theme branding.

## Specify a base theme

To specify the base theme for a theme, edit the theme definition file, and add a **\$ref** section that points to the base theme.

In the following example, the **\$ref** section references the **fabrikam** sample theme that is included as part of the module library.

```
{
  "$ref": "@msdyn365-commerce-modules/fabrikam-design-kit/dist/lib/modules/fabrikam/fabrikam.definition.json",
  "$type": "themeModule",
  "description": "This is SDK template theme module",
  "friendlyName": "adventureworks",
  "name": "adventureworks"
}
```

## Examples

In the following example, the **add-theme** command-line interface (CLI) command is used to create a base theme.

```
yarn msdyn365 add-theme base
```

Here is the definition file for the base theme that is created.

### basetheme.definition.json

```
{
  "$type": "themeModule",
  "description": "This is SDK template theme module",
  "friendlyName": "base",
  "name": "base"
}
```

Next, the CLI command is used to create a theme that is named **extended**.

```
yarn msdyn365 add-theme extended
```

The definition file for the extended theme can now reference the base theme by using a relative path, as shown here.

### extended.definition.json

```
{
  "$ref": "../base/base.definition.json",
  "$type": "themeModule",
  "description": "This is SDK template theme module",
  "friendlyName": "extended",
  "name": "extended"
}
```

### Include base theme styles

By default, base theme styles aren't included in the extended theme. To include the base theme styles, you can add a reference in the **extended.theme.scss** file, as shown here. This example also shows how to add other styles.

### extended.theme.scss

```
@import "../base/styles/base.theme.scss";

body {
  background-color: red;
}
```

The following example resembles the previous example. It shows that you can also add a reference to the base theme in the **extended.definition.scss.json** file, if you want.

### extended.definition.scss.json

```
{
  "$ref": "../base/styles/base.definition.scss.json",
  "name": "extended"
}
```

## Additional resources

[Theming overview](#)

[Create a theme](#)

[Configure theme settings](#)

[Configure theme style presets](#)

[Extend a theme to add module extensions](#)

[Override a module library component in a theme](#)

[Add custom resources to your customization code](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add custom resources to your customization code

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to add custom static resources such as font, image, and Cascading Style Sheet (CSS) files to your software development kit (SDK) customization code so that they can be accessed from within your theme.

Some scenarios require adding custom static resources such as font, image, or CSS files that can be accessed from within a module or a theme. These static files can be added to a `/public` folder within your SDK customization code so that they will be included in the configuration package generated with the [CLI `pack` command](#). Relative paths can then be used to access the resources.

## Example

Resources can be added to the `/public` directory or any subdirectory under this directory. For example, adding a font called "NewFont-Regular" could include adding multiple font files such as "NewFont-Regular.eot", "NewFont-Regular.woff", "NewFont-Regular.ttf", and "NewFont-Regular.svg" to the `/public` directory.

After adding files to the `/public` directory, relative paths in the Sassy CSS (SCSS) file can then be used to point to them. The following SCSS example is taken from a theme that is located in the `/src/themes/spring` folder. When the command-line interface (CLI) command `pack` is used, the SCSS code is compiled into a zip file created in the `/build/public/static/css/spring` directory. A relative path is then used to access the font.

```
@import "bootstrap/scss/bootstrap";

@font-face {
  font-family: 'NewRocker-Regular';
  src: url('.././././NewFont-Regular.eot');
  src: url('.././././NewFont-Regular.eot?#iefix') format('embedded-opentype'),
        url('.././././NewFont-Regular.woff') format('woff'),
        url('.././././NewFont-Regular.ttf') format('truetype'),
        url('.././././NewFont-Regular.svg#NewFont-Regular') format('svg');
}

body {
  font-family: 'NewFont-Regular', Fallback, sans-serif;
}
```

## Dynamic access to the public path

When you deploy your e-Commerce customizations using Microsoft Lifecycle Services (LCS), the physical paths to any files stored in the `/public` directory will change, so you cannot provide static physical paths to these files from within your code. However, there is a helper API named `getAsset` that is available to help you obtain the path.

The following example from a custom module `.tsx` file shows how to use the `getAsset` API to generate a URL to a font file stored in the `/public/webfonts` directory.

```
import { getAsset } from '@msdyn365-commerce/core';

const url = `${getAsset('webfonts/fa-solid-900.woff', this.props.context.request)}`;
```

# Additional resources

[CLI command reference](#)

[Theming overview](#)

[Create a new theme](#)

[Configure theme settings](#)

[Configure theme style presets](#)

[Extend a theme to add module extensions](#)

[Override a module library component in a theme](#)

[Extend a theme from a base theme](#)

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Script injectors

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic covers script injectors that can be used to add scripts to online pages in Microsoft Dynamics 365 Commerce.

## Overview

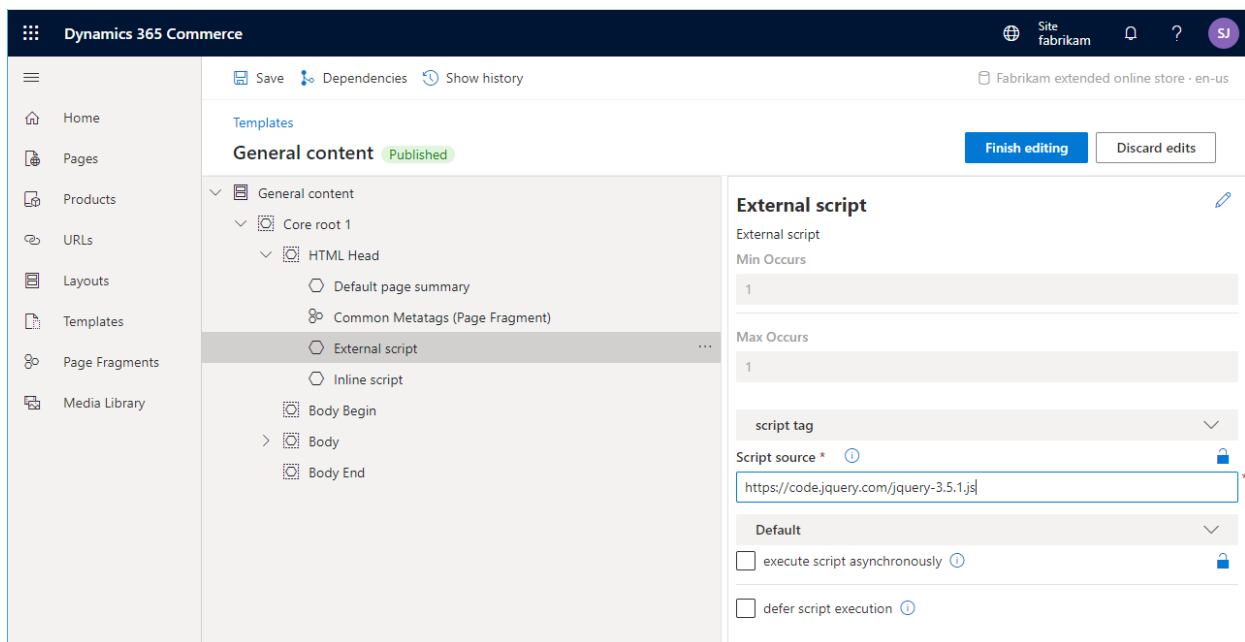
The Dynamics 365 Commerce module library provides two prebuilt script injector modules: **external script** and **inline script**. You can add these modules to a page or page template to inject inline or external scripts into the HTML head, body begin, or body end of a page as needed. For example, you can add a script for integration with third-party analytics, or other service scripts.

## External script module

The external script module allows you to add JavaScript from external sources by providing a URL that points to a valid JavaScript file. After the module is added using Commerce site builder, the script URL can be added to the **Script source** property.

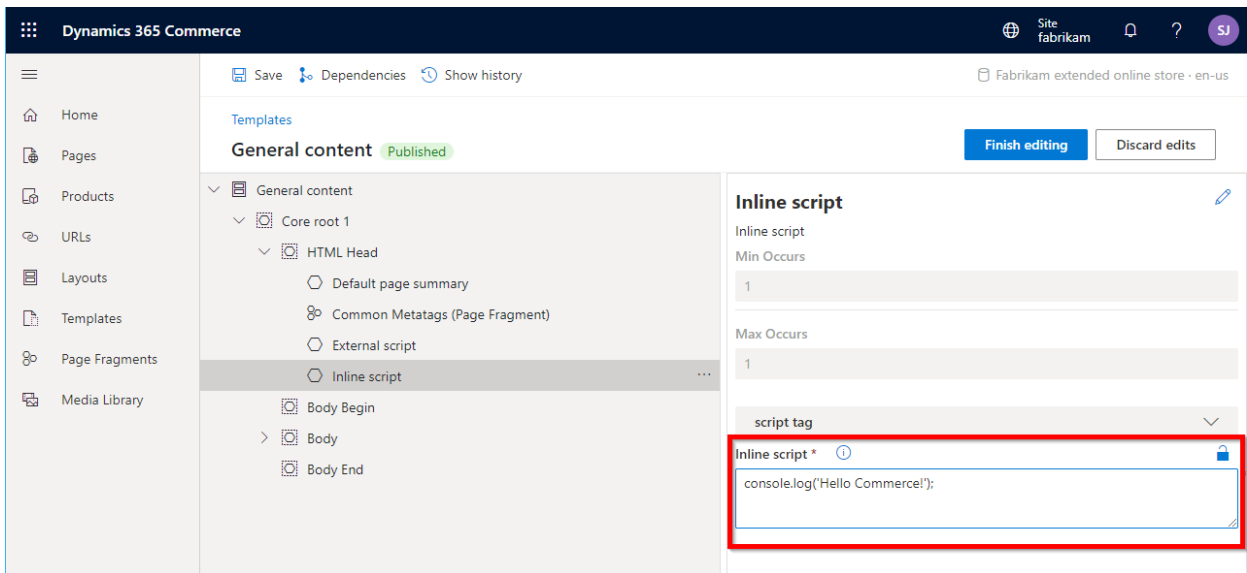
The external script module includes the **execute script asynchronously** and **defer script execution** configuration properties. The **execute script asynchronously** property specifies whether the script should be executed asynchronously. The **defer script execution** property specifies whether the script should be executed when the page has finished parsing configuration properties.

The following illustration shows an external script injector module that is being configured on a page template. The **Script source** property box is where you add the URL that points to the script source code that will be injected into the HTML for the rendered page.



## Inline script module

The inline script module allows you to add inline JavaScript code directly to a page template or page. After the module is added using Commerce site builder, the script can be pasted into the **Inline script** property box, as highlighted in the following illustration.



#### NOTE

When adding script to the **Inline script** property box, you do not need to add the outer `<script>` tags, just the inline JavaScript code.

## Custom script injector modules

In some cases, you might have to inject scripts into your site or site pages, but the prebuilt script injector modules from the module library aren't flexible as you require. For example, you might want additional configuration fields to be exposed in the authoring tools. In these cases, you can extend the prebuilt script injector modules into new custom script injector modules. You can put a custom script injector module on a page, in a shared template, or in a master template.

### Create a custom script injector

To create a custom script injector, use the following command to create a new module.

```
C:\repos\MySite>yarn msdyn365 add-module my-script-injector
```

Next, open the `my-script-injector.definition.json` file and change the `"$type"` property to `"scriptModule"`. After providing a friendly name and description, add the `"Script"` and `"HTML head"` categories, and `"script"` to the `"tags"` property. These categories and tags will allow the script injector to access the applicable page slots already defined inside a page template.

Script modules contain a special `"attributes"` section that defines where the module can be placed. The attributes include `"allowInBodyBegin"`, `"allowInBodyEnd"`, and `"allInHead"`, and the attribute values can be `true` or `false`. The default values for all three attributes are `false`, so it's important to specify where the module can be loaded.

Configurations can also be added to allow the script to be added from within Commerce site builder.

The following example shows a configured custom script injector definition file.

```

{
  "$type": "scriptModule",
  "friendlyName": "My Script Injector",
  "name": "my-script-injector",
  "description": "Used to add custom script to a page.",
  "categories": [
    "Script",
    "HTML head"],
  "tags": ["script"],
  "attributes": {
    "allowInBodyBegin": true,
    "allowInBodyEnd": true,
    "allowInHead": true
  },
  "dataActions": {
  },
  "config": {
    "scriptSource": {
      "friendlyName": "Script source",
      "description": "The script source. Can be an external URL or a relative URL. Relative URLs are
resolved from the public folder",
      "type": "string",
      "group": "script tag",
      "required": true
    },
    "async": {
      "friendlyName": "execute script asynchronously",
      "description": "specifies that the script is executed asynchronously",
      "type": "boolean",
      "default": false
    },
    "defer": {
      "friendlyName": "defer script execution",
      "description": "Specifies that the script is executed when the page has finished parsing",
      "type": "boolean",
      "default": false
    }
  }
}

```

### Modify the script injector view file

You can modify the `my-script-injector.tsx` React file and the `my-script-injector.view.tsx` `Analytics.tsx` view file, if needed.

To create a custom script injector you can use `HtmlHeadInclude`, which is a React [Higher-Order Component](#) that allows you to insert elements into the head of the page. It is simple to use, and you can add any elements you want in your view file between the `<HtmlHeadInclude>` tags, as shown in the following example.



```
import { HtmlHeadInclude } from '@msdyn365-commerce/core-internal';
import * as React from 'react';
import { IMyScriptInjectorViewProps } from './my-script-injector';

export default (props: IMyScriptInjectorViewProps) => {
  const scriptContents = `window.ga=window.ga||function(){(ga.q=ga.q||[]).push(arguments)};ga.l=+new
Date;ga('create', 'UA-XXXX-Y', 'auto');ga('send', 'pageview')`;
  return (
    <HtmlHeadInclude>
      <script data-load-point='headStart'>
        {scriptContents}
      </script>
      <script data-load-point='headStart' async src={props.config.scriptSource} />
    </HtmlHeadInclude>
  );
};
```

#### NOTE

Inline script content should be saved as a string and then inserted into the script, and the **data-load-point** attribute must be specified on script tags. This attribute controls where the script tag should be placed. Possible values include **headStart**, **headEnd**, **bodyStart**, and **bodyEnd**.

The **HtmlHeadInclude** component can also be used to insert **<title>**, **<meta>**, **<link>**, and **<style>** tags into the head of an HTML document. Unlike scripts, these elements do not need a **data-load-point** attribute as they will always be placed in the head.

After a custom script injector module is deployed to a Dynamics 365 Commerce environment, it will appear in site builder.

## Additional resources

[Online channel extensibility overview](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure and enable connectors

2/18/2021 • 11 minutes to read • [Edit Online](#)

This topic describes connectors and explains how to configure and enable them in Microsoft Dynamics 365 Commerce.

Connectors let you connect your Dynamics 365 Commerce site to external third-party services to perform tasks such as capturing analytics, logging, and experimenting. Some third-party service providers require a paid license for their service before it can be used. For more information, contact your service provider.

As of Commerce release version 10.0.13, the only supported type of connector is the experimentation connector. As of Commerce release version 10.0.17, support for the geoLookup connector has been added. In future versions, you will be able to configure and enable other types of connectors.

## Configure and enable connectors

You can add connectors to your Commerce site by adding them as a dependency in your `package.json` file. Alternatively, you can implement them directly in your configuration package code under the `\src\connectors` directory.

### Connector settings file

Connectors are configured and enabled in the `connector.settings.json` file under the `\src\settings` directory. If no `connector.settings.json` file exists, you can manually create one. In this file, you can specify the experimentation connector that you want to use and configure it as you require. Only one experimentation connector can be used at a time.

The following example shows the contents of the `connector.settings.json` file that defines an experimentation connector.

```
{
  "experimentation": {
    "name": "msdyn365-exp-test-2",
    "config": {
      "sdkKey": "EXPERIMENTATION_PROVIDER_KEY",
      "key2": "value2"
    },
    "cacheConfig": {
      "ttlInSeconds": {
        "experimentation": 1800,
        "experimentationDataFile": 300
      },
      "ttrInSeconds": {
        "experimentation": 10
      }
    }
  }
}
```

### Connector settings file schema

- **experimentation** – This object contains all the information that is required to start and enable your experimentation connector.
- **name** – This setting specifies the name of the experimentation connector that is used. You can find the name of the connector in the connector's definition file. The type of the connector must be `experimentationConnector`.

- **config** – This section allows for any configuration object that the connector requires for initialization and to start to communicate with the third-party service. To learn what information is required, look in the **configSchema** section of the connector's definition file or in the connector's README file.
- **cacheConfig** – You can specify the timings that are used when some experimentation-related entities are cached. In this section, **tllInSeconds** refers to the amount of time that an entity can remain in the cache before it's considered stale, and **ttrInSeconds** refers to the amount of time before an entity is refreshed. The connector's README file should include a list of recommended cache timings.
- **experimentation** – This setting controls the cache timings for getting the list of available experiments that are configured in your third-party provider during **getExperiments()**. The default TTL (time to live) is 1,800 seconds, and the default TTR (time to refresh) is 60 seconds.
- **experimentationDataFile** – This setting controls the cache timings for having the configuration passed down to the client during **getConfigForClientSideInit()**. The default TTL is 1,800 seconds, and the default TTR is 60 seconds.

## Experimentation connector

An experimentation connector lets you connect your application to an external experimentation service provider. By adding this type of connector to your application, configuring it, and enabling it, you can create and run experiments in Commerce site builder, and track outcomes, to provide the best experience for your customers.

An experimentation connector consists of three parts:

- A connector definition file (in JavaScript Object Notation [JSON] format)
- A provider file
- A listener file

### Connector definition file

A connector definition file is used to register and provide configuration metadata data to your application. This metadata includes the type of connector, the name of the connector, a description of the connector, and the configuration schema. The name of the connector definition file is in the format **<CONNECTOR\_NAME>.connector.json**.

The following example shows the contents of a connector definition file.

```
{
  "$type": "experimentationConnector",
  "name": "msdyn365-exp-test",
  "description": "Test connector implementation",
  "configSchema": {
    "type": "object",
    "properties": {
      "projectId": {
        "type": "string",
        "description": "The project ID to use for experimentation"
      }
    },
    "required": ["projectId"]
  }
}
```

### Connector definition file schema

- **\$type** – The type of connector. Because the definition file in the preceding example is for an experimentation connector, the type is **experimentationConnector**.
- **name** – The name of the connector. This name must be unique across all connectors.
- **description** – The description of the connector.
- **configSchema** – The configuration schema. A configuration schema lets you provide a JSON schema that

validates the configuration that is given to you at application startup, so that your connector can be initialized correctly. For example, when you initialize your connector, you need the **projectId** value to make an API call that is required for communication with the third-party experimentation service. You can specify this value in the preceding JSON file to ensure that the configuration that is provided matches your connector's requirements.

### **Provider file**

An experimentation provider file is required to initialize a connector. It also enables the connector to interact with Commerce site builder to present a list of available experiments that are configured in your third-party experimentation service. The name of the provider file is in the format **<CONNECTOR\_NAME>.provider.ts**.

The provider file should implement the following interface.

```

export interface IExperimentationProvider {
  /**
   * Allows the experimentation connector to do any startup related tasks
   * using the config provided by the partner.
   *
   * This method is only called once during server startup.
   * @param config The configuration provided in connector.settings.json
   */
  initialize(config: any): Promise<boolean>;

  /**
   * Returns the configuration that should be passed to the experimentation connector
   * when it is initialized client-side
   */
  getConfigForClientSideInit(): Promise<any>;

  /**
   * Initializes the experimentation provider on the browser (client-side) so that
   * it may activate experiments for a user.
   *
   * @param config The config that is required to initialize the experimentation connector
   * client-side. The result of getConfigForClientSideInit() is passed into this method
   */
  initializeClientSide(config: any): boolean;

  /**
   * Returns a list of all the experiments currently configured whether active or not.
   * This list will be cached and periodically refreshed.
   * @param page Optional argument that specifies the page to return.
   * @param items Optional argument that specifies the maximum number of objects to return per request.
   */
  getExperiments(page?: string, items?: string): Promise<IExperiments[]>;

  /**
   * Returns a list of experiments and variants a user will be a part of based
   * off the userId. Optional attributes can provide the connector with additional criteria
   * to determine which experiments a user should be a part of.
   *
   * @param userId userId unique to a user if signed in or unique to a session if user is anonymous.
   * userId will be generated from hash if user is signed-in to deterministically generate sanitized
   * userIds.
   * @param attributes Optional user related attributes
   */
  getVariantsForUser(userId: string, attributes?: {
    [index: string]: string;
  }): IVariants[];

  /**
   * Activates experiment(s) a user is currently being served. This call will be made
   * client-side after the connector has been initialized client-side
   *
   * @param userId userId unique to a user if signed in or unique to a session if user is anonymous.
   * userId will be generated from hash if user is signed-in to deterministically generate sanitized
   * userIds.
   * @param experiments The experiments the user is participating in.
   * @param attributes Optional user related attributes
   */
  activateExperiment(userId: string, experiments: IVariants[], attributes?: {
    [index: string]: string;
  }): boolean;
}

```

In addition, some of the functions use the following types for their return types and arguments.

```

/**
 * Variations on each experiment
 */
export interface IVariations {
  friendlyName: string;
  id: string;
  status: State;
  weight?: string;
}

/**
 * Experiments
 */
export interface IExperiments {
  friendlyName: string;
  id: string;
  status: State;
  variations: IVariations[];
  createdAt?: string;
  lastModifiedDate?: string;
  lastModifiedBy?: string;
  description?: string;
  type?: string;
  link?: string;
  resultLink?: string;
}

/**
 * Experiments
 */
export interface IVariants {
  variantId: string;
  experimentId: string;
  moduleId?: string;
}

```

### Listener file

An experimentation listener file is required to track user conversion events. The listener file implements a logger interface that hooks into the event logging framework of the software development kit (SDK) to subscribe to specific user actions. The name of the listener file is in the format `<CONNECTOR_NAME>.listener.ts`.

The listener file should implement the following interface.

```

export interface IExperimentationListener {
  /**
   * Initializes the experimentation listener on the browser (client-side) so that
   * it may keep track of any conversion events related to the current experiments.
   *
   * @param config The config that is required to initialize the experimentation connector
   * client-side. The result of the provider file's getConfigForClientSideInit() is passed into this
   method
   * @param userId The user ID of the current user being served the experiment
   */
  initializeClientSide(config: any, userId: string): boolean;
  /**
   * Tracks a successful user conversion event.
   *
   * @param eventType The name of the event that occurred
   * @param payload Any additional tags or data related to the conversion event
   * @param attributes Optional parameter containing user attributes pertaining to the user that triggered
   the event
   */
  trackEvent(eventType: string, payload: any, attributes?: any): void;
}

```

## GeoLookup connector

A geoLookup connector lets you connect to an external geolocation service provider. By adding this type of connector to your e-commerce site and configuring it, you can generate geolocation information for e-commerce site users.

### Enable and configure a geoLookup connector

Connectors are enabled and configured in the `connector.settings.json` file under the `\src\settings` directory of the SDK.

The following example shows a geoLookup connector being enabled in the `connector.settings.json` file.

```

{
  "geoLookup": {
    "name": "GeoLocationTest",
    "config": {
      "apiKey": "GEOLOCATION_SERVICE_PROVIDER_API_KEY"
    },
    "cacheConfig": {
      "ttlInSeconds": {
        "geoLookup": 10
      }
    }
  }
}

```

- **geoLookup** – This object contains all the information that is required to start and enable a geoLookup connector.
- **name** – This setting specifies the name of the geoLookup connector. You can find the name of the connector in the geoLookup connector's definition file. The connector type must be `geoLookupConnector`.
- **config** – This section allows for any configuration object that the connector requires for initialization and communication with the third-party service. To learn what information is required, look in the `configSchema` section of the geoLookup connector's definition file, or in the connector's README file. Change the `apiKey` value to a value that is provided by the service provider.
- **cacheConfig** – Here, you can specify the timings that are used when geolocation entities are cached. The `ttlInSeconds` value specifies the amount of time, in seconds, that an entity can remain in the cache before

it's considered stale. The **ttrInSeconds** value specifies the amount of time, in seconds, before an entity is refreshed. The geoLookup connector's README file should include a list of recommended cache timings.

- **geoLookup** – This setting controls the cache timings for getting geolocation information that is generated by the third-party service provider. The default TTL is 120 seconds.

### Anatomy of a geoLookup connector

A geoLookup connector consists of two parts:

- A connector definition file in JSON format
- A provider file

#### Connector definition file

The connector definition file is used to register and provide configuration metadata data to your application. The name of the provider file is in the format **<CONNECTOR\_NAME>.connector.json**. The metadata includes the type of connector, the connector's name, a description of the connector, and the configuration schema, as shown in the following example of a connector definition file.

```
{
  "$type": "geoLookupConnector",
  "name": "msdyn365-geoLookup-test",
  "description": "Test connector implementation",
  "configSchema": {
    "type": "object",
    "properties": {
      "apiKey": {
        "type": "string",
        "description": "Api key for using the geoLookup API"
      }
    },
    "required": ["apiKey"]
  }
}
```

#### Connector definition file schema

- **\$type** – The type of connector. Because the definition file in the preceding example is for a geoLookup connector, the type is **geoLookupConnector**.
- **name** – The name of the connector. This name must be unique across all connectors.
- **description** – The description of the connector.
- **configSchema** – The configuration schema. A configuration schema lets you provide a JSON schema that validates the configuration that is given to you at application startup, so that your connector can be initialized correctly. For example, when you initialize your connector, you need the **projectId** value to make an API call that is required for communication with the third-party service. You can specify this value in the preceding JSON file to ensure that the configuration that is provided matches your connector's requirements.

#### Provider file

A provider file is required to initialize a connector. The name of the provider file is in the format **<CONNECTOR\_NAME>.provider.ts**.

The provider file should implement the following interface.



```

export interface IGeoLookupProvider {
  /**
   * Allows the geoLocation connector to do any startup related tasks
   * using the config provided by the partner.
   *
   * This method is only called once during server startup.
   * @param config The configuration provided in connector.settings.json
   */
  // tslint:disable:no-any
  initialize(config: any): Promise<boolean>;

  /**
   * Geolocation lookup connector will get location information based on the ip address
   * @param ip The ip address
   */
  getGeoInformation(ip: string): Promise<IGeoLocation>;
}

```

In addition, some of the functions use the following types for their return types and arguments.

```

export interface IGeoLocation {
  country?: string;
  region?: string;
  city?: string;
  zipCode?: string;
  [otherProperty: string]: string | undefined;
}

```

Geolocation information that is generated will be saved in the `requestContext.geoLocation` object.

## Additional resources

[Online channel extensibility overview](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Telemetry logger

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic covers the telemetry logger in Dynamics 365 Commerce.

## Overview

The Dynamics 365 Commerce online software development kit (SDK) includes a custom telemetry logger that you can use to log to multiple resources at various levels but also maintain a unified context on both the server and the client.

## Access the telemetry logger

By default, the telemetry logger is available to all react components. It can be accessed with the `this.props.telemetry` property from the module view code file.

Sometimes you will want to access the telemetry logger in a shared component instead of having to pass it down through the properties of every component in your module. The SDK repository includes a `WithContext()` utility that lets you inject the telemetry logger directly into your component.

When the telemetry logger runs in a development environment, all logs are also shown in the console.

## Trace logging

The messages that are logged can range in importance from debug statements to critical errors. This logging is available through the trace logging feature of the telemetry logger. The trace logger uses the following logging methods and levels.

```
enum LogLevel {
    Trace = 'trace',
    Debug = 'debug',
    Information = 'information',
    Warning = 'warning',
    Error = 'error',
    Critical = 'critical',
    None = 'none'
}
```

### Trace logging APIs

The following example shows the primary application programming interface (API) that is used to log a trace message.

```
telemetry.log(
    logLevel: LogLevel,           // One of the log levels listed in the enum above
    messageTemplate: string,     // A string that follows a message template format (see below for
more info)
    logOptions?: TelemetryLogOptions // Object wrapping up the optional parameters for the log statement
)
```

The trace logging system uses message templates to enforce a structured logging system for telemetry. Therefore, the logs can be captured and rendered in both human-friendly and machine-friendly formats. The `messageTemplate` argument of the `.log` call uses what are known as *named holes* for any event data. When

you're writing trace log messages, you can write human-readable messages that include event data. To include event data, mark the place in the message template by using `{<name>}`, where `<name>` is the name that you want to use for the event data. The actual value that will fill that named hole is provided in the `logOptions.values` parameter. The `logOptions` parameter contains any optional parameters that should be included for a trace log. Here is an example.

```
type TelemetryLogOptions = {
  values?: unknown[]; // Holds any arguments that are meant for placeholders in the message template
  customTags?: string[]; // Array of custom tags to add to log. Custom tags can be used to group
  message in the telemetry back-end
  exception?: Error; // Exception that can be attached to the log. Will contain details like stack
  trace info
};
```

The benefit of this structured logging system is that when log messages are rendered so that a human can read them, the system can replace the named holes with the values that have been provided. However, when the messages are sent and stored, or when the telemetry is processed by a machine, the event data can be kept separate. Event data can be used, for example, to aggregate or filter specific messages without requiring any string parsing. For more information about structured logging and message templates, see [Message Templates](#).

The following examples show some trace log calls.

```
telemetry.log(LogLevel.Debug, "{user} says {word}", {values: ["Bill", "Hi!"]});
// Output: "Bill says Hi!"

telemetry.log(LogLevel.Debug, "Customer {customer} purchased item {productID}", {values: [12345, 321]});
// Output: "Customer 12345 purchased item 321"

telemetry.log(LogLevel.Debug, "Module {id} threw error {code} while rendering", {values: [123, 404],
  customTags: ["Module Error"], exception: error});
// Output: "Module 123 threw error 404 while rendering"
// The customTags and exception will be attached to the log call, and can be viewed in the telemetry back-
end
```

There are also wrapper methods that simplify the logging calls. However, these wrapper methods can log only strings. There is one wrapper method for each log level, as the following example shows.

```
this.props.telemetry.trace("This will log at log level 'trace'");
this.props.telemetry.debug("This will log at log level 'debug'");
this.props.telemetry.information("This will log at log level 'information'");
this.props.telemetry.warning("This will log at log level 'warning'");
this.props.telemetry.error("This will log at log level 'error'");
this.props.telemetry.critical("This will log at log level 'critical'");
```

If you want to include additional strings or other objects in the wrapper methods, you can add them to the message by passing them as arguments at the end of the call. All additional arguments will be converted to strings, joined to the first string message, and separated by a comma. Here is an example.

```
this.props.telemetry.trace("This is the first message", "This is the second message", {some object})

// Output will be: "This is the first message,This is the second message,{.toString result of {some
object}}"
```

The console logs are controlled by the `?debug=true` query string.

## Exception logging

Logging of **Error** objects (and classes that inherit from them) can be done through the `.exception(Error)` API. Here is an example.

```
this.props.telemetry.exception(new Error("Something is broken!"));
```

### **.error vs. .exception**

You might be confused about when you should use `.error()` to log an error in your application, and when you should use `.exception()`. This confusion can arise because the names are similar, and because you can use `.error()` to log **Error** objects, by passing the **Error** objects as additional parameters.

The best guidance is to use `.exception()` to log actual **Error** objects and `.error()` to log string messages that state that an error has occurred in the business logic. Generally, `.exception()` logs are more easily correlated with issues and allow for faster debugging when real issues arise. The messages from `.error()` are treated as another trace log, and more detailed analysis might be required to find the issue than if you use `.exception()`. Therefore, it can take more time to recognize that an issue has occurred. In addition, `.exception()` allows for better tracking across different requests. Therefore, it supports features such as automatic alerting when an issue begins to affect many requests.

## Additional resources

[Online channel extensibility overview](#)

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Package configurations and deploy them to an online environment

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how to package configurations and deploy them to your Microsoft Dynamics 365 Commerce online environment.

## Overview

When your local site configurations (modules, data actions, and themes) are ready to be deployed to your online environment, you must package and deploy them by using Microsoft Dynamics Lifecycle Services (LCS).

## Package the local site configurations for upload

The `yarn msdyn365 pack` command creates a package of the local site configurations. You can then use the package to upload the configurations to an online environment via LCS.

The command should be run from the root directory of your local online software development kit (SDK) files. The output of the command is a new zip file in the same directory.

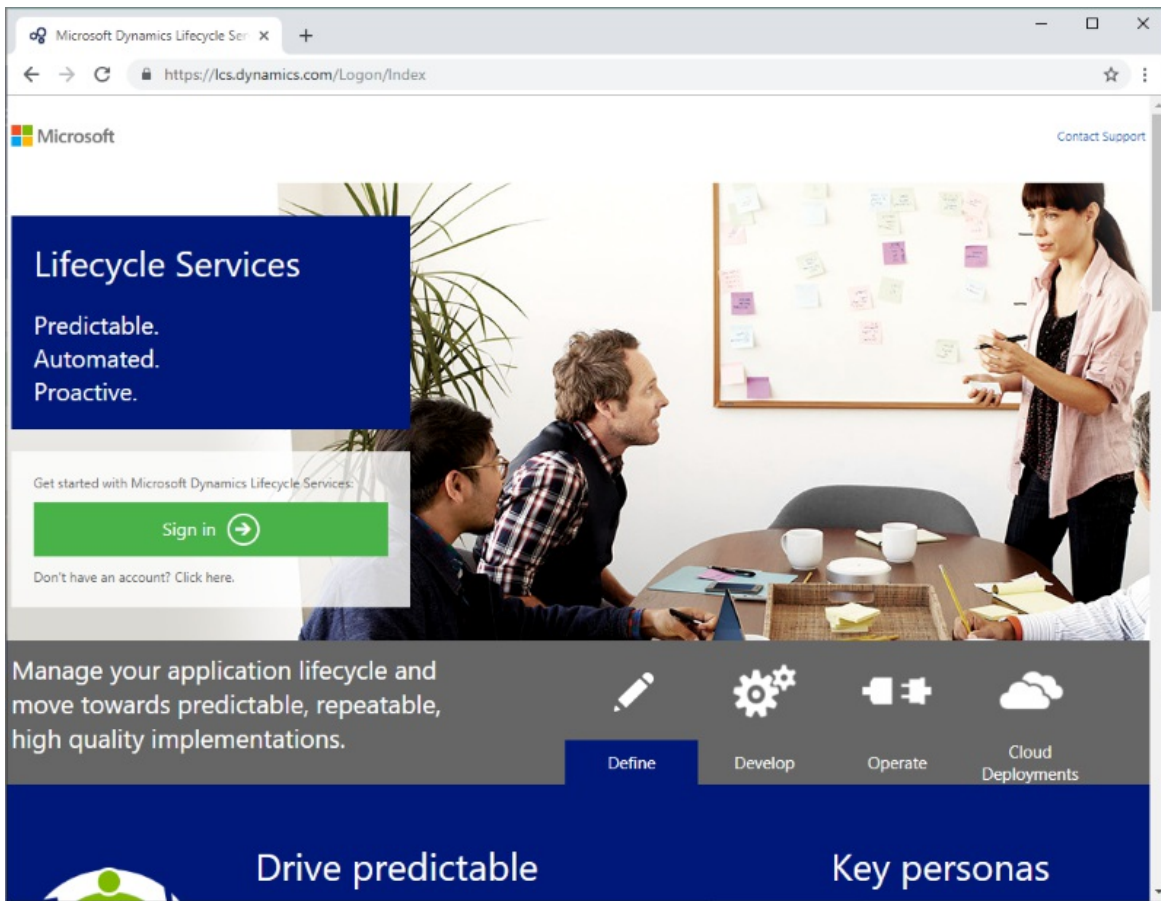
Here is an example.

```
c:\repos\D365.Commerce.Fabrikam>yarn msdyn365 pack
```

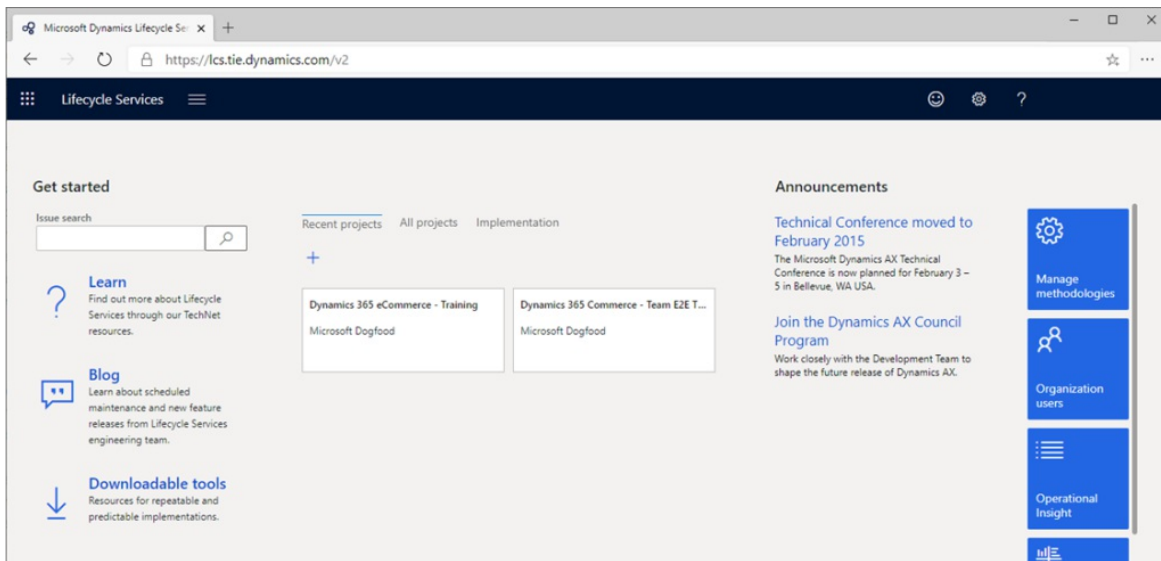
## Upload a package by using LCS

To upload a site configuration package by using LCS, follow these steps.

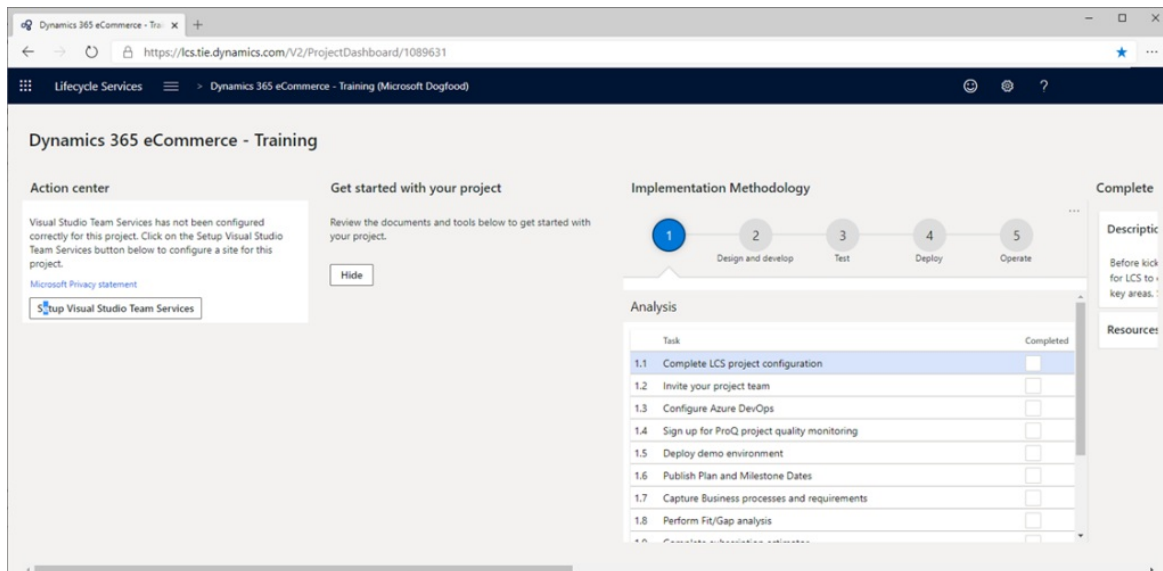
1. Go to <https://lcs.dynamics.com>. (Alternatively, if you're using the test integration LCS server, go to <https://lcs.tie.dynamics.com>). You should see a sign-in page that resembles the following illustration.



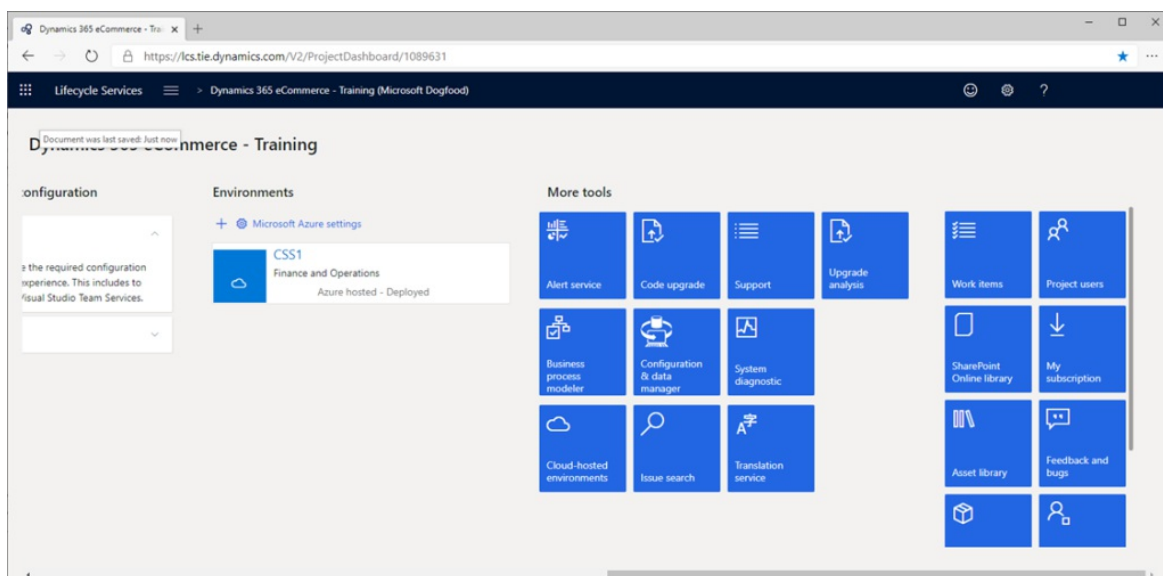
2. Select **Sign in**, and enter your LCS-provided account credentials. The main dashboard appears.



3. Select the e-Commerce project that you will be using. The project dashboard appears.

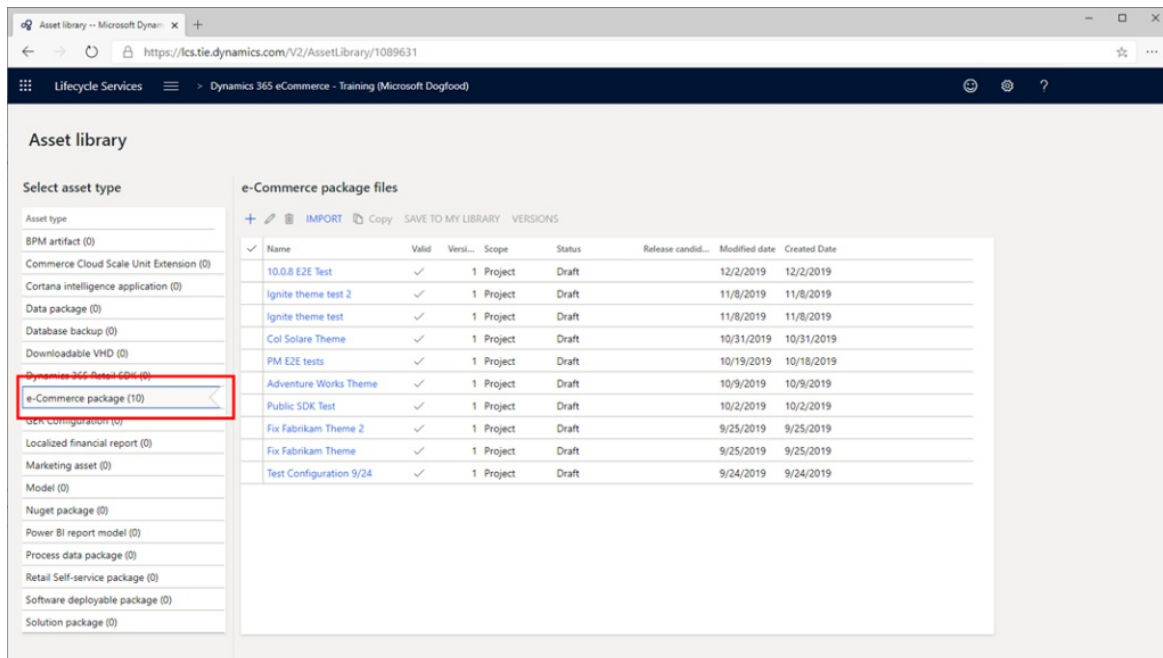


4. Scroll to the right to see more options.

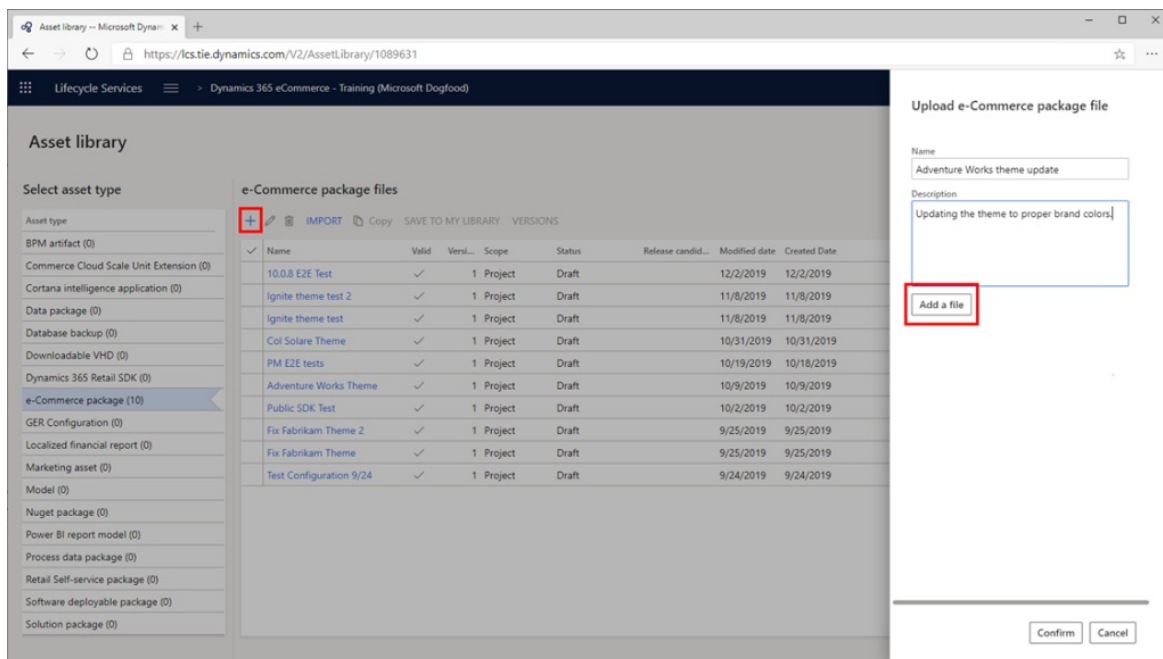


5. To upload the package, in the **More tools** section, select the **Asset Library** tile.

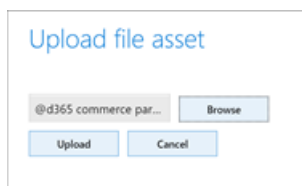
6. On the **Asset library** page, in the left pane, select the **e-Commerce package** tab. If you don't see the **e-Commerce package** tab, you must enable e-Commerce features. Contact your Microsoft Commerce representative to obtain the required code.



7. Select the plus sign (+).
8. In the **Upload e-Commerce package file** dialog box, enter a name and description for the package, and then select **Add a file**.



9. In the **Upload file asset** dialog box, select **Browse**, and browse to the location of the package zip file that you created earlier. Select the file, and then select **Upload**.



10. When the upload is completed, you're returned to the **Upload e-Commerce package file** dialog box. Select **Confirm** to process the upload.



## Upload e-Commerce package file

Name

Description

Upload successful

[Add a file](#)

[Confirm](#) [Cancel](#)

While the upload is being processed, you might have to refresh the page to see status updates. The processing can take between 45 and 50 minutes. When it's completed, a success or failure message is shown.

### Asset library

Select asset type

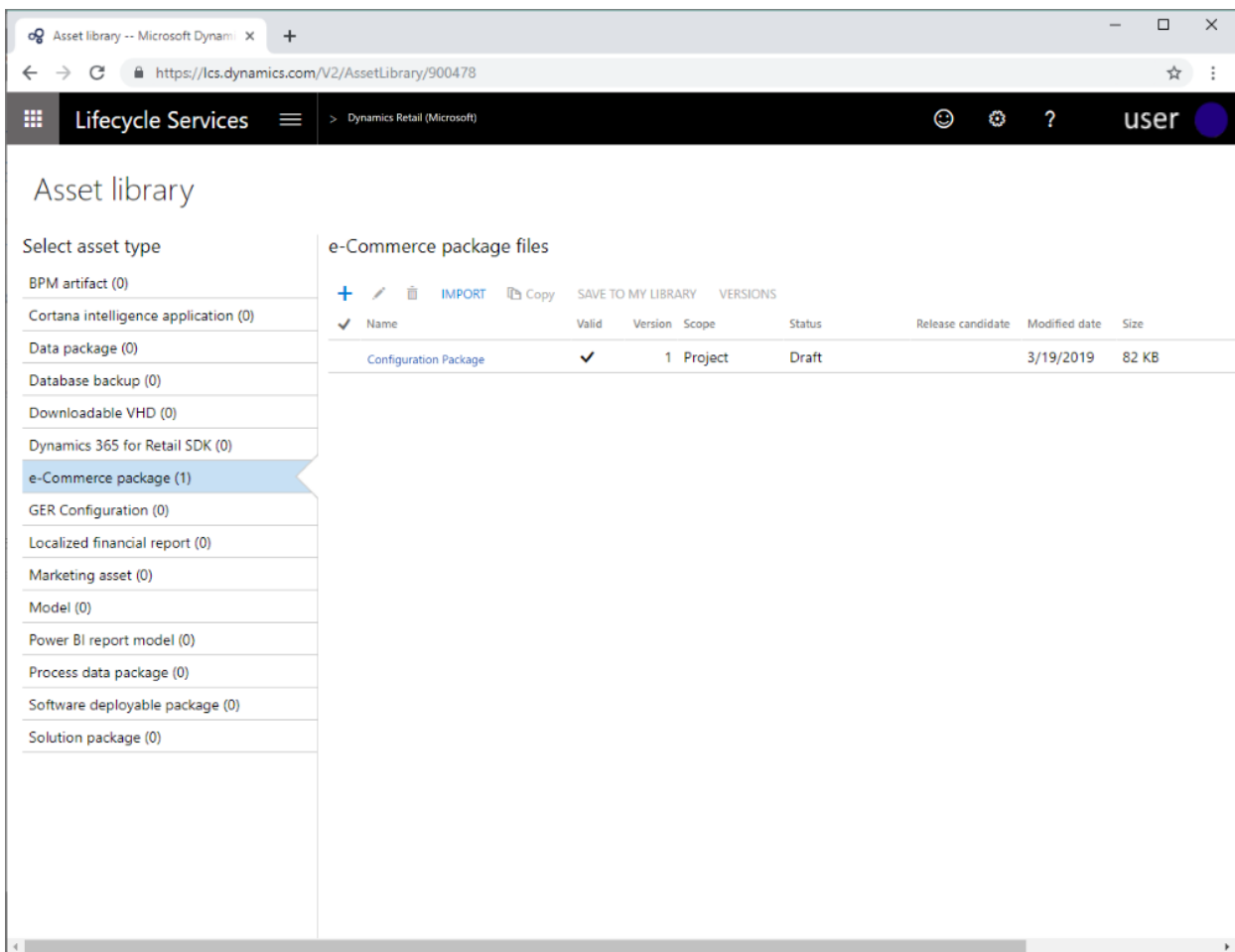
- BPM artifact (0)
- Cortana intelligence application (0)
- Data package (0)
- Database backup (0)
- Downloadable VHD (0)
- Dynamics 365 for Retail SDK (0)
- e-Commerce package (1)**
- GER Configuration (0)
- Localized financial report (0)
- Marketing asset (0)
- Model (0)
- Power BI report model (0)
- Process data package (0)
- Software deployable package (0)
- Solution package (0)

#### e-Commerce package files

[+](#)
[✎](#)
[🗑](#)
[IMPORT](#)
[📄 Copy](#)
[SAVE TO MY LIBRARY](#)
[VERSIONS](#)

✓ Name	Valid	Version	Scope	Status	Release candidate	Modified date	Size
<a href="#">Configuration Package</a>	🔄	1	Project	Draft		3/19/2019	85 KB

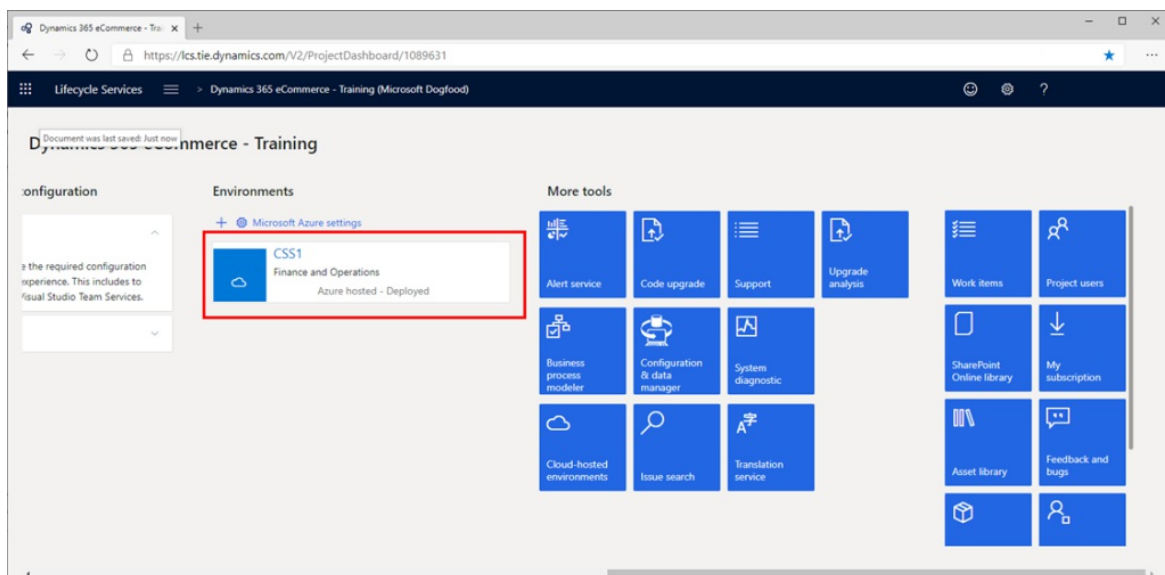
If the processing was successfully completed, a check mark appears in the **Valid** column.



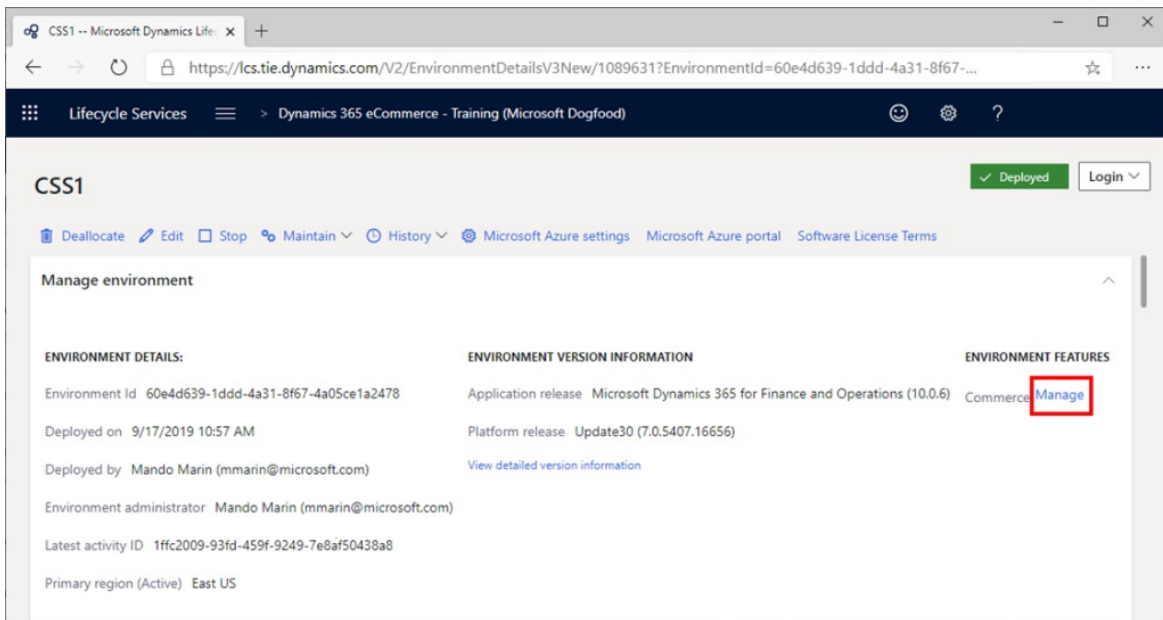
## Deploy a package

To deploy a package, follow these steps.

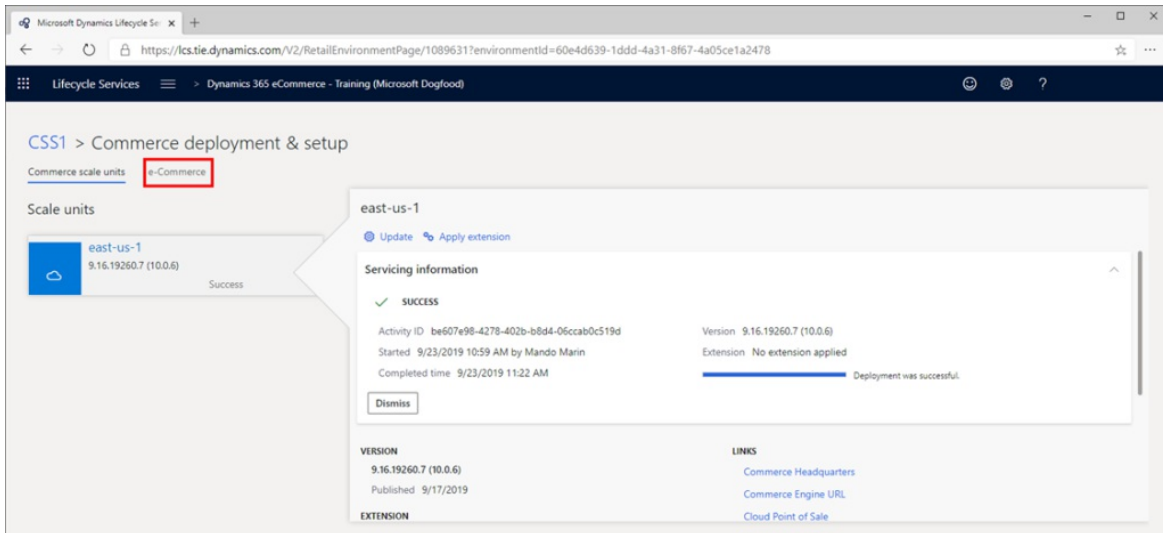
1. In LCS, go to the project dashboard, and select the environment that you want to deploy a package to. For example, in the following illustration, the **RushE2E-TIE-SB3** pre-production environment is selected.



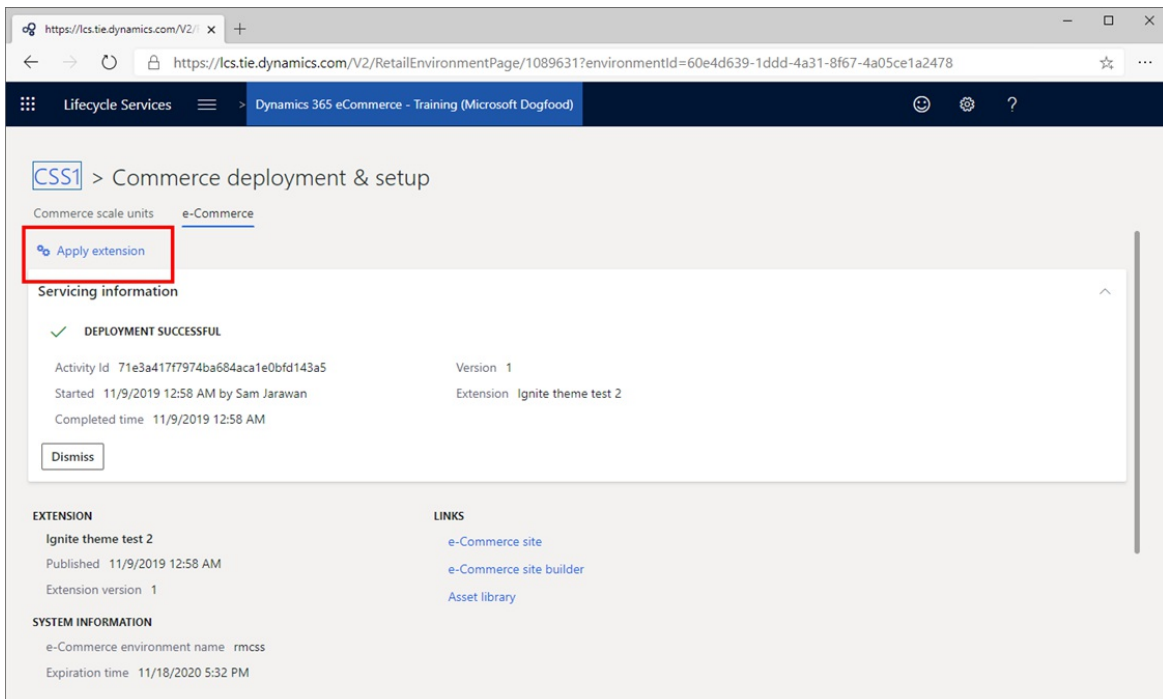
2. In the **Environment features** section on the right side of the page, select **Manage**.



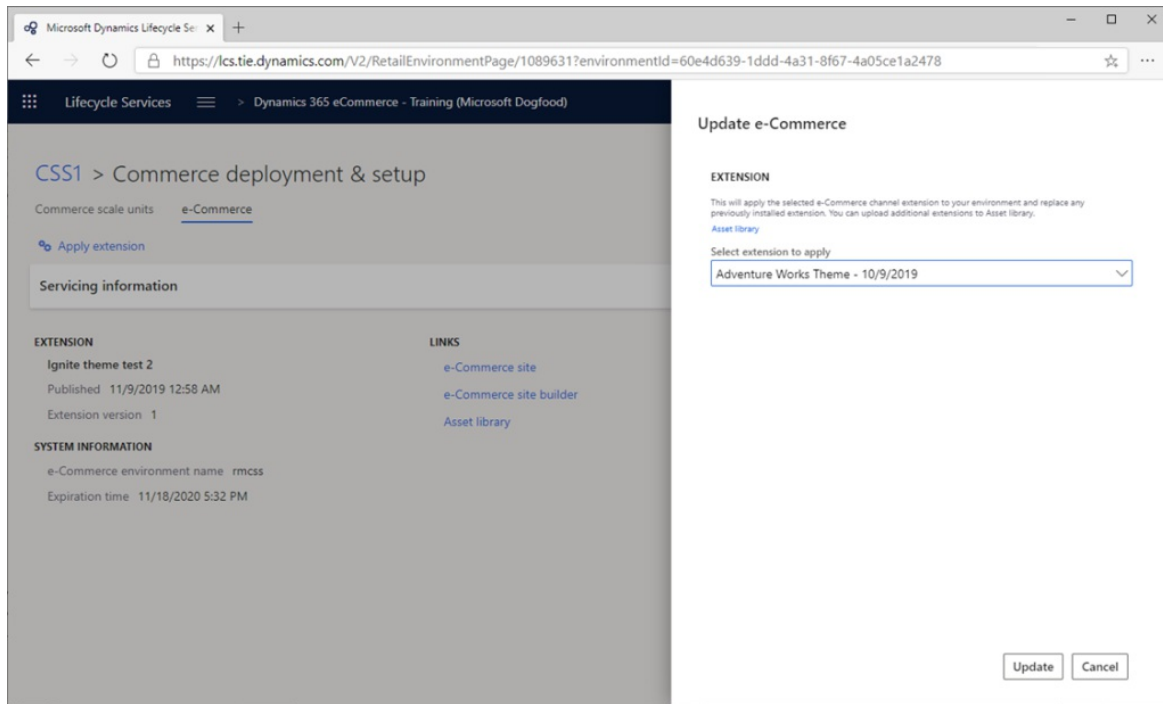
3. Select the E-Commerce (Preview) tab.



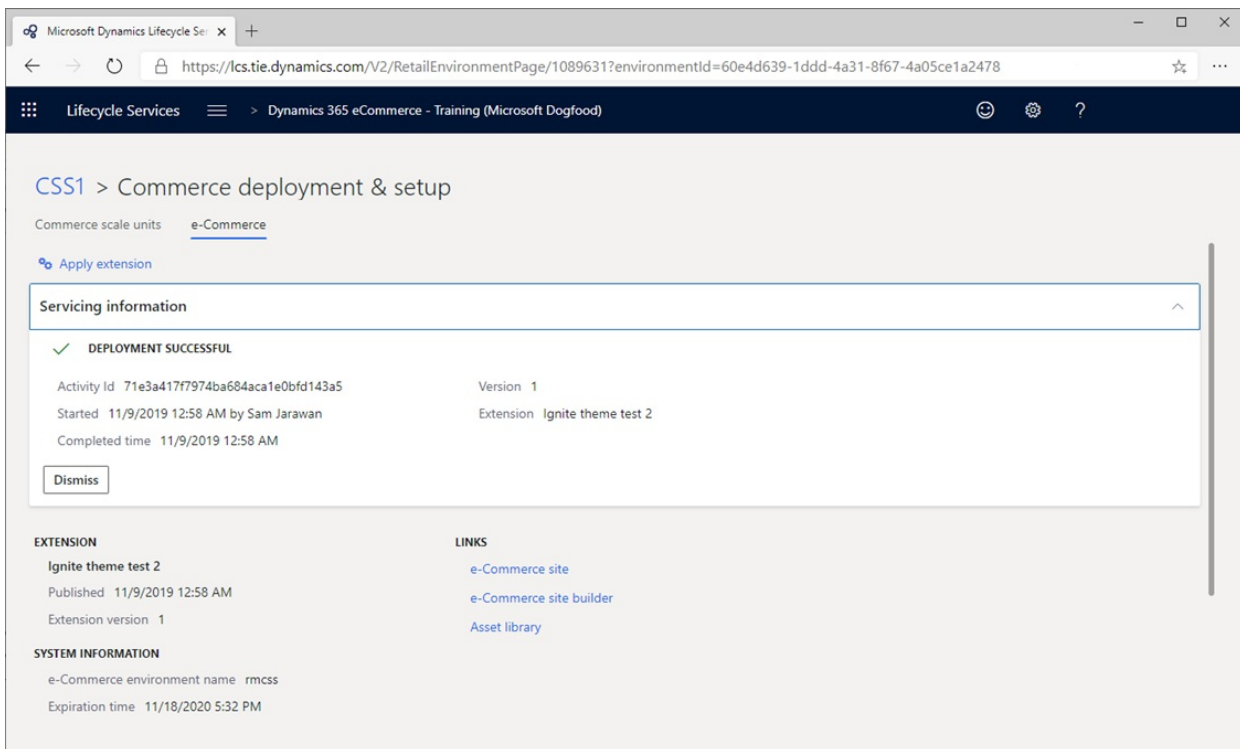
4. Select Apply extension to select the package to deploy.



5. In the **Update e-Commerce** dialog box, select the package that you uploaded earlier, and then select **Update**.



You can now track the deployment status in the **Details** section.



After the deployment is completed, you should see your changes in the authoring tools or on pages that are rendered. For example, new modules or themes will be available to page authors, or changes are rendered and will appear in the online environment.

## Additional resources

[SDK and core library updates](#)

[Microsoft Dynamics Lifecycle Services](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# SDK and module library updates

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic covers regular updates that will be released as part of the Microsoft Dynamics 365 Commerce online software development kit (SDK).

## Overview

Regular updates will be released as part of the Dynamics 365 Commerce online SDK and module library. These updates might include new features or product fixes. Product release notes will be provided for all changes.

## Dependency versions in the packages.json file

The SDK packages.json file that is included in the SDK root directory controls which versions of the SDK, module library, Fabrikam design kit, and Retail Server proxy are downloaded. The specific entries can be found in the "dependencies" section, as shown in the following example. Note that the version numbers might differ, depending on when the SDK was acquired.

```
...
"dependencies": {
  "@msdyn365-commerce-modules/starter-pack": "9.25",
  "@msdyn365-commerce-modules/fabrikam-design-kit": "9.25",
  "@msdyn365-commerce/bootloader": "^1.0.0",
  "@msdyn365-commerce/retail-proxy": "9.25",
}
...
```

- **@msdyn365-commerce-modules/starter-pack** – This entry represents the module library, which includes the set of starter modules and data actions. The preceding example is configured to pull down only the specified version.
- **@msdyn365-commerce-modules/fabrikam-design-kit** – This entry represents the Fabrikam design kit, which includes the Fabrikam theme. The Fabrikam theme defines specific Cascading Style Sheets (CSS) and module view overrides for the set of modules in the module library. The preceding example is configured to pull down only the specified version.
- **@msdyn365-commerce/retail-proxy** – This entry represents the Retail Server proxy, which is used to access the set of APIs for the Commerce Scale Unit. The preceding example is configured to pull down only the specified version.
- **@msdyn365-commerce/bootloader** – This entry represents the SDK. The caret (^) symbol ensures that the **yarn** command always pulls down the latest released version.

The version numbers that are used in the preceding example are in the format *X.Y.Z*, where *X* is the major version, *Y* is the minor version, and *Z* is the patch version.

SDK dependencies are backward-compatible and can be pulled down at any time. The module library minor versions are dependent on the Commerce Scale Unit. Therefore, they can't be higher than the versions that are shown in the table that follows.

Patch versions won't change dependencies on the Commerce Scale Unit. Therefore, they can be updated at any time. The tilde (~) symbol can be used with version numbers to ensure that any patch versions that might include software updates are always pulled down. The following example shows how the tilde is used to pull down the latest patch version.

```

...
"dependencies": {
  "@msdyn365-commerce-modules/starter-pack": "~9.26.0",
  "@msdyn365-commerce-modules/fabrikam-design-kit": "~9.26.0",
  "@msdyn365-commerce/bootloader": "^1.0.0",
  "@msdyn365-commerce/retail-proxy": "~9.26.0",
}
...

```

The following table maps module library versions to Commerce Scale Unit versions. The same module library versions that are mapped to the Commerce Scale Unit should be used for the Retail Server proxy and Fabrikam design kit. The table lists current supported versions as of **January 25, 2021**.

COMMERCE SCALE UNIT VERSION	MAXIMUM MODULE LIBRARY VERSION
10.0.10	9.20.x
10.0.11	9.21.x
10.0.12	9.22.x
10.0.13	9.23.x
10.0.14	9.24.x
10.0.15	9.25.x
10.0.16	9.26.x

## Pull updates

The SDK, module library, and other dependency updates are optional and can be pulled down to a local development environment by using the **yarn** command in the SDK source code. When this command is run, the latest dependencies are pulled down, based on the versions that are specified in the `packages.json` file. Before you run the **yarn** command, you should delete the `yarn.lock` file from the root directory. Optionally, you can also delete the `/node_modules` folder to get a clean set of dependency files.

After the latest dependencies are pulled down, you can run **yarn start** to run the Node server on your development environment and test the new SDK and module library components.

When a configuration package is created by using the **yarn msdyn365 pack** command-line interface (CLI) tool, all dependencies are updated to their local versions during the packaging process. The package that is created can then be uploaded to an online site by using Microsoft Dynamics Lifecycle Services (LCS).

## Determine the latest released versions of the dependency packages

To determine the latest released versions of the dependency packages, follow these feed links:

- [Module library feed](#)
- [SDK feed](#)
- [Retail Server proxy feed](#)

## Determine the versions deployed on an e-Commerce site

To determine the deployed versions of the SDK, module library, and Retail Server proxy that are used on an e-Commerce site, right-click a site page, select **View page source**, and then search for the following strings to find the version number:

- **commerceSDKVersion** – The SDK version.
- **commerceSSKVersion** – The module library version.
- **retailServerProxyVersion** – The Retail Server proxy version.

## Update the app.settings.json file

In the SDK `/src/settings` directory, there's a file called **app.settings.json**. This file surfaces global settings to Commerce site builder that are used by the module library set of modules and can contain additional settings for custom modules. The settings are located in site builder at **Site Settings > Extensions**.

When upgrading the module library, there may be new settings that are applicable and need to be added manually. The latest app.settings.json file can be found in the [online SDK GitHub settings directory](#). This file contains settings for the latest available module library. If you haven't made any additions to the app.settings.json file, you can copy the contents of this file to your version or do a diff and merge the new settings manually. If you're updating to an older module library, you can find the specific app.settings.json file under the specific build branch located on the [GitHub branches page](#). You'll need to manually merge the contents to your app.settings.json file.

When a local configuration package is created by using the **yarn msdyn365 pack** command-line interface (CLI) tool, the local app.settings.json file updates will be included. The package that is created can then be uploaded to an online site by using Microsoft Dynamics Lifecycle Services (LCS). You will then see the new settings in the site builder tool.

## Additional resources

[Package configurations and deploy them to an online environment](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Best practices for Dynamics 365 Commerce development

2/18/2021 • 7 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic describes some best practices to follow when developing Dynamics 365 Commerce customizations.

The Dynamics 365 Commerce platform provides a rich online software development kit (SDK) for developer extensibility. Custom modules, data actions, and themes can be created, or modules from the Commerce [module library](#) can be extended. It is important to consider web site performance when building Commerce customizations. Standard best practices used for website development are applicable including minimizing HTML, JavaScript, and CSS files, and optimizing images.

## Infrastructure setup

The [Dynamics 365 Commerce ecosystem](#) comprises various subsystems including Commerce headquarters, Commerce Scale Unit, the web storefront that includes the Node rendering service, Commerce site builder, and the product recommendations services. For best performance, the various components should all be deployed to the same region. The specific region decision should be made in accordance with the customer's regional location.

## NOTE

Commerce Scale Unit allows you to select a specific region during setup.

## Minimize HTML, CSS, and JavaScript file sizes

The Dynamics 365 Commerce online SDK provides development extensions that use TypeScript and Sassy Cascading Style Sheets (SCSS) files. When a configuration package is built by using the [yarn msdyn365 pack](#) command, or when the Node server is started in a local development environment by using the [yarn start](#) command, the TypeScript files are compiled down to JavaScript files, and the SCSS files are compiled down to Cascading Style Sheets (CSS) files. These files are also minified to help reduce network bandwidth. You should make sure that extra, unused JavaScript and CSS files aren't included in your extension package.

### Reduce JavaScript by excluding unused modules

Dynamics 365 Commerce comes with a large set of modules referred to as the Commerce [module library](#). If there are modules that won't be used on an e-Commerce site, they can be excluded to reduce the JavaScript chunk size. The excluded modules will not be rendered on the live e-Commerce site or available within Commerce site builder when authoring pages.

Modules can be excluded by adding the module name to the **excludeModules** property in the SDK's `"/src/settings/platform.settings.json"` file, as shown in the following example.

```
{
  "excludeModules": ["<EXCLUDED_MODULE_NAME1>", "<EXCLUDED_MODULE_NAME2>"]
}
```

You can verify that the module was successfully excluded by comparing the chunk size displayed after a build, or by testing the module in a development environment. For the latter method, you can confirm that the excluded module is not rendered by using the URL `http://localhost:4000/modules?type=<your-module-name>` (after running the Node server by using the "yarn start" command).

## Optimize images

One of the biggest performance hits to a web page can be the downloading of images. You should use CSS whenever possible to generate images for items such as buttons, but in cases where you need marketing or product images you should upload images to the Commerce site builder [Media Library](#). Images uploaded to the Media Library should be of high quality and resolution to cover all web site usage scenarios. Images served from the Media Library will automatically be resized using an image resizer service.

### Image resizer service

When rendered inside of a module, served images are processed by an image resizer service included in the Commerce rendering engine. This is important for responsive design because as the screen size gets smaller, images will automatically be scaled down to the optimum size for each particular module.

It's important to follow some guidelines to ensure that images are resized correctly. Modules can specify image sizes for particular view ports in the `theme.settings.json` file. For more information, see [Configure theme settings](#).

When building modules with images, the HTML should always include the width and height parameters. If the width and height are not provided, image caching will not be optimized through the image resizer.

### Image types and file sizes

There are three aspects that are important when determining the file size of an image:

- The resolution of the image (width and height).
- How the image is encoded (JPEG, GIF, or PNG).
- The quality parameter (JPEG only).

JPEG uses lossy compression that decreases the file size by discarding image detail. The amount of detail discarded is controlled by the quality parameter, which is a number between 0 and 100, with 100 being the best quality. A lower-quality parameter number results in a lower-quality image, but also a smaller file size.

PNG is a lossless format, so no image detail is lost but the image file size will be larger. For images with text, sharp lines, or color gradients, PNG may be a better choice because the JPEG format may show undesirable artifacts as a result of the lossy compression.

GIF is also a lossless format, but it only supports 256 colors in a single image. For images with text or sharp lines that also don't have many colors, GIF may be a better format choice over PNG or JPEG. GIF also has support for simple animations.

Ultimately, the goal is to find the right balance to maintain image quality while keeping the image size as small as possible.

## Cache configuration

Caching is often used on static content that doesn't often change, such as images, product content, and JavaScript and CSS files. Some scenarios may require custom cache settings to achieve the best performance

results.

### Image caching

The default content delivery network (CDN) cache time for images is set to 5 minutes. This means that after 5 minutes the next request to get a specific image will need to be retrieved from the origin, and so will be slower. Increasing the cache time setting is possible, but it must be done by opening a support ticket.

### Data caching

Product-specific data is cached in the e-Commerce rendering Node layer. Caching times are different for each entity type and can be configured inside of the `cache.settings.json` file in the SDK `"/src/settings/"` directory. For more information, see [Data cache settings](#).

## Browser hint meta tags

Browser hint meta tags such as `preconnect` and `dns-prefetch` can be used to tell browsers what resources are needed ahead of time to render a page. Such tags will cause a browser to initiate network connections before it typically would, to make time-expensive network calls earlier. This is useful for adding additional rendering of image, font, or file endpoints used within a page.

The `preconnect` browser hint meta tag can be used if a page relies on resources coming from external origins or domains to initiate a TCP connection. This will cause DNS lookup, TCP handshake, and TLS negotiation to happen before the reference in the HTML document.

The following is an example of a `preconnect` meta tag hint used in HTML.

```
<link rel="preconnect" href="https://<DOMAIN_TO_PRECONNECT>">
```

The `dns-prefetch` browser hint meta tag can be used if a resource is likely to be navigated to or used on a page. DNS prefetching can resolve a domain's address earlier than usual to avoid this time-expensive step later.

The following is an example of a `dns-prefetch` browser hint meta tag used in HTML.

```
<link rel="dns-prefetch" href="https://<DOMAIN_TO_PREFETCH>">
```

Adding meta tags to a page can be done in Commerce site builder using the **Metatags** module. The **Metatags** module should be added to a page template's "HTML Head" section. For more information, see [Work with templates](#).

## Performance analysis

It's very important that e-commerce site pages be performance tested before they go live. You can use a wide range of existing webpage performance testing tools for this purpose. At a minimum, you can use your web browser's F12 Developer Tools to examine the network loads for individual parts of a page. This approach can help identify any performance bottlenecks so that you can investigate them further.

## Restrict your e-commerce website from loading inside external website HTML iframe elements

The `frame-ancestors` directive can be used to restrict the loading of your e-commerce site inside external website HTML iframe elements. For more information on how to configure this feature in Commerce site builder, see [Manage Content Security Policy](#).

## Additional resources

[Dynamics 365 Commerce architecture overview](#)

[Online channel extensibility overview](#)

[System requirements](#)

[Set up a development environment](#)

[Module library overview](#)

[Configure theme settings](#)

[Data cache settings](#)

[Work with templates](#)

[Manage Content Security Policy](#)

[Platform settings file](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 Commerce online SDK FAQ

2/18/2021 • 3 minutes to read • [Edit Online](#)

## IMPORTANT

Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic summarizes answers to questions frequently asked by users of the Dynamics 365 Commerce online software development kit (SDK).

### **After upgrading to module library version 9.27 (Commerce version 10.0.17 release), buy box module view extensions might generate a compilation error.**

The compilation error is caused by code sharing that is related to the product quick view module that was introduced in the Commerce version 10.0.17 release. Because the quick view module shares lots of functionality with the buy box module, some common components were moved to a common folder, so that the buy box and quick view modules can share the code.

To fix the compilation error, update any import references in the buybox.tsx view file, as shown in the examples that follow.

This example shows the old code for imports in buybox.view.tsx.

```
import { IBuyboxViewProps } from '../..';
import {
  IBuyboxAddToCartViewProps,
  IBuyboxAddToOrderTemplateViewProps,
  IBuyboxAddToWishlistViewProps,
  IBuyboxFindInStoreViewProps,
  IBuyboxKeyInPriceViewProps,
  IBuyboxProductConfigureDropdownViewProps,
  IBuyboxProductConfigureViewProps,
  IBuyboxProductQuantityViewProps,
  IBuyboxShopSimilarLookViewProps
} from './components';
```

This example shows the new code for imports in buybox.view.tsx.

```
import { IBuyboxAddToCartViewProps, IBuyboxAddToOrderTemplateViewProps, IBuyboxAddToWishlistViewProps,
  IBuyboxKeyInPriceViewProps, IBuyboxProductConfigureDropdownViewProps, IBuyboxProductConfigureViewProps,
  IBuyboxProductQuantityViewProps, IBuyboxShopSimilarLookViewProps } from '../..common';
import { IBuyboxViewProps } from './buybox';
import { IBuyboxFindInStoreViewProps } from './components/buybox-find-in-store';
```

### **After upgrading to module library version 9.24 (10.0.14 release), cloned modules that use data actions may display the error, "UserAuthorizationException. Customer account number on the request was wrong".**

The following list of [core data actions](#) have a signature change that moves the user account number parameter to the second parameter (instead of the first) and is now set as an optional parameter. In most scenarios where the user account number is no longer needed, the data action will execute in the context of the current signed-in user. In some custom scenarios where the user account number is different than the user account number of the signed-in user, you can fetch the user account number by using the [get-customer](#) data action and passing it to the data action.

Core data actions with signature changes include:

- **add-address**
- **get-address**
- **get-customer**
- **get-loyalty-card**
- **get-loyalty-transaction-estimation**
- **issue-loyalty**

The module library modules have been updated with the correct calling pattern to the above data actions, so you won't receive any errors in these modules. However, if one of the modules was previously [cloned](#) it will still have the older data action signature and display this error at runtime, "UserAuthorizationException. Customer account number on the request was wrong". The signatures will need to be updated accordingly. One way to resolve this issue is to temporarily clone the module library module again with a new name, then differentiate the new module code with the previously cloned custom module and merge the changes. The temporary module can then be deleted.

**After upgrading to module library version 9.23 (10.0.13 release), cloned modules or view extensions that import "CartlineComponent" or "WishListIconComponent" components may display the errors "export 'CartlineComponent' was not found in '@msdyn365-commerce/components'" or "export 'WishListIconComponent' was not found in '@msdyn365-commerce/components'".**

The "CartlineComponent" and "WishListIconComponent" components have been renamed to "CartLineItemComponent" and "WishListIconComponent" respectively. If the previous component names are used in either a cloned module or a view extension, the build errors mentioned above will be displayed. To fix these issues, update the previous component names to the new component names in the cloned module or view extension code.

## Additional resources

[Core data actions](#)

[Clone a module library module](#)

[Best practices for Dynamics 365 Commerce development](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

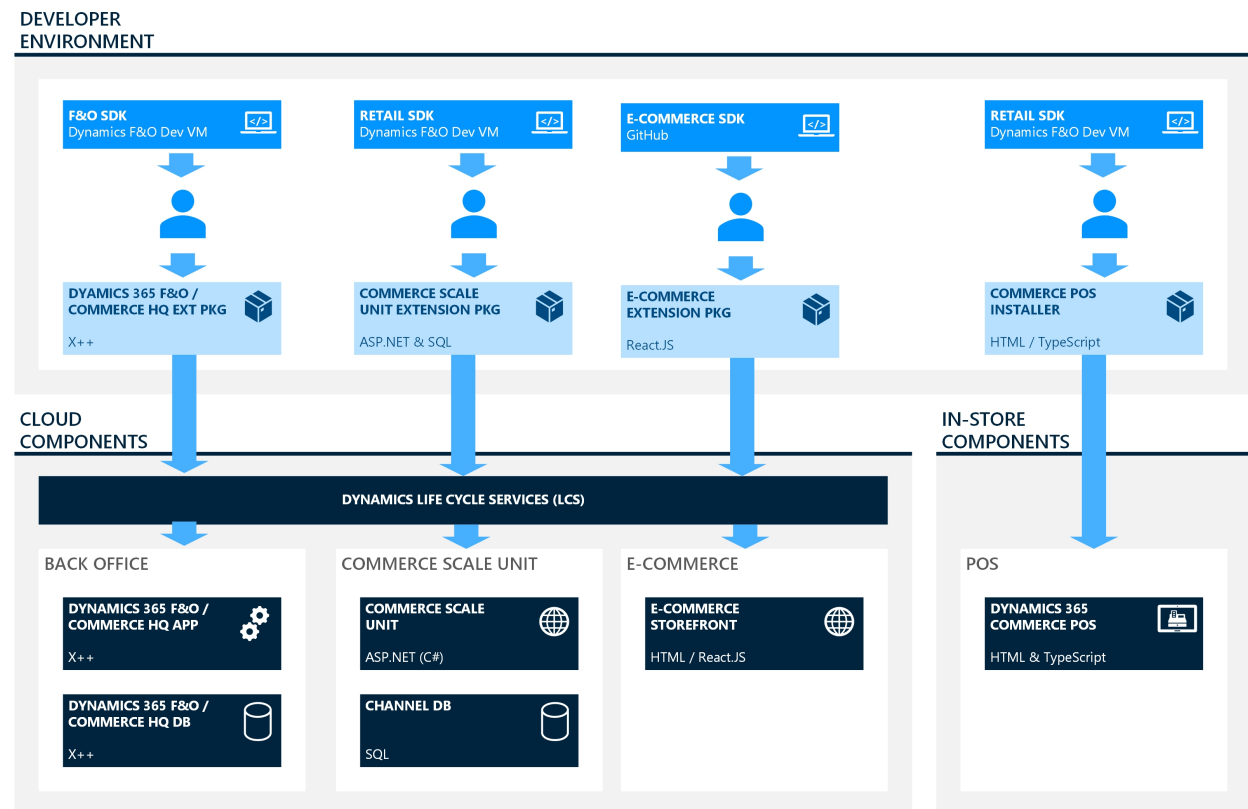
# Retail software development kit (SDK)

2/18/2021 • 15 minutes to read • [Edit Online](#)

This topic provides an overview of the Retail software development kit (SDK). Microsoft Dynamics 365 Commerce provides a rich SDK that developers can use to customize and add new features to the product. The multi-tier architecture of the Dynamics 365 Commerce provides simplified options for customizing and extending the client, business logic, and data layers independently of each other. The Retail SDK includes libraries, NuGet packages, a point of sale (POS) application, code samples, templates, and tools. You can use it to create extensions apps, add features, and change existing functionality of Commerce.

## Retail SDK overview

The Retail SDK includes the code, code samples, templates, and tools that are required to extend or customize existing Commerce functionality. The SDK supports rapid development, full MSBuild integration, and package generation. The following image shows the relationship between the development environment and the cloud components.



### NOTE

The Retail SDK supports the Transport Layer Security (TLS) 1.2 standard. Any customization that you build by using the Retail SDK should follow the TLS 1.2 standard.

## Download the Retail SDK

The Retail SDK is available in development environments that are provisioned via Microsoft Dynamics Lifecycle Services (LCS), in the virtual hard disks (VHDs) that are downloaded from LCS, and in hotfix packages that are deployed to the LCS environment. For more information, see [Deploy and access development environments](#) and

## [Apply updates to cloud environments.](#)

To access the Retail SDK, sign in to the development virtual machine (VM), and go to the K:\RetailSDK folder. You can obtain new versions of the Retail SDK by applying any Commerce binary hotfix from LCS to the development environment. After hotfix deployment is completed, you can find the new version of the SDK inside the K:\RetailSDK\Update folder.

If the current version of the Retail SDK contains extensions, the configuration files and extension projects must be merged from the previous version of the SDK to the new version after an upgrade. This merge is required only if your previous version of the SDK includes extensions, and those extensions must be migrated to the new version. For more information and detailed instructions, see [Upgrade the Retail channel extension to the latest Retail SDK](#). We recommend that you integrate the SDK with a source control system such as Git or Azure.

## Full MSBuild integration

The Retail SDK is a build system. A simple MSBuild command from the root of the SDK folder builds everything. This functionality eliminates questions about how you should build and where you should build from. It also ensures consistency and reproducibility. Therefore, the Retail SDK can easily be integrated with a build pipeline such as Azure Pipelines. For more information, see [Set up Commerce SDK build pipeline](#).

## Prerequisites

To develop or build extensions by using the Retail SDK, you must have the following components:

- Visual Studio 2017 Community, Professional, or Enterprise edition (VM) that has the following components:
  - .NET Desktop development
  - Universal Windows Platform development
  - ASP.NET and web development
  - Azure development
  - Node.js development
  - .NET Core cross-platform development
  - Mobile development with .NET (required for hybrid app development)
- The following runtimes:
  - [sdk-2.1.202-windows-x64-installer](#)
  - [sdk-2.1.513-windows-x64-installer](#)
  - [runtime-2.0.9-windows-x64-installer](#)
  - [runtime-2.1.17-windows-x64-installer](#)
  - TypeScript version 2.2.2

Visual Studio 2017 has TypeScript 3.1 as the default version. You must install version 2.2.2, because the POS app is based on that version. In Visual Studio, select **Tools > Get Tools and Features**. On the **Individual components** tab, select the **TypeScript 2.2 SDK from SDKs, libraries, and frameworks** section, and install it.

## Build the Retail SDK

Before you start development via the Retail SDK, you must restore all the packages by using MSBuild to do a full build from the root of the SDK folder.

1. Open the developer Command Prompt window for Visual Studio 2017 or the MSBuild 15.0 Command



Prompt window.

2. In the Command Prompt window, go to the Retail SDK folder.
3. Run the **msbuild /t:rebuild** command from the root of the SDK folder. The `dirs.proj` file in the root of the SDK folder (`RetailSDK\dirs.proj` or `RetailSDK\Code\dirs.proj`) contains all the details that are required to build the full SDK.

Administrator: Developer Command Prompt for VS 2017

```
K:\RetailSDK>msbuild /p:Configuration=Release_
```

## Retail SDK components

The following table shows the folders that the Retail SDK contains to help with extension development. The folder structure and descriptions in this table are based on Retail SDK version 10.0.13.

FOLDER OR FILE	DESCRIPTION
Assets	This folder contains scripts and configuration files that are required for packaging. Only these configuration files ( <code>HardwareStation.Extension.config</code> , <code>RetailProxy.MPOSOffline.ext.config</code> , <code>CommerceRuntime.Ext.config</code> , and <code>CommerceRuntime.MPOSOffline.Ext.config</code> ) can be edited so that they include extension binary details for packaging. <ul style="list-style-type: none"><li>• <b>manifest.json</b> – The SDK binary version.</li></ul>
BuildTools	This folder contains scripts, sample certificates, and a <code>Customization.settings</code> (packaging metadata) file. Don't change any files in this folder, except <code>Customization.settings</code> .
Database	This folder contains shared database scripts. Extensions must copy the extension scripts to the <code>Database\Upgrade\Custom</code> folder.
Documents	This folder contains instructions for running some of the samples.
OnlineStore	This folder contains the end-to-end sample e-Commerce storefront solution that was built by using the Retail proxy.
Packages	The Retail deployable package that is generated after the SDK build for packaging will be copied to this folder ( <code>Packages\RetailDeployablePackage</code> ). The Retail deployable package is deployed to different environments (test, sandbox, and production) by using LCS.

FOLDER OR FILE	DESCRIPTION
PaymentExternals	<p>Extension payment assemblies must be copied. The following three subfolders hold various payment files:</p> <ul style="list-style-type: none"> <li>• <b>IPaymentProcessor Assemblies</b> – This folder contains the assembly that implements the <b>IPaymentProcessor</b> interface and its dependent assemblies.</li> <li>• <b>Payment Web Files</b> – This folder contains the callback HTML, JavaScript, or Cascading Style Sheets (CSS) files that are required to make the payment accepting page available. Payment connector developers will provide these web files if their payment accepting page requires them.</li> <li>• <b>IPaymentDevice Assemblies</b> – This folder contains the assembly that implements the <b>IPaymentDevice</b> interface and payment request handlers, and the interface's dependent assemblies. These assemblies are used in Retail Hardware station and Retail Modern POS to communicate with payment terminal devices.</li> </ul> <p>Additionally, all extensions that are related to payment connectors should be put in this folder before you create the deployment packages.</p>
Payments	<p>The folder contains the sample Payment Connector project for E-Commerce Add-in for Dynamics 365 Commerce.</p>
pkgs	<p>This folder contains all the NuGet packages (reference libraries) that are required to build the extension projects and tools for packaging and Retail proxy generation.</p>
POS	<p>This folder contains the POS app and extension project:</p> <ul style="list-style-type: none"> <li>• <b>App</b> – Modern POS–specific views and other items.</li> <li>• <b>Contracts</b> – Public contracts for POS extensions. Extension can consume only these contracts for POS extensions.</li> <li>• <b>Extensions</b> – Sample Extension projects and POS.Extension project that extension can consume.</li> <li>• <b>Folder SharedApp</b> – POS views that are shared between Cloud POS and Modern POS.</li> <li>• <b>Folder Web</b> – Cloud POS–specific views and other items.</li> <li>• <b>CloudPos.sln</b> – The Cloud POS solution file.</li> <li>• <b>ModernPos.sln</b> – The Modern POS solution file.</li> </ul>
References	<p>This folder serves as the single location where all binaries are stored. It's used to resolve every project's binary references. The list of files includes external non-Commerce binaries and also Microsoft Commerce binaries. Additionally, this folder serves as the global drop location for any binaries that are built from the Retail SDK.</p>

FOLDER OR FILE	DESCRIPTION
SampleExtensions	<p>This folder contains the sample projects and templates for extensions:</p> <ul style="list-style-type: none"> <li>• <b>CommerceRuntime</b> – Sample extension projects for business logic extensions (Commerce runtime [CRT] triggers, handlers, and new service extension).</li> <li>• <b>HardwareStation</b> – Sample Hardware station extension projects.</li> <li>• <b>HybridApp</b> – Android and iOS shell apps for the POS. Extension can build these apps and deploy them to the Android and iOS platforms.</li> <li>• <b>OnlineStore</b> – The sample online storefront app.</li> <li>• <b>RetailProxy</b> – The sample C# proxy project for POS offline mode. As of version 10.0.11, the C# proxy is obsolete (deprecated). The Retail server extension libraries can be used directly in offline mode. You don't have to have separate proxy libraries.</li> <li>• <b>RetailServer</b> – Sample Retail server extension projects.</li> <li>• <b>SampleExtensionsTest</b> – The sample project for creating an extension test project.</li> <li>• <b>ShoppingApp</b> – The sample mobile app (Retailer shopping app) for users in Android and iOS.</li> <li>• <b>TypeScriptProxy</b> – Sample proxy projects that show how to generate TypeScript for the POS.</li> </ul>
dirs.proj	This project file directs the build order.
Microsoft-version.txt	This file includes the Microsoft application version of the Retail SDK.

## Extension components in the Retail SDK

The following tables provide information about the components in the Retail SDK that must be customized for different scenarios. Only the sample projects inside the RetailSDK\SampleExtensions folder can be changed for extension purposes. No other files or projects/scripts in the Retail SDK should be changed.

### Client (POS)

SCENARIO	Extend the POS for user experience (UX) changes, client logic, workflows, and simple validations.
COMMERCE SDK REFERENCE	<p>\RetailSDK\POS</p> <p>Open the ModernPos.sln or CloudPos.sln file, and add an extension to the POS.Extension project. Don't change anything in the core POS app/web projects.</p>
TECHNOLOGY	TypeScript, HTML, and CSS
DOCUMENTATION	<a href="#">Run the point of sale (POS) samples</a>

### CRT

SCENARIO	Extend CRT to add or change business logic (for example, logic for calculating tax, prices, or discounts).
COMMERCE SDK REFERENCE	\RetailSDK\SampleExtensions\CommerceRuntime Open the CommerceRuntimeSamples.sln file.
TECHNOLOGY	C#
DOCUMENTATION	<a href="#">Commerce runtime (CRT) and Retail Server extensibility</a>

### Retail server

SCENARIO	Create a Retail server extension to expose new Commerce APIs to the client.
COMMERCE SDK REFERENCE	\RetailSDK\SampleExtensions\RetailServer Open any of the sample extensions inside the RetailServer folder.
TECHNOLOGY	Open Data Protocol (OData) and C#
DOCUMENTATION	<a href="#">Create a new Retail Server extension API (Retail SDK version 10.0.11 and later)</a> <a href="#">Create a new Retail Server extension API (Retail SDK version 10.0.10 and earlier)</a>

### TypeScript proxy

SCENARIO	A TypeScript proxy is required if new Retail server extensions must be consumed in the POS or E-Commerce clients.
COMMERCE SDK REFERENCE	\RetailSDK\SampleExtensions\RetailServer Open any of the sample extensions inside the RetailServer folder.
TECHNOLOGY	OData and C#
DOCUMENTATION	<a href="#">Create a new Retail Server extension API (Retail SDK version 10.0.11 and later)</a> <a href="#">Create a new Retail Server extension API (Retail SDK version 10.0.10 and earlier)</a>

### Hardware station

SCENARIO	A Hardware station is required to add or change logic that is related to peripherals.
COMMERCE SDK REFERENCE	\RetailSDK\SampleExtensions\HardwareStation Open the HardwareStationSamples.sln file.
TECHNOLOGY	C#

DOCUMENTATION	<a href="#">Integrate POS with a new hardware device</a>
---------------	--

### Payment connector

SCENARIO	Integrate the POS with a new payment connector.
COMMERCE SDK REFERENCE	<pre>\RetailSDK\SampleExtensions\HardwareStation\Extension.PaymentSample</pre> <p>Open the HardwareStation.Extension.PaymentSample.sln file.</p>
TECHNOLOGY	C#
DOCUMENTATION	<a href="#">Create an end-to-end payment integration for a payment terminal</a>

## Best practices for naming

The C# source code in the Retail SDK uses the Contoso namespace. Therefore, it's easier to distinguish Microsoft types and extension types. If your extension code references a type from the Microsoft binary, use **Microsoft.Dynamics** for the reference, to distinguish between Microsoft libraries and the libraries from the extension. The extension libraries must not begin with the **Microsoft.Dynamics** name.

## Deployment packages

After extension development (CRT, Retail Server, database scripts, POS, and Hardware station), you can use the Retail SDK to generate deployment packages. Packages can be deployed to test, sandbox, and production environments. For more information, see [Create deployable packages](#).

## Dependencies, build order, and full builds

You should build all the extensions and required out-of-box projects. ([Use MSBuild to do a full build from the root of the SDK folder](#).)

- You should build your extension, POS, and packaging projects, but you don't have to build the sample projects that are included in the Retail SDK. You can edit the dirs.proj file in the Retail SDK to remove unwanted sample projects, but don't remove the packaging and POS projects from the list.
- Include the extension project in the dirs.proj file of the appropriate folder. In that way, when you run MSBuild from the root of the SDK folder, all the extension and required out-of-box projects are built.
- The dirs.proj file in the root of the SDK folder is sequenced in the correct order to build all the required projects and then the packaging project. The sequence must be correct. Otherwise, the project and dependencies won't be built correctly.

## Regular configuration/code signing

For Modern POS, create an app package signing certificate to build correctly, or use Cloud POS. For information about how to create a PFX file, see [Create a certificate for package signing](#). Then copy the PFX file to the BuildTools folder, and update the BuildTools\Customization.settings file with the correct name by using the **ModernPOSPackageCertificateKeyFile** element.

The BuildTools\Customization.settings file holds most of the configuration values for the Retail SDK.

```

<!-- This section is for global settings and code signing. Any build file will inherit these values if
applicable.
also use these values during package generation. -->
<AssemblyNamePrefix>MyCompany</AssemblyNamePrefix>
<CustomAssemblyVersion Condition="'$(CustomAssemblyVersion)' == ''>1.0.0.0</CustomAssemblyVersion>
<CustomVersion Condition="'$(CustomVersion)' == ''>1.0.0.1</CustomVersion>
<CustomName Condition="'$(CustomName)' == ''>MyCompany Retail Customization</CustomName>
<CustomDescription Condition="'$(CustomDescription)' == ''>MyCompany Retail
Customization</CustomDescription>
<CustomPublisher Condition="'$(CustomPublisher)' == ''>MyCompany Ltd.</CustomPublisher>
<CustomCopyright Condition="'$(CustomCopyright)' == ''>MyCompany (c) 2015</CustomCopyright>

<SignAssembly Condition="'$(SignAssembly)' == ''>true</SignAssembly>
<DelaySign Condition="'$(DelaySign)' == ''>>false</DelaySign>
<AssemblyOriginatorKeyFile Condition="'$(AssemblyOriginatorKeyFile)' == '' and '$(SignAssembly)' == 'true'">
  $(MSBuildThisFileDirectory)\StrongNameSigningCert-Contoso.snk</AssemblyOriginatorKeyFile>

<ModernPOSPackageCertificateKeyFile Condition="'$(ModernPOSPackageCertificateKeyFile)' == ''>
  $(MSBuildThisFileDirectory)\ModernPOSAppxSigningCert-Contoso.pfx</ModernPOSPackageCertificateKeyFile>

<RetailServerLibraryPathForProxyGeneration Condition="'$(RetailServerLibraryPathForProxyGeneration)' == ''>
  $(SdkReferencesPath)\Microsoft.Dynamics.Retail.RetailServerLibrary.dll</RetailServerLibraryPathForProxyGener
ation>

```

The following values are the global values. These values control how the build manages binaries, components, and how packages are named, versioned, and code-signed.

- **AssemblyNamePrefix**
- **CustomAssemblyVersion**
- **CustomVersion**
- **CustomName**
- **CustomDescription**
- **CustomPublisher**
- **CustomCopyright**
- **SignAssembly**
- **AssemblyOriginatorKeyFile**
- **ModernPOSPackageCertificateKeyFile**
- **RetailServerLibraryPathForProxyGeneration**

It's a good practice to sign your assemblies by using a strong name, even though a strong name isn't required. For information about how to create your own key file if you don't already have one, see [How to: Create a public-private key pair](#).

The installer files that are generated for self-service components such as Modern POS, Hardware station, and Store scale unit can be signed by using [SignTool.exe](#).

Both the key file for the strong name and the app package signing certificate can be stored inside the BuildTools folder or in Azure Key Vault. For a password-protected or secured certificate, use Azure Key Vault.

## Customizing the build

### Adding new projects

It's easy to add new projects to the Retail SDK's build system. You can either clone one of the many existing projects or start a new project. You just have to make some adjustments in a text editor, as shown in the following illustration. The relative path of the **Import** elements should be adjusted, and the **AssemblyName** element should use the predefined **AssemblyNamePrefix** property. These adjustments are required to get

various tasks for free, such as versioning, code signing, uniform assembly naming, and automatic dropping to the References folder.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/d
3 <Import Project="..\..\..\BuildTools\Microsoft.Dynamics.RetailSdk.Build.props" />
4 <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props"
5 <Import Project="..\..\..\BuildTools\Microsoft.Dynamics.RetailSdk.Build.settings" />
6 <PropertyGroup>
7 <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
8 <Platform Condition="'$(Platform)' == ''">AnyCPU</Platform>
9 <ProjectGuid>{2CB843E4-8BEB-4A57-A72B-66D1849FFED6}</ProjectGuid>
10 <OutputType>Exe</OutputType>
11 <RootNamespace>Contoso.Retail.Ecommerce.Publishing</RootNamespace>
12 <AssemblyName>$(AssemblyNamePrefix).Retail.Ecommerce.Publishing</AssemblyName>
13 <ShippingSourceCode>true</ShippingSourceCode>
14 <TargetFrameworkProfile />
15 <TargetFrameworkVersion>v4.5.1</TargetFrameworkVersion>
16 <DefaultLanguage>en-US</DefaultLanguage>
17 </PropertyGroup>
18 <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
27 <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
35 <ItemGroup>
11 <ItemGroup>
16 <ItemGroup>
22 <ItemGroup>
39 <ItemGroup>
49 <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" Condition="'$(WindowsApp
50 <Import Project="$(SdkRootPath)\BuildTools\Microsoft.Dynamics.RetailSdk.Build.targets" /
51 </Project>
```

### Changing the build order or adding to the build

MSBuild traversal files (dirs.proj files) are used to build the whole directory tree of the Retail SDK. The following illustration shows the main traversal file of the Retail SDK. Similar files might also exist in subdirectories. Notice that Visual Studio solution files (.sln files) are similar to traversal files. Both types of file direct the MSBuild engine to process other build scripts.

```
1 <Project ToolsVersion="14.0" DefaultTargets="Build" BuildInParallel="false" xmlns
2 <Import Project="BuildTools\Microsoft.Dynamics.RetailSdk.Build.props" />
3 <Import Project="BuildTools\Microsoft.Dynamics.RetailSdk.Build.settings" />
4 <ItemGroup>
5 <ProjectFiles Include="CommerceRuntime\CommerceRuntime.sln" />
6 <ProjectFiles Include="HardwareStation\HardwareStation.sln" />
7 <ProjectFiles Include="OnlineStore\OnlineStore.sln" />
8 <ProjectFiles Include="Payments\PaymentSdk.sln" />
9 <ProjectFiles Include="SampleExtensions\dirs.proj" />
10 <ProjectFiles Include="Extensions\dirs.proj" />
11 <ProjectFiles Include="Proxies\dirs.proj" />
12 <ProjectFiles Include="Pos\ModernPos.sln" />
13 <ProjectFiles Include="Pos\CloudPos.sln" />
14 <ProjectFiles Include="Packages\dirs.proj" />
15 </ItemGroup>
16 <Import Project="$(SdkRootPath)\BuildTools\Microsoft.Dynamics.RetailSdk.Buil
17 </Project>
```

After new code is added, most of it should be put in a new folder. You must also add it to the traversal structure by adding it to one or more dirs.proj files. In the previous illustration, the Extensions folder is highlighted on line 10. The quickest way to get started with a new dirs.proj file is to copy an existing file, correct the paths in the Import elements, and update the ProjectFiles elements in the ItemGroup element.

### Customizing build scripts

When you must implement new build steps, remember that the existing scripts might be updated by a Retail SDK update later. The best practice is to minimize edits to any file, or to add new files instead. If you require new global MSBuild properties, the BuildTools\Microsoft.Dynamics.RetailSDK.Build.props file is a good place to add them. Likewise, the BuildTools\Microsoft.Dynamics.RetailSDK.Build.targets file can be used to add new build processing targets.

If only one project requires special handling, it's better to explicitly make the change there. If you require new

local MSBuild properties, add a local.props file in the same directory. Alternatively, if you require local build processing targets, add a local.targets file.

## Developer productivity

The **CommerceRuntime** and **RetailServer** extension dynamic-link libraries (DLLs) must be copied into the bin folder of the locally installed RetailServer web application. Users can configure the Customization.setting file so that the DLLs are automatically copied into the bin folder of the local RetailServer web application whenever new versions of these files are built from the extension project.

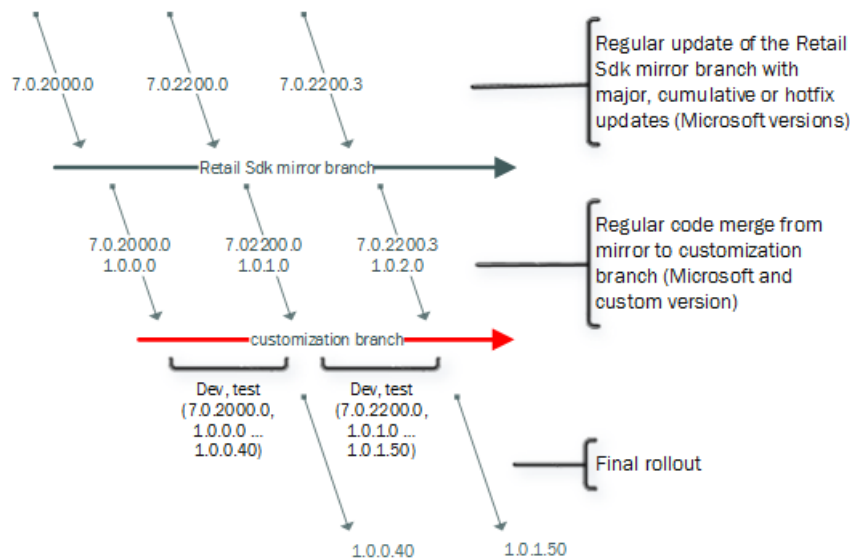
```
47 <!-- This section is for development only features -->
48 <PropertyGroup>
49 <RetailServerDropBinariesOnBuild Condition="'$(RetailServerDropBinariesOnBuild)' == ''>true</RetailServerDropBinariesOnBuild>
50 <RetailServerInstallationLocalBinariesPath Condition="'$(RetailServerInstallationLocalBinariesPath)' == ''>I:\RetailServer\WebRoot\bin</RetailServerInstallationLocalBinariesPath>
51 </PropertyGroup>
52 </Project>
```

## Application Lifecycle Management

A good Application Lifecycle Management (ALM) solution provides version control, builds, automated builds, planning tools, tracking tools, dashboards, customization, and more. The organization of the Retail SDK supports these tasks.

### Branching and versioning

To work efficiently in a team, or even just to be able to go back and look at some changes that were made earlier, you must have a good branching strategy and versioning discipline. The following illustration shows a simple branching strategy that might work well for most teams. The version numbers are fictitious. For more information, see, [Adopt a Git branching strategy](#).



### Retail SDK mirror branch

It's important to emphasize that the non-customized Retail SDK should be stored in your source control. You don't have to store every version, but the versions that your team wants to snap to should be added. (Those versions might be cumulative updates or hotfixes.) Only a simple merge of all changes (that is, additions, changes, and deletions) should be done. No other development work should occur in this branch. The Retail SDK has its own version. All Commerce binaries and packages that are included have the same version. The version can also be found in the root of the SDK folder, in a file that is named Microsoft-version.txt.

### Customization branch

For development, a new customization branch should be created. At the beginning of the initial branch-out, this branch will be an exact copy of the Retail SDK mirror branch. It's the branch that will be used for the team's



development. The version of the customization branch must be incremented at least every time that a build is created for testing. It can even be incremented every day. The file version to increment is defined by using the **CustomVersion** property in the Customization.setting file. If you update the version and rebuild, all binaries, packages, and manifest files are updated accordingly.

The **CustomAssemblyVersion** property should be updated only when the update isn't backward-compatible and/or for major new releases. In other words, you should rarely update this property. In the previous illustration, the current file version of the customization branch is 1.0.2.\* (based on Microsoft version 7.0.2200.3). The file version of the first rolled-out release was 1.0.0.40 (based on Microsoft version 7.0.2000.0). When a testing phase is completed, and the final packages are being deployed with that version, it's important that you either increment the version or create a source control label.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Upcoming changes in the Retail SDK

2/18/2021 • 4 minutes to read • [Edit Online](#)

In a future release of Dynamics 365 Commerce, a new set of features is planned to simplify the development and servicing process for commerce channel customization. To uptake some of these features, you may need to upgrade, recompile, or do some minor code changes in the extensions that you've built. No changes will be required in the extension logic files or manifest that you've created. However, you will have to move your extension files to the new templates, update the packaging model, and recompile Commerce runtime, proxy, and Commerce Scale Unit extensions to map to the new library reference model and integrated development environment (IDE).

Microsoft is publishing this topic before the new features are released, so that you can plan and prepare to uptake them.

Here is the list of new features that are planned for enhancement of the Retail software development kit (SDK), and that will require some work on your side before you uptake them.

## Independent packaging model

Microsoft is developing a new feature that is named the *independent packaging model*. This packaging model helps separate the extensions from the core and services independently. Microsoft plans eventually to support separate packages for extensions too. If there are independent software vendor (ISV) solutions and partner extensions, they can be packaged and serviced separately. ISV solutions and partner extensions no longer have to be combined.

To enable Retail Modern POS to support this new packaging model, Microsoft is changing the point of sale (POS) framework so that it supports the Microsoft Windows optional package extension model. The way that extensions are developed won't change. Only the way that extensions are packaged and deployed will change. In the new model, all Modern POS extensions will be created as separate .appx files. Core POS will load those files as add-ins, and it will run under the core Modern POS app identity. Previously, core POS and extensions were packaged as one .appx file. Now, there will be .appx files for both core and extension add-ins, so that developers can independently service just your extension .appx files. The extensions that are developed will work for both Cloud POS and Modern POS. Although the template might differ, the code is shared between both apps.

### NOTE

When you uptake the new feature, changes are required in the extensions that you previously built. You must move your extension files to the new project type in a separate package and then complete a configuration update. No change is required in the extension logic files or the manifest that you created. However, you must move your extension logic files and manifest to the new template, and then rebuild, test, and validate them.

Independent package model development is not supported in Windows server operating system or LCS development box, you must use your own Windows 10 box.

## Prerequisites

- Visual Studio 2017, version 15.1
- Windows 10, version 1703
- Windows 10, version 1703 SDK

## Development tools

Currently, the Retail SDK samples and other templates inside the Retail SDK work only with Microsoft Visual Studio 2015. To support some of the upcoming changes in the Retail SDK and the new independent packaging model, Microsoft is planning to upgrade from Visual Studio 2015 to either Microsoft Visual Studio 2017 or the latest version.

The new templates and Retail SDK samples can be compiled only in Visual Studio 2017 or the latest supported version. **Therefore, you should plan to upgrade from Visual Studio 2015 to Visual Studio 2017 on your development virtual machines (VMs) when this new feature is released. Starting 10.0.11, Retail SDK components can be developed and compiled only in VS 2017, it will not work in VS 2015.**

## Retail SDK reference folder to NuGet Gallery

All the reference binaries for the Commerce runtime, Commerce Scale Unit, Commerce proxy, commerce tool, and so on, are published in the Retail SDK\Reference folder. Currently, your extensions reference that folder. However, Microsoft is planning to move away from this model and publish the SDK references in the NuGet Gallery instead.

### NOTE

For this new model, you should change your extension project so that it references the NuGet Gallery instead of the Retail SDK\reference folder. This change will require a reference update and recompilation in your extension project.

This new model will simplify the update process. For example, if you require any updated references in the Retail SDK, you must currently go to Microsoft Dynamics Lifecycle Services (LCS), apply the latest binaries updates, and so on. In the NuGet approach, you can just right-click to get the updated version.

## Commerce packaging tool

The channel installers are being enhanced so that you can use them to install just the extensions. You will be able to decide whether to use one combined installer for core and all extensions, or whether to install extensions and core separately.

## Retail SDK samples to GitHub

Microsoft is planning to move the samples from the Retail SDK to GitHub. Because the samples are published only as sample code, this change won't affect your extensions. GitHub makes it easy to get the latest samples. You will no longer have to go through the LCS process.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Development and ALM changes from version 10.0.10 to 10.0.13

2/18/2021 • 7 minutes to read • [Edit Online](#)

It's common for admins of Microsoft Dynamics 365 Commerce environments to pause updates and then leapfrog to the current version. This topic will help ensure that you land safely. There are significant changes and updates to the development and build tools.

Microsoft tries to provide updates that include no breaking changes, and you won't have to refactor your customizations because of these changes. Customizations that follow best practice patterns and that were developed on earlier versions can be deployed on the latest service update.

## Who should read this topic

If you're in a technical role (developer, administrator, or manager) and work with Dynamics 365 Commerce, continue to read this topic. There have been substantial improvements and changes, and there is a wealth of information and documentation that describes what has changed and what other changes are coming. The purpose of this topic is to bring all that information together into a single place, so that you have a better understanding of specific changes that you will have to make to update your development environments or adjust your Application Lifecycle Management (ALM) processes.

## Overall improvements, changes, and notes

The following features are available across products and users. Some features apply to all versions, whereas others apply only to version 10.0.10 and later.

- [All-in-one deployable packages](#) for X++ will merge all your custom code and independent software vendor (ISV) models into a single custom package.
  - In version 10.0.10 and later, custom payment connectors for card-not-present can be included in your deployable package.
  - In version 10.0.13, all-in-one packages will be **mandatory**.
- The commerce deployable package that is created by using the [Retail software development kit \(SDK\)](#) is no longer deployed to Application Object Server (AOS). Instead, you upload your self-service installers to Microsoft Dynamics Lifecycle Services (LCS) and use synchronization. For more information, see [Synchronize self-service installers in Dynamics 365 Commerce](#).
- Use a Microsoft-hosted build agent instead of a dedicated build machine for these purposes:
  - To build [application deployable packages](#)
  - To build [commerce deployable packages](#)

### NOTE

By using a build agent, you will increase your Azure consumption. However, you should see cost savings if you can decommission your dedicated build server.

For a general overview of each application release, always check the "What's new or changed" topics:

- [What's new or changed in Finance and Operations apps home page](#)

- [What's new or changed in Dynamics 365 Commerce](#)
- [What's new and changed in Platform updates](#)

## What's changed in the 10.0.10 release

- Custom payment connectors for card-not-present can be included natively in your X+ + application deployable package. For more information, see [All-in-one deployable packages](#).
- Upload your self-service installers to LCS, and use synchronization. For more information, see [Synchronize self-service installers in Dynamics 365 Commerce](#).

### NOTE

The process for uploading the installer is a manual process. However, Microsoft is working to incorporate the new Commerce asset types into the [DevOps pipeline tools for Dynamics 365 Finance and Operations](#), so that you will be able to automate the upload to LCS.

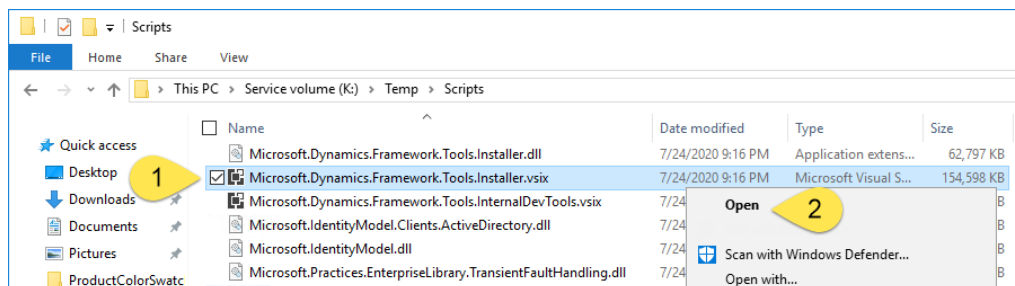
- Custom retail deployable packages no longer have to include the Adyen payment dynamic-link libraries (DLLs).

## What's changed in the 10.0.11 release

- The Retail SDK has been updated to Visual Studio 2017. Visual Studio 2015 is installed on the templates for preconfigured virtual machines (VMs) for developers. However, to build the Retail SDK, you must have Visual Studio 2017 installed. Therefore, you will have to [manually update Visual Studio](#) and rebuild the Retail SDK.

Microsoft is working to rebuild the virtual hard disk (VHD) templates that are used to provision developer and test environments. When this work is completed, the VHD templates will automatically include Visual Studio 2017. Although there is no confirmed date for when the rebuilt templates will be available, Microsoft expects them to be ready soon after version 10.0.13 is generally available in September 2020. For more information, see [Action Required - .NET version and Visual Studio 2017](#).

- Install the Visual Studio tools by installing the Visual Studio extension. To download the extension, follow these steps:
  1. Download the latest service update package from the Shared asset library in LCS. This package is a zip file.
  2. Open the zip file, and find the **DevToolsService\Scripts** folder.
  3. Extract the **Scripts** folder to your local computer (for example, **K:\Temp\Scripts**).
  4. Find the **Microsoft.Dynamics.Framework.Tools.Installer.vsix** file. The size of this file should be about 150 megabytes (MB).
  5. Right-click the file, and then select **Open**.



For more information, see [Update the Visual Studio development tools](#).

- Rebuild the Retail SDK.
  1. Go to the folder that contains your unmodified version 10.0.11 Retail SDK.
  2. Create a backup copy of the Retail SDK for safekeeping.
  3. Open Visual Studio 2017 as an admin, and then open each standard solution file in the SDK folder. You will probably receive a message that states that Visual Studio will close and reopen, and that a computer restart might be required. Select **Continue**.
  4. Open the Visual Studio 2017 Command Prompt window as an admin, and verify that you're using the correct version of the Visual Studio developer command prompt. Visual Studio 2017 uses version 15 of the developer command prompt (whereas Visual Studio 2015 uses version 14). The version number appears in the command window when it opens.
  5. Verify that the standard SDK can be compiled. Go to the root of the **RetailSDK** folder, and run the following command.

```
msbuild /t:rebuild
```

If the build fails, you probably skipped one of the earlier steps.

- References to `PackageReference` (NuGet package reference) are updated. Because of this change, project references are much easier to maintain. Additionally, because of this change, you must manually update *every* custom project in the Retail SDK. Expect to spend about five minutes per project. Look at how the standard projects were updated, and use those updates as a model.
  1. Make sure that everything can be compiled, and that a package can be created from your local developer machine.
  2. Make sure that your build server is updated to Visual Studio 2017.

#### WARNING

Your build server will probably fail, because the names of the NuGet reference folders are very long and will exceed the file path limit of 260 characters.

- Retail SDK file size: The size of an unmodified retail deployable package is now about 340 MB. If customizations are included, the file size might increase to 350 MB. If you try to deploy the file to your Commerce Scale Unit (cloud) (previously known as Retail Cloud Scale Unit [RCSU]), you will receive an error message that states that you can't deploy [packages that are larger than 300 MB](#). Follow these steps to fix this issue:
  1. Follow the instructions in [Deploy the deployable packages](#), and manually remove the self-service installer files.
  2. Upload the much smaller package to LCS, and then continue your deployment as usual.

#### NOTE

The build and deployment scripts that are embedded in the deployable package are hard-coded to check for the self-service installers.

## What's changed in the 10.0.12 release

- It's easier to generate the [Commerce proxy](#).

## What's changed in the 10.0.13 release

- In version 10.0.13, application development that uses X++ requires Visual Studio 2017. If you haven't yet updated to Visual Studio 2017, review the section about version 10.0.11 earlier in this topic.
- Microsoft is working to rebuild the VHD templates that are used to provision developer and test environments. When this work is completed, the VHD templates will automatically include Visual Studio 2017. Although there is no confirmed date for when the rebuilt templates will be available, Microsoft expects them to be ready soon after version 10.0.13 is generally available in September 2020. For more information, see [Action Required - .NET version and Visual Studio 2017](#).

## Where to go for help

1. First, double-check and triple-check to make sure that you followed every step in the relevant topics that are listed earlier in this topic.
2. Check with your partner. Your partner is familiar with your business and setup. Your setup might involve a unique customization that your partner can quickly identify and address.
3. Check the [Retail SDK FAQ](#). It's updated as common issues are identified.
4. Submit a support request through LCS. Provide as much information as you can about the issue. When you submit the request, send a copy to your FastTrack solution architect.
5. Expect to share your screen with your partner, the support specialist, your solution architect, or even a combination of the three. Be prepared to reproduce the error.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Merge the build systems for Commerce and Finance

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes the steps for merging the build systems for Dynamics 365 Commerce and Dynamics 365 Finance. The Lifecycle Services (LCS)-integrated build experience supports both code upgrades and new projects. The Commerce software development kit (SDK) is a self-contained MSBuild-based build system. Many customizers want to make productive changes in both Microsoft Commerce and Finance components. This topic outlines the manual steps for merging both build systems using Azure DevOps.

## Enable the build system

To get started, you must follow all the steps to get a full continuous build system up and running. For information, see [Developer topology deployment with continuous build and test automation](#). After deployment, you create the build definition and build steps. Build at least one time, so that you become familiar with it and are sure that you can build without errors. Then move to the next step.

## Prepare the Retail SDK

### Getting the Retail SDK

If you don't already have the Retail SDK in the same Microsoft Azure DevOps project, add it now. You will find the SDK in any developer or build topology. Follow the branching documentation in [Retail software development kit \(SDK\) architecture](#). We recommend that you create your Retail SDK mirror and your SDK customization branch at this time. After your SDK customization branch is ready, and it has been submitted in the same Azure DevOps project as Commerce, you can start.

## Install NuGet.exe

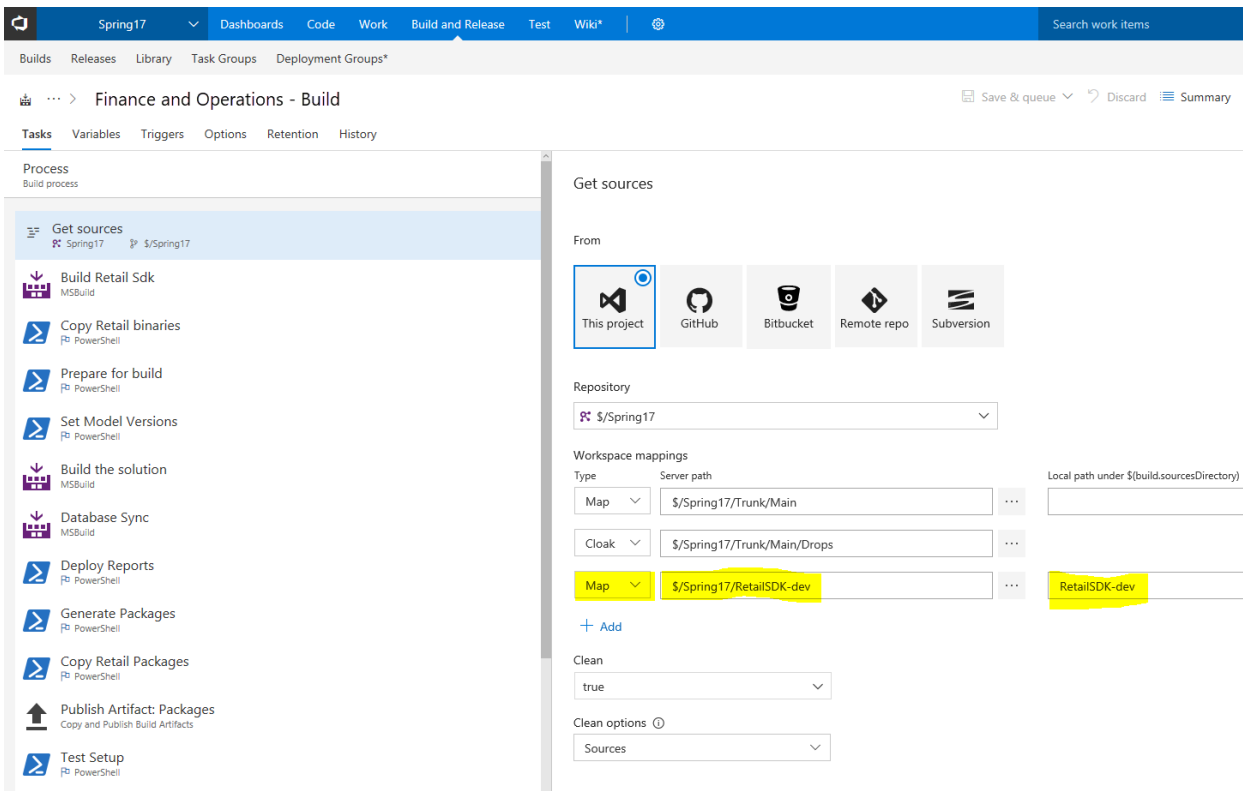
Some of the dependency packages and references have moved to NuGet packages to minimize the file merge and the size of the SDK. These are available for download from the NuGet.org. When you build the Retail SDK these dependencies are automatically pulled from the NuGet.org based on the packages.config file. For this to work, you need to install the [NuGet command line interface](#) and add the nuget to the Windows path after downloading nuget.exe from NuGet.org. The following steps show how to add the nuget to the Windows path:

1. Open the Windows menu and type **Path**. The **Edit the system environment variables** will be available.
2. In that menu, click **Environment variables** on the lower right.
3. In the next window, under **System variables**, select **Path** and click **Edit**.
4. Add an entry for the folder where you would like to store the nuget.exe file or store the nuget.exe file in a folder that is already listed.

## Add a repository mapping for the Retail SDK

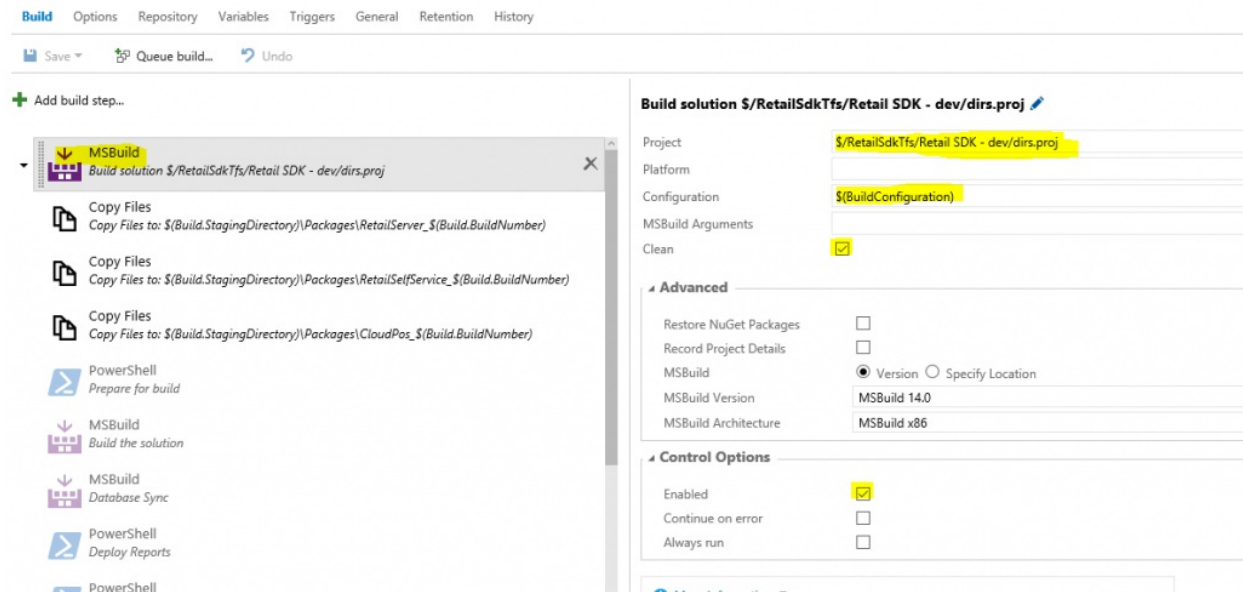
Edit the build definition so that it includes the location of the Retail SDK. (In other words, add a map.)





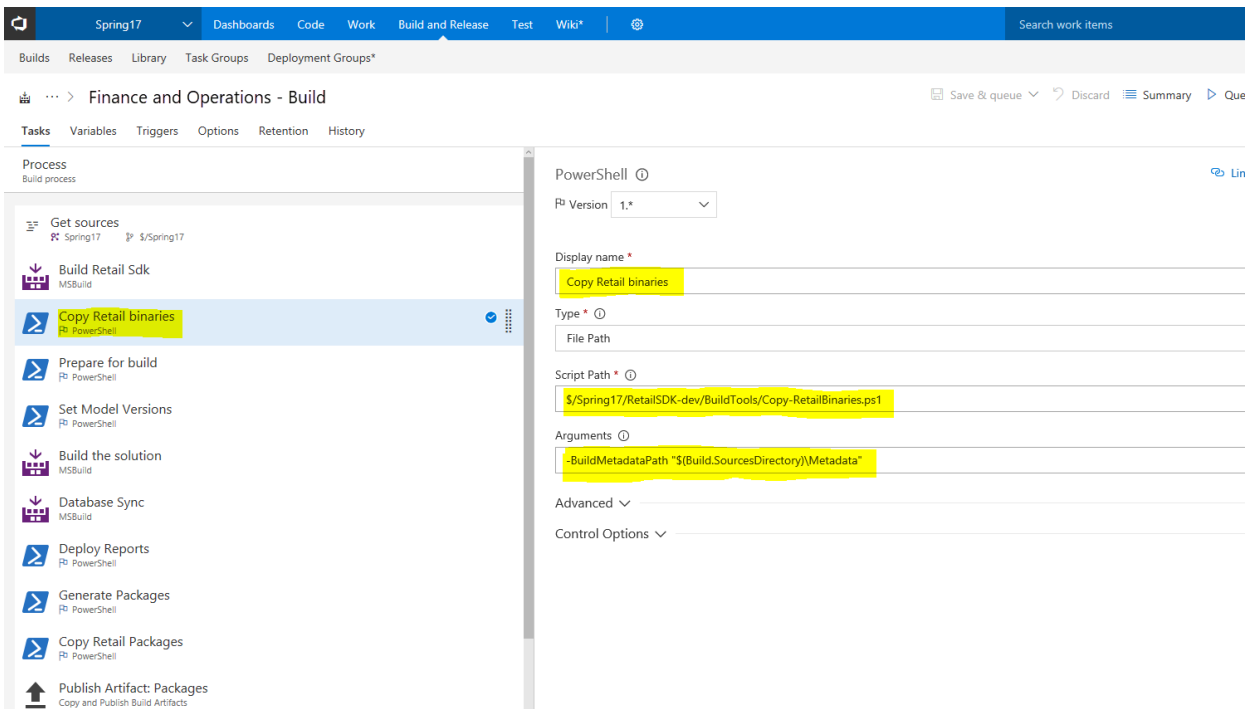
## Add a new build step to build the Retail SDK

Add a new step at the beginning of the build pipeline, as shown in the following screen shot.



## Add a copy step for binaries from the Retail SDK to the Commerce build

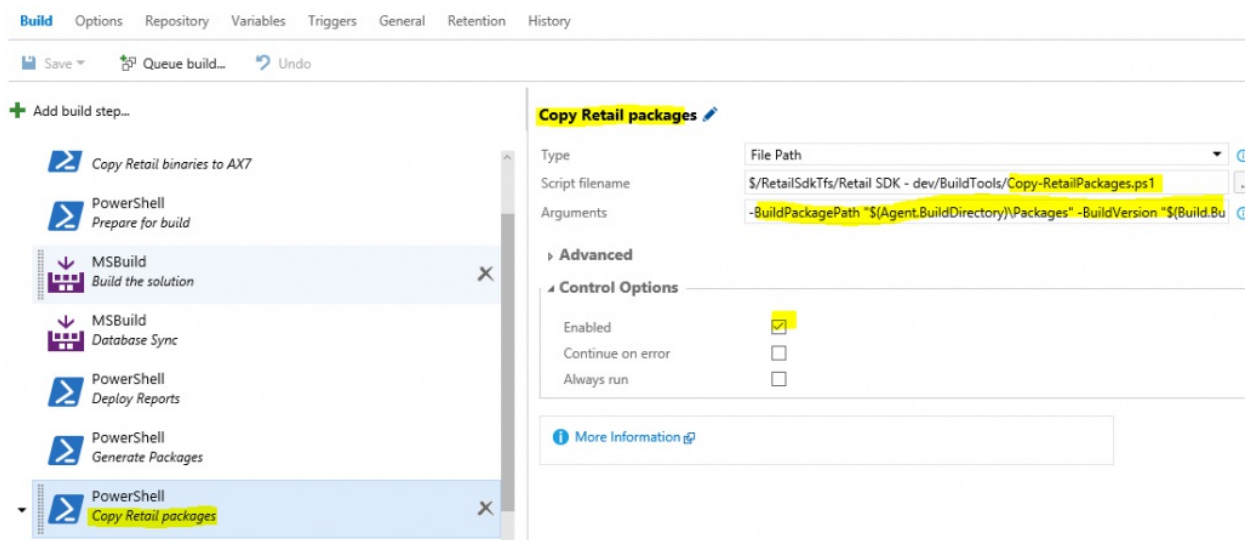
This build step enables Microsoft to copy the latest built Commerce binaries to the Commerce bin folder, if Microsoft shares files/binaries. Make sure that you complete this step immediately after you add a build step for the Retail SDK, as described in the previous section.



## Add a copy step for all Commerce packages

Make sure that this step occurs after the "PowerShell: Generate packages" step (see image below). Here are the arguments.

```
-BuildPackagePath "$(Agent.BuildDirectory)\Packages" -BuildVersion "$(Build.BuildNumber)"
```



## Optional: Referencing a Commerce DLL

You must complete this task only if you must add built Retail binaries to the package. In this case, you must follow these three steps:

1. Use a normal AXReference in your Commerce project.
2. Add the corresponding AXReference folder and the XML file inside it to Azure DevOps.
3. Update the Copy-RetailBinaries.ps1 file with the appropriate file commands to get the binary file from the Retail SDK to the Retail bin folder. The Microsoft Windows PowerShell file includes a sample that copies the PricingEngine.dll file into the ApplicationSuite bin folder. Depending on the modules that you're building, the files and folders must be changed so that they are in a different location.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Migrate the Retail SDK from Visual Studio 2015 to Visual Studio 2017

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic explains what has changed in the 10.0.11 release of the Retail software development kit (SDK), how to migrate it to Visual Studio 2017, and how to update an extension project reference library to NuGet.

## What has changed in the 10.0.11 release

- The Retail SDK has been updated to Visual Studio 2017.
- References have been updated to PackageReference (NuGet package reference).

## Retail SDK updated to support Visual Studio 2017

The Retail SDK now runs on Visual Studio 2017. In release 10.0.11 and later, all Retail SDK components, including Modern POS (MPOS), Cloud POS (CPOS), the Commerce runtime (CRT), Retail Server, the proxy, and Hardware station (HWS), can be built and compiled only in Visual Studio 2017. You can't use Visual Studio 2015.

## References updated to PackageReference

The Retail SDK reference libraries use PackageReference. All the SDK samples use the PackageReference model. All the SDK reference libraries are converted to NuGet packages, and libraries are removed from the RetailSDK\Reference folder. The NuGet packages are in the ..\RetailSDK\Code\pkgs or ..\RetailSDK\pkgs folder. The following example shows the reference to **Microsoft.Dynamics.Commerce.Runtime**.

```
<ItemGroup>
  <PackageReference Include="Microsoft.Dynamics.Commerce.Runtime" Version="9.21.x" />
</ItemGroup>
```

## What is affected

- If you want to deploy a new merged package (that is, an extension and out-of-box changes) on version 10.0.11 or later, you must migrate your solution to the SDK on Visual Studio 2017. No code changes are required to migrate and build your solution.
- Hard-coded references in extension projects must be migrated to PackageReference (NuGet reference).

### Migrate to the SDK for Visual Studio 2017

There are two ways to migrate:

- Deploy a new development and build environment from Microsoft Dynamics Lifecycle Service (LCS), and manually install the Visual Studio 2017. LCS developer VM with Visual Studio 2017 will be available in a future release.
- Update extensions to Visual Studio 2017 in an existing development environment:
  - Install Visual Studio 2017 Community, Professional, or Enterprise edition on the existing build and development virtual machine (VM) with the following workloads:
    - .NET Desktop development
    - Universal Windows Platform development

- ASP.NET and web development
- Azure development
- Node.js development
- .NET Core cross-platform development
- Mobile development with .NET (required for hybrid app development)
- If you manually install Visual Studio 2017, install the following prerequisites on the development VM. If you don't install these prerequisites, compilation will fail, and .NET SDK and runtime errors will be generated:
  - [sdk-2.1.202-windows-x64-installer](#)
  - [sdk-2.1.513-windows-x64-installer](#)
  - [runtime-2.0.9-windows-x64-installer](#)
  - [runtime-2.1.17-windows-x64-installer](#)
  - Install Typescript version 2.2.2. In Visual Studio, go to **Tools > Get Tools and Features**. Select the **Individual components** tab and select the **TypeScript 2.2 SDK from SDKs, libraries, and frameworks** section and install it. VS 2017 has Typescript 3.1 as default, please include 2.2.2 also because the POS app is built based on Typescript 2.2.2.

## Build the Retail SDK

Follow these steps to build the Retail SDK.

1. Open the Developer command prompt for Visual Studio 2017 or the MSBuild 15.0 command prompt. Build the out-of-box Retail SDK by running `msbuild /t:rebuild` from the root of the SDK folder (where you can find the `dirs.proj` file). The folder is `RetailSDK\dirs.proj` or `RetailSDK\Code\dirs.proj` in most installations.
2. Merge your extension to the new SDK folder. For information about how to merge extension with the SDK, see [Upgrade the Retail channel extension to the latest Retail SDK](#).
3. After the extensions have been merged, update all the hard-coded references to `PackageReference` by using the NuGet packages.

## Update the reference in the CRT and Retail Server extension projects

1. a. Use any of the sample CRT projects in the Retail SDK (`..\RetailSDK\SampleExtensions\CommerceRuntime`) as a template and migrate your CRT extension to this new format. The new samples uses the Visual Studio 2017 formats for project dependencies (NuGet references).
2. In the NuGet Package Manager, add the local NuGet repository folder. For information about how to create a local NuGet repository, see [Install and manage packages in Visual Studio using the NuGet Package Manager](#).

### NOTE

All the SDK reference libraries are converted to NuGet packages, and libraries are removed from the `RetailSDK\Reference` folder. The NuGet packages can be found in the `..\RetailSDK\Code\pkgs` or `..\RetailSDK\pkgs` folder.

3. To add the NuGet package reference to the project, right-click the **Dependencies** node in the project, and then select **Manage NuGet Packages**.
4. In the NuGet Package Manager, add the required packages. For example, if the project requires the CRT library reference, add the `Microsoft.Dynamics.Commerce.Runtime` NuGet package. After the NuGet

package reference is added, the project file will be updated with the package reference, as shown in the following example.

```
<ItemGroup>
  <PackageReference Include="Microsoft.Dynamics.Commerce.Runtime" Version="9.21.x" />
</ItemGroup>
```

#### NOTE

PackageReference also supports floating versions, where the version is updated with the floating version number. For more information about floating versions, see [How NuGet resolves package dependencies](#). When the floating version is used, extensions no longer have to update the reference for every update, because NuGet will automatically resolve to the latest version. For example, the package reference might resemble `<PackageReference Include="Microsoft.Dynamics.Commerce.Runtime" Version="9.21.x" />`.

In a similar way, update the references for all the Retail Server, proxy, and Hardware station extension projects.

### Retail Server and Proxy extensions

Migrate the Retail Server and Proxy extensions to the new extension model released in version 10.0.12. Starting in version 10.0.12, the same Retail Server extension library can be used for offline use, no separate C# proxy library is needed, however a client typescript proxy is still required. For more information, see [Create a new Retail Server extension API](#). This step is recommended but not required, as the Retail Server and proxy extension libraries will continue to work until the old extension model is deprecated.

## What isn't affected

You don't have to change the extensions code that was written in previous versions of the Retail SDK. You must update references and recompile only for the new SDK.

If you have existing pipelines in Azure Pipelines not based on build machine agent that are set up for the Retail SDK build will continue to work. In the MSBuild task step, change the MSBuild version to 15.0, if this change is required.

Follow the steps to set up a build pipeline in Azure DevOps without using build VM and build agent from the build machine. For more information, see [Set up Commerce SDK build pipeline](#).

## Azure DevOps pipeline using build machine agent

The same build machine used for MSBuild with the Azure DevOps pipeline can be used with SDK version 10.0.11. Perform the following steps on the build machine for the SDK:

1. Install Visual Studio 2017 on the build machine.
  2. Optionally, run msbuild (msbuild version 15.0) from the developer command prompt for Visual Studio 2017 on the build machine. Open the developer command prompt for Visual Studio 2017 and navigate to the Retail SDK root folder. Type `msbuild dirs.proj` and make sure that the MSBuild completes successfully.
  3. On the build machine, add an environment variable for the MSBuild 15.0. Go to **System Properties > Environment Variables > System variables**. Select the **Path** variable, and then select **Edit**. In the **Edit environment variable** window, select **New** and add the path variable for MSBuild 15.0. Move it to the top of the list of PATH variables. For example, the path will be something like - C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise\MSBuild\15.0\Bin. The path will change based on where you installed Visual Studio 2017. To get the path for MSBuild from the developer command prompt for Visual Studio 2017, type **where MSBuild**.
- To validate the config information

- Open a regular "CMD" window (not the Visual Studio command prompt).
  - Run **msbuild /version**.
  - Verify that the first result in the list is MSBuild version is 15.9.\* or greater.
4. Restart the Azure DevOps build agent on the build machine.
  5. In Azure DevOps pipeline, change the MSBuild version to 15.0 or later.

If the build from Azure DevOps pipeline fails with a NuGet error, the Azure pipeline may not be not using MSBuild version 15.0 for NuGet restore or the extension projects are not upgraded to use the package reference model.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail software development kit (SDK) extensibility samples

2/18/2021 • 3 minutes to read • [Edit Online](#)

The Retail software development kit (SDK) includes extensibility samples. These samples are a good way to learn about different ways to customize Dynamics 365 Commerce.

The Retail SDK includes extensibility samples. You can also use these projects as boilerplate projects to get started with a specific customization. For example, to start a new commerce runtime (CRT) extension, you can just copy one of the CommerceRuntime sample projects into a new folder, adjust the project import paths, and then start working in the project. For most of the samples, additional step-by-step instructions are available in the Retail SDK\Documents folder. The following table provides a high-level list of the samples.

SAMPLE NAME	DESCRIPTION	TASKS THAT ARE DEMONSTRATED
CrossLoyalty	This sample includes two retailers, AdventureWorks and Contoso. The Contoso retailer accepts loyalty points that customers earn from AdventureWorks. The sample shows how to create a simple new CRT service and call it when a button in Retail Modern POS is clicked. It simulates the cross-loyalty scenario. There are changes to the configuration, CRT, Commerce Scale Unit, Commerce Proxy, and point of sale (POS) (both Modern POS and Cloud POS). This sample supports offline mode for Modern POS.	<ul style="list-style-type: none"><li>• Create a new CRT and Commerce Scale Unit extension, or a new service.</li><li>• Create a new operation as a separate project.</li></ul>
EmailPreference	This sample shows how to use extension properties to extend an entity. The extended entity is persisted in both the Commerce and channel databases, and the POS client enables access to the value. The new value is written synchronously to Commerce via the RetailRealtimeTransaction service. No customization is required in the CRT or Commerce Scale Unit, because extension properties flow automatically. There are changes to pages, tables, the Real-time Service (RTS) client, Commerce Data Exchange (CDX), the channel database, and the POS (both Modern POS and Cloud POS). This sample doesn't support offline mode.	<ul style="list-style-type: none"><li>• Modify an existing Modern POS/Cloud POS view or page.</li><li>• Extend the Real-time Service.</li><li>• Use extension properties to store custom field values in the database.</li></ul>



SAMPLE NAME	DESCRIPTION	TASKS THAT ARE DEMONSTRATED
StoreHours	<p>This sample shows how to create a new business entity (StoreHours) across both Commerce and the channel. There are changes to tables, CDX, the channel database, the CRT, Commerce Scale Unit, and the POS (both Modern POS and Cloud POS). This sample supports offline mode for Modern POS.</p>	<ul style="list-style-type: none"> <li>• Create a new CRT and Commerce Scale Unit extension, or a new service.</li> <li>• Modify an existing MPOS/Cloud POS view or page.</li> <li>• Send the custom table data to the channel by using CDX.</li> </ul>
HealthCheck	<p>This sample shows how to expand an existing CRT service by adding functionality. In this case, the health of some other system can be checked as part of the RunHealthCheckServiceRequest service that already exists. There are changes to the CRT and CRT Test Host.</p>	
ExtensionProperties	<p>This sample shows the following customization strategies for the CRT:</p> <ul style="list-style-type: none"> <li>• Extension properties for service, entity, request, and response</li> <li>• Triggers</li> <li>• Notifications</li> <li>• Notification handlers</li> </ul>	<p>Use extension properties to store custom field values in a database.</p>
CommerceRuntime Test Host	<p>This tool mimics a typical CRT host that resembles Commerce Scale Unit. It supports simple testing of CRT extensions that doesn't require changes in Commerce Scale Unit or UI clients. You just register the services that are required for the CRT and run it.</p> <p><b>Note:</b> RealTimeTransaction service calls that might be part of a CRT extension won't work correctly. To test these service calls, use the RetailServer Test Client sample instead.</p>	
RetailServer Test Client	<p>You can use this simple application to make Commerce Scale Unit calls, offline mode calls through the Commerce Proxy, or both. This sample application acts as a client. It resembles the POS clients but requires no UI changes. Therefore, it supports rapid testing and development. You can use this tool to verify customizations of the channel database, RTS, CRT, and/or Commerce Scale Unit before you hand them off to the UI team.</p>	

SAMPLE NAME	DESCRIPTION	TASKS THAT ARE DEMONSTRATED
OnlineStore	This sample is an ASPNET implementation that showcases the use of the Ecommerce SDK. It includes both a store front and a publishing job.	
OpenIdConnectUtility	This sample lets you call Commerce Scale Unit in a customer context (C2) by using a Google Identity.	
CashDispenser (HardwareStation)	This sample shows how to customize Retail Hardware Station so that a cash dispenser can be integrated.	Create a new Hardware Station project and support for new peripherals.
Rambler (HardwareStation)	This sample shows how to customize Hardware Station so that a Rambler Mobile Card Reader can be integrated.	Create a new Hardware Station project and support for new peripherals.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create deployable packages

2/18/2021 • 15 minutes to read • [Edit Online](#)

This topic explains how to create a Commerce deployable package (which is a package that contain all the extensions) for the following components and deploy the package to your environment by using Microsoft Dynamics Lifecycle Services (LCS):

- Commerce runtime (CRT)
- Commerce proxy
- Commerce Scale Unit
- Modern POS
- Cloud POS
- Hardware station
- Channel database scripts
- Payment connector
- Hybrid app (IOS and Android POS app)

## Deployable package types

The following table lists Commerce package types for deployment:

PACKAGE TYPE	DESCRIPTION	APPLIES TO
Retail deployable package	Combined package that contains Commerce scale unit (CRT, RS, DB) extension, CPOS, and self-service packages.	Commerce scale unit environment
Commerce cloud scale unit package	Commerce scale unit (CRT, RS, DB) extension.	Commerce scale unit environment
All in one deployable package	Payment extension for Finance and Operations.	Finance and Operations environment

## Retail deployable package

Retail deployable package is one combined package that contains all your customizations together with all the metadata that is required for deployment. You can use this deployable package to deploy your customizations to various environments. You can do the deployment by using the automated flow in LCS, or you can do it manually by using the scripts that are provided inside the package. This topic guides you through the process of generating the deployable package.

### IMPORTANT

All customizations for the Commerce components are packaged as a single deployable package. Separate packages for individual components, such as Modern POS, Cloud POS, Commerce Scale Unit, CRT are not supported. You must package all extensions as a single deployable package, even if you must merge or combine extensions from independent software vendors (ISVs) or various partners.

If your customizations were built and packaged as individual component packages by using a version of the Retail software development kit (SDK) that is older than application version 7.1.1541.3036, the packages are no longer supported for deployment in LCS. You must uptake the hotfix in [KB 4015062](#), and then rebuild and repackage your customizations.

For detailed information about the Retail SDK, see [Retail software development kit \(SDK\) architecture](#).

## Generate a separate package for Commerce Cloud Scale Unit (CSU)

If you have only CSU extension (CRT, RS, and channel database) then you can generate the separate CSU package instead of generating the full Retail Deployable package, which includes both CSU and self-service packages (MPOS, CPOS, RSSU and HWS).

Retail SDK version 10.0.16 or later supports generating a separate package for Commerce Cloud Scale Unit. This package can be uploaded to LCS > **Asset library** > **Commerce Cloud Scale Unit Extensions** and deployed to CSU.

### Steps to generate a CSU package

#### Option 1

1. Clone or download the Scale unit packaging project from [Dynamics365 Commerce ScaleUnit Samples](#).

Select the correct release branch version according to your SDK/application release. Detailed steps to clone can be in [Download Retail SDK samples and reference packages from GitHub and NuGet](#).

2. Add the extension Commerce runtime, Retail server, and channel database extension project as a Project reference to the scale unit packaging project.
3. Build the scale unit project. This project will generate the **CloudScaleUnitExtensionPackage.zip** output package in the project bin output folder. CloudScaleUnitExtensionPackage.zip package can be uploaded to LCS and deployed to CSU.

Select the correct version of the **Microsoft.Dynamics.Commerce.Sdk.ScaleUnit** NuGet version in the NuGet package manager in Visual Studio according to your SDK/application version.

#### Option 2

1. Create a new C# class library project with Target framework .NET Standard 2.0.
2. Add the **Microsoft.Dynamics.Commerce.Sdk.ScaleUnit** NuGet package as a dependency to the project.

Select the correct version of the Microsoft.Dynamics.Commerce.Sdk.ScaleUnit NuGet version according to your SDK/application version.

Consume the Microsoft.Dynamics.Commerce.Sdk.ScaleUnit package from [https://pkgs.dev.azure.com/commerce-partner/Registry/\\_packaging/dynamics365-commerce/nuget/v3/index.json](https://pkgs.dev.azure.com/commerce-partner/Registry/_packaging/dynamics365-commerce/nuget/v3/index.json). You can add the package source location in the nuget.config file of your extension project file.

```
<packageSources>
  <add key="dynamics365-commerce" value="https://pkgs.dev.azure.com/commerce-
partner/Registry/_packaging/dynamics365-commerce/nuget/v3/index.json" />
  <add key="nuget.org" value="https://api.nuget.org/v3/index.json" />
</packageSources>
```

3. Add the extension Commerce runtime, Retail server, and channel database extension projects as a Project reference to the scale unit packaging project.
4. Build the scale unit project. This project will generate the **CloudScaleUnitExtensionPackage.zip** output package in the project bin output folder. CloudScaleUnitExtensionPackage.zip package can be uploaded to LCS and deployed to CSU.

The Commerce runtime extension config, Web.Config, will be generated by the scale unit packaging project. You do not have to create the extension config files manually.

### Deploy the package to CSU

1. Go to <https://lcs.dynamics.com/v2>.
2. Sign in to LCS and open a project. Then, on the hamburger menu, select Asset library.
3. Select the Commerce **Cloud Scale Unit Extension** asset type, and then select the + button to upload the package. Provide a package name and description and then add the package file by selecting Add file.
4. After the upload is complete, select Confirm to complete the upload process.
5. The package will be validated by LCS in a few minutes. After validation is complete, mark the package as Release candidate.
6. After upload, the package needs to be deployed to the environment. For more information, follow the steps outlined in Apply updates and extensions to Commerce Scale Unit (cloud).

## Finance and Operation payment package

To create a payment package, follow the steps in [Create Commerce payment packaging for Finance and Operations deployment](#).

## Steps to create a combined Retail deployable package

There are two ways to generate a commerce deployable package. You can use the Commerce build automation, or you can generate the package manually by using the build tools in the Retail SDK. This topic focuses on the manual method.

1. Customize or add functionality to the Commerce stack.
2. Use the build tools to identify the customized installation package, code-sign it, and specify the customized CRT, Commerce Scale Unit, and Hardware station assemblies, and customized database scripts.
3. After all the settings have been specified in the **Customization.settings** file in the ...\**Retail SDK\BuildTools** folder, run **msbuild /t:rebuild** on the root of the Retail SDK folder. You can use either the MSBuild build tool or the Microsoft Visual Studio developer command-line tool to generate the deployable packages. Before you build the package, put all the customized assemblies in the ...\**Retail SDK\References** folder. Additionally, put the modified configuration files, such as **CommerceRuntime.Ext.config**, **CommerceRuntime.MPOSOOffline.Ext.config**, **HardwareStation.Extension.config**, and **RetailProxy.MPOSOOffline.ext.config**, in the ...\**Retail SDK\Assets** folder.

## Retail SDK build tools – Customization settings

Most of the configuration values that the Retail SDK uses to build and package customizations are set in the BuildTools\Customization.settings files. These values define metadata that controls how binaries, components, and packages are named, versioned, and code-signed. After you define this metadata, the Retail SDK build

system uses it to identify the customization assets and package them for all the Commerce components.

The following configuration settings are available in the Customization.settings file:

- **AssemblyNamePrefix** – Specify the prefix name for the assembly. When you build the Retail SDK, all the assemblies are prefixed with this name.
- **CustomAssemblyVersion** – Specify the custom assembly version for all assemblies that are built by using the Retail SDK.
- **CustomVersion** – Specify the custom file version for all assemblies that are built by using the Retail SDK.
- **CustomName** – Specify the custom name for the assembly.
- **CustomDescription** – Specify the description for the assembly.
- **CustomPublisher** – Specify the publisher for the assembly.
- **CustomPublisherDisplayName** – Specify the publisher display name.
- **SignAssembly** – Specify **True** to sign the assembly during the build.
- **DelaySign** – Specify **True** to delay signing of the assets during the build.
- **AssemblyOriginatorKeyFile** – Specify the strong name key to use to sign the assembly.
- **ModernPOSPackageCertificateKeyFile** – Specify the Personal Information Exchange (PFX) file to use to sign Modern POS and Hardware station.
- **RetailServerLibraryPathForProxyGeneration** – Specify the customized Commerce Scale Unit assembly to use for proxy generation (both TypeScript and C# proxies).

For Retail SDK version 7.1 and earlier versions, you must specify the name of the Retail Server assembly here.

For Retail SDK version 7.2 and later versions, use the commerce generator tool for proxy generation. However, if you're using the proxy on the e-Commerce client side, specify the assembly name here.

- The **ItemGroup** section includes the following settings:
  - **ISV\_CommerceRuntime\_CustomizableFile** – Specify the details of all the customized CRT and dependent assemblies. You can have multiple entries, one for each assembly.

#### NOTE

If the extension depends on Newtonsoft.Json.Portable or some other assemblies, then explicitly include it. Don't assume that these assemblies will be included by default in the packaging or Commerce Scale Unit folder because the out-of-band (OOB) Commerce Scale Unit or CRT is using this. In the future, if the OOB functionalities don't use these assemblies, it could be removed. As a result, you should always explicitly include all of the extension dependent assemblies in order to package and place them in the correct folder.

#### Example

```
ISV_CommerceRuntime_CustomizableFile Include="$(SdkReferencesPath)\MyCrtExtension.dll"
```

- **ISV\_RetailServer\_CustomizableFile** – Specify the details of all the customized Commerce Scale Unit assemblies. You can have multiple entries, one for each assembly.

#### Example

```
ISV_RetailServer_CustomizableFile Include="$(SdkReferencesPath)\MyRetailServerExtension.dll"
ISV_RetailServer_CustomizableFile Include="$(SdkReferencesPath)\MyRetailServerExtension2.dll"
```

- **ISV\_RetailProxy\_CustomizableFile** – Specify the details of all the customized Commerce proxy assemblies. You can have multiple entries, one for each proxy assembly.

#### Example

```
ISV_RetailProxy_CustomizableFile Include="$(SdkReferencesPath)\MyRetailProxyExtension.dll"
```

- **ISV\_HardwareStation\_CustomizableFile** – Specify the details of all the customized Hardware station assemblies. You can have multiple entries, one for each customized Hardware station assembly.

#### Example

```
ISV_HardwareStation_CustomizableFile Include="$(SdkReferencesPath)\MyHardwareStationExtension.dll"
```

- **ISV\_CustomDatabaseFile\_Upgrade\_Custom** – Specify the details of all the customized database scripts.

#### Example

```
ISV_CustomDatabaseFile_Upgrade_Custom Include="$(SdkRootPath)\Database\Upgrade\Custom\SqlUpdatev1.sql"
```

#### IMPORTANT

Before you start the build process, you must put extension assemblies in ...Retail SDK\References and custom database scripts under ...RetailSDK\Database\Upgrade\Custom.

### Database scripts

Database scripts are packaged together with the Commerce Scale Unit and Modern POS Offline packages, and are run when Commerce Scale Unit and Modern POS are installed. If there are multiple custom database scripts, they are run in alphabetical order. Therefore, to run the scripts in a specific order, you must name them accordingly. The CRT.RETAILUPGRADEHISTORY table tracks the scripts that are already applied to the database. Therefore, the next package upgrade runs only the upgrade scripts that don't have an entry in the CRT.RETAILUPGRADEHISTORY table.

For more details about Channel database extensions, see [Channel database extensions](#).

## Update the extension configuration files

If you have any new extensions in CRT, Commerce Scale Unit, Hardware station, or Retail proxies, you must register the details of the extension assemblies in the <composition> section of the relevant extension configuration file. You can find all the extension configuration files in the ...RetailSDK\Assets folder. Because all extensions are loaded based on the information in the extension configuration files, you must register your assemblies there.

Before you generate the package, you must update the following configuration files if you have any customization in that area:

- **CommerceRuntime.Ext.config** – Register all your CRT extensions and dependent assemblies. Also this is where you need to include the Commerce Scale Unit extension dependent assemblies.

## NOTE

If the extension depends on Newtonsoft.Json.Portable or some other assemblies, then explicitly include it. Don't assume that these assemblies will be included by default in the packaging or Commerce Scale Unit folder because the out-of-band (OOB) Commerce Scale Unit or CRT is using this. In the future, if the OOB functionalities don't use these assemblies, it could be removed. As a result, you should always explicitly include all of the extension dependent assemblies in order to package and place them in the correct folder.

## Example - How to register extension assemblies and extension key value pair configurations

```
<?xml version="1.0" encoding="utf-8"?>
<commerceRuntimeExtensions>
  <composition>
    <!-- Register your own assemblies here. -->
    <add source="assembly" value="my custom library" />
  </composition>
</commerceRuntimeExtensions>
<?xml version="1.0" encoding="utf-8"?>
<commerceRuntimeExtensions>
  <composition>
    <!-- Register your own assemblies here. -->
    <add source="assembly" value="my custom library" />
  </composition>

  <settings>
    <add name="ext.myCustomKey1" value="myCustomValue1" />
    <add name="ext.myCustomarea.myCustomKey2" value="myCustomValue2" />
  </settings>
</commerceRuntimeExtensions>
```

- **CommerceRuntime.MPOSOffline.Ext.config** – Register all your CRT extensions, dependent assemblies and extension Key Value pair configurations. The key name for the extension configuration values must be prefixed with "ext." as the CommerceRuntime initialization will enforce this convention and will not load otherwise, additional prefixes can be added to represent the sub-area they control. Ex: "ext.CusomStorageConfig.CustomKeyCart"

## Example - How to register extension assemblies and extension key value pair configurations

```
<?xml version="1.0" encoding="utf-8"?>
<commerceRuntimeExtensions>
  <composition>
    <!-- Register your own assemblies or types here. -->
    <add source="assembly" value=" my custom library" />
  </composition>
</commerceRuntimeExtensions>
<?xml version="1.0" encoding="utf-8"?>
<commerceRuntimeExtensions>
  <composition>
    <!-- Register your own assemblies or types here. -->
    <add source="assembly" value=" my custom library" />
  </composition>

  <settings>
    <add name="ext.myCustomKey1" value="myCustomValue1" />
    <add name="ext.myCustomarea.myCustomKey2" value="myCustomValue2" />
  </settings>
</commerceRuntimeExtensions>
```

- **HardwareStation.Extension.config** – Register all your Hardware station extensions.



## Example

```
<?xml version="1.0" encoding="utf-8"?>
<hardwareStationExtension>
  <composition>
    <!-- Register your own assemblies or types here. -->
    <add source="assembly" value=" my custom library" />
  </composition>
</hardwareStationExtension>
```

- **RetailProxy.MPOSOOffline.ext.config** – Register all your commerce proxy extensions.

## Example

```
<?xml version="1.0" encoding="utf-8"?>
<retailProxyExtensions>
  <composition>
    <!-- Register your own proxy extension assemblies. -->
    <add source="assembly" value=" my custom library" />
  </composition>
</retailProxyExtensions>
```

## Commerce Scale Unit extension assemblies

Before you start the package, you must add an entry for the Commerce Scale Unit extension assemblies in the <extensionComposition> of the Commerce Scale Unit web.config file, so that the assemblies are loaded and used. You can find the web.config file in the Retail SDK\Packages\RetailServer\Code folder.

The following illustration shows an example of a web.config file.

```
<retailServer>
  <!-- Please specify comma separated versions. -->
  <version supportedVersions="7.0" deprecatedVersions="" />
  <endpoints commerceEndpoint="Commerce" authenticationEndpoint="Auth"/>
  <authentication UserTokenLifetime="24:0:0" Issuer="https://commerce.dynamics.com/auth" AllowInsecureHttp="true" CertTI
  <!-- if maxPageSize is set to 0 request size is uncapped. -->
  <pagingConfiguration maxPageSize="1000" />
  <extensionComposition>
    <!-- Please use fully qualified assembly names for ALL if you need to support loading from the Global Assembly Cache
    if you host in an application with a bin folder, this is not required. -->
    <add source="assembly" value="MyRetailServerExtension" />
  </extensionComposition>
</retailServer>
<system.serviceModel>
  <bindings>
    <ws2007FederationHttpBinding>
      <binding name="SamlBearerTokenBindingConfig"
        closeTimeout="00:01:00"
        openTimeout="00:01:00"
        receiveTimeout="00:10:00"
        sendTimeout="00:10:00"
        maxBufferPoolSize="67108864"
        maxReceivedMessageSize="67108864">
```

## Shared Hardware station web.config

If you are not using the legacy payment connector, then comment the legacy payment connector and enable the non-legacy connector in the web.config file. By default, the legacy payment connector is enabled in the shared hardware station web.config.

## Example

To disable the legacy connector, open the web.config file from \RetailSDK\References\Microsoft.Dynamics.Retail.HardwareStation.WebHost.x.x.x.x\Pkg\bin and comment the legacy connector. Enable the non-legacy connector under the composition section, as shown in the following sample code.

#### NOTE

x.x.x.x in the web.config folder path

(\RetailSDK\References\Microsoft.Dynamics.Retail.HardwareStation.WebHost.x.x.x.x\Pkg\bin) is the version number. This will vary based on your Retail SDK version number.

```
<composition>
  <!-- Defaulting to legacy payment devices.
  <add source="assembly"
  value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.Legacy.PaymentDeviceAdapter"/>
  -->
  <add source="assembly"
  value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.PaymentDeviceAdapter" />
</composition>
```

#### NOTE

Do not add or change any custom settings in the above mentioned example or in any of the channel config files. The only supported modification is to add custom assemblies details in the composition section.

As part of your extension or package, do not edit any of the following config files. These config files will be updated with the latest file from core Microsoft package during deployment and your changes will be lost.

- CommerceRuntime.config
- dllhost.exe.config
- HardwareStation.Dedicated.config
- HardwareStation.Shared.config
- workflowFoundation.config
- Hardware station - Web.config

## Install NuGet.exe

Some of the dependency packages and references have moved to NuGet packages to minimize the file merge and the size of the SDK. These are available for download from the NuGet.org. When you build the Retail SDK these dependencies are automatically pulled from the NuGet.org based on the packages.config file. For this to work, you need to install the [NuGet command line interface](#) and add the nuget to the Windows path after downloading nuget.exe from NuGet.org. The following steps show how to add the nuget to the Windows path:

1. Open the Windows menu and type **Path**. The **Edit the system environment variables** will be available.
2. In that menu, click **Environment variables** on the lower right.
3. In the next window, under **System variables**, select **Path** and click **Edit**.
4. Add an entry for the folder where you would like to store the nuget.exe file or store the nuget.exe file in a folder that is already listed.

## Generate a Retail deployable package

To generate the Retail deployable package, open the MSBuild command prompt or Developer command prompt for Visual Studio 2017 (for SDK version lower than 10.0.11 use MSBuild command prompt or Developer command prompt for Visual Studio 2015). On the developer virtual machine, search for **msbuild** or **Developer command prompt for VS 2017** on the **Start** menu and navigate to the root of the SDK folder. Run the following command. MSBuild will find the dirs.proj file in the SDK root folder and build all the projects included in the dirs.proj (remove the sample projects include in the dirs.proj file). If you want to build your extension project, then update the dirs.proj file with your extension projects.

```
msbuild /p:Configuration=Release
```

## Packages

After the build is completed, deployable packages are generated as a zip file (RetailDeployablePackage.zip) in the Retail SDK\Packages\RetailDeployablePackage folder.

### NOTE

There won't be separate packages for the various Commerce components. All the packages will be combined into one bundle package that is named RetailDeployablePackage.

## Deploy the packages

1. Go to <https://lcs.dynamics.com/v2>.
2. Sign in to LCS, and open a project. Then, on the hamburger menu, select Asset library.
3. Select the **Software deployable package** asset type, and then select the + button to upload the package. Provide a package name and description and then add the package file by selecting **Add file**.
4. After the upload is complete, select **Confirm** to complete the upload process.
5. The package will be validated by LCS in a few minutes. After validation is complete, mark the package as Release candidate.
6. After upload, the package needs to be deployed to the environment. For more information, follow the steps outlined in [Apply updates and extensions to Commerce Scale Unit \(cloud\)](#).

For information about how to deploy the packages either manually or by using the automated flow in LCS, see [Apply a deployable package](#) and [Install a deployable package](#).

LCS has a 300 MB limitation on the package size. If the package size is greater than 300 MB, LCS will not allow deployment of the package. To reduce the size and deploy to RCSU, remove any of the self-service exes (ModernPOSSetup, StoreSystemSetup, or HardwareStationSetup installers). Unzip the package and remove any of the self-service exes and zip the package again. The self-service packages are not deployed to Cloud Commerce scale unit. Sync the self-service package to AOS following the steps in [Synchronize self-service installers in Dynamics 365 Commerce](#).

## Upload self-service installers to LCS and synchronize to Dynamics 365 Commerce

After the Retail deployable package is generated, go to the RetailSDK\Packages\RetailDeployablePackage\content.folder\RetailSelfService\Packages folder and upload all the self-service installers found in that folder to LCS **project > Asset library > Retail Self-service package**. Synchronize the package to Dynamics 365 Commerce by navigating to **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters** and select the **Channel deployment** tab and then select **Check for package updates** to perform synchronization.

For detailed information, see [Synchronize self-service installers in Dynamics 365 Commerce doc](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Development in cloud-hosted environments without admin access

2/18/2021 • 4 minutes to read • [Edit Online](#)

As of Microsoft Dynamics 365 for Finance and Operations, Enterprise edition, Platform update 12, customers will no longer have access to virtual machine (VM) administrator accounts on development or build environments that are running in Microsoft subscriptions. This restriction only applies to new deployments of Platform update 12 (or newer) environments. Environments that were deployed before this update, and have been updated to Platform update 12, will still have administrator access.

You can use a remote desktop (RDP) to access these restricted environments using the non-admin user provided on the Lifecycle Services (LCS) environment page. For more information about environments that don't allow administrator access, see [Development and build VMs that don't allow administrator access FAQ](#).

If you deploy an environment using Microsoft Azure subscription in LCS, then you will not have admin access in this environment. If you need admin access in your environment, use your Azure subscription and deploy the environment using LCS. You can also use the downloadable VHD and deploy it in your Azure virtual machine (VM) or host it locally to get full admin access.

If you don't have admin access in the environment, you will not be able to test and debug using Modern POS. You can still do all commerce customization for POS if you are testing the customization, you must use Cloud POS in that environment. From a customization perspective, there is no difference between Cloud POS and Modern POS - any customization will work both in Cloud POS and Modern POS. There is no additional logic or code for customization completed in Cloud POS in order to work in Modern POS or vice versa, unless you added logic that is browser-specific or UWP app-specific for Hardware and other scenarios. Another option is to do all development work in the environment using Modern POS and test it in some other environment where you have admin access to install MPOS. In most cases, you should be able to test using Cloud POS, expect if you want to test for offline scenarios. If you want to test offline scenarios, you can create a Modern POS installer using the build script, and then test it in your test environment or some other POS registers.

## If you are using Cloud POS for development, set up the following configuration before opening the Cloud POS project

1. Open Visual Studio and go to **View > Application Explorer**. Wait for Internet Information Services (IIS) Express to start with all the Commerce websites deployed. You should see the IIS tray icon in the task bar with all the websites running, such as Cloud POS and Commerce Scale Unit.
2. To debug CRT/RS extensions, attach the CRT/RS project to the IIS Express process.
3. When you open the Cloud POS project from the Retail SDK, IIS Express may fail with the following error.

```
Filename: redirection.config
Error: Cannot read configuration file
```

## To resolve this issue

1. Close Visual Studio.
2. Copy the `aspnet.config` and `redirection.config` files to `%userprofile%\Documents\IISExpress\config`.
3. Open the `applicationhost.config` file in the `%userprofile%\Documents\IISExpress\config` folder.

- In **applicationhost.config**, change the physicalPath of RetailCloudPos to point to your SDK location. For example, physicalPath="K:\RetailSDK\POS\Web". The overall section will look like the following:

```
<site name="RetailCloudPOs" id="4" serverAutoStart="true">
  <application path="/" applicationPool="Dynamics365">
    <virtualDirectory path="/" physicalPath="K:\RetailSDK\POS\Web" />
  </application>
</site>
```

- Save the changes to **applicationhost.config**
- Rename the %userprofile%\Documents\IISExpress\config folder. Do not delete the files because you will copy the **applicationhost.config** file to a new location in **step 8**.
- Start Visual Studio again with the Cloud POS project. The %userprofile%\Documents\IISExpress\config folder will be recreated with the default config files.
- Copy the **applicationhost.config** file from the folder that you renamed in **step 6**, to the folder created in **step 7**.
- Edit ..\RetailSDK\POS\Web\Pos.Web.csproj and change the default **IISURL** node value to your Cloud POS URL and set **UseIIS** to false.

```
Ex:
<UseIIS>False</UseIIS>
<IISUrl>https://YourCPOSURL.com</IISUrl>
```

- Right-click the Pos.Web project and select **Properties**.
- In the **Properties** window, select the **Web** tab. Select the **Start URL** radio option and set the start URL as your Cloud POS URL and under the **Servers** section select **IIS Express as the server** from the drop down menu and set the **Project URL** as your cloud POS URL.

If you are using IIS for development (VMs with admin access), then you need to set the UseIIS node value to False and select Local IIS from the Server drop-down menu.

For example, `https://usnconeboxax1pos.cloud.onebox.dynamics.com`.

- Save the changes.
- Right-click the Pos.Web project and select **Set as StartUp Project**.
- Press F5 to run the Pos.Web Project or click the **Run** button in the Visual Studio menu.

## Install the Developer topology prerequisites

Cloud-hosted development environments do not include Typescript version 2.2.2 by default, and the default Windows host file does not contain the Cloud POS URL to use for local debugging. The **Developer topology prerequisites** package installs Typescript 2.2.2 and updates the Windows host file with your Cloud POS URL. Deploying the package will take a few minutes.

To install the Developer topology prerequisites:

- Go to Lifecycle Services (<https://lcs.dynamics.com>) and sign in.
- Under **Projects**, select the project where your dev environment is deployed.
- In the **More tools** section, click the **Asset library** tile.
- Select **Software deployable package** for the Asset library type and click **Import**.
- In the **Shared asset library list**, select **Developer topology prerequisites** and click **Pick**.
- Go to the **Environments** section, and click your development environment.

7. Click the **Maintain** tab and select **Apply updates**.
8. Select **Developer topology prerequisites** and click **Apply**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up Commerce SDK build pipeline

2/18/2021 • 3 minutes to read • [Edit Online](#)

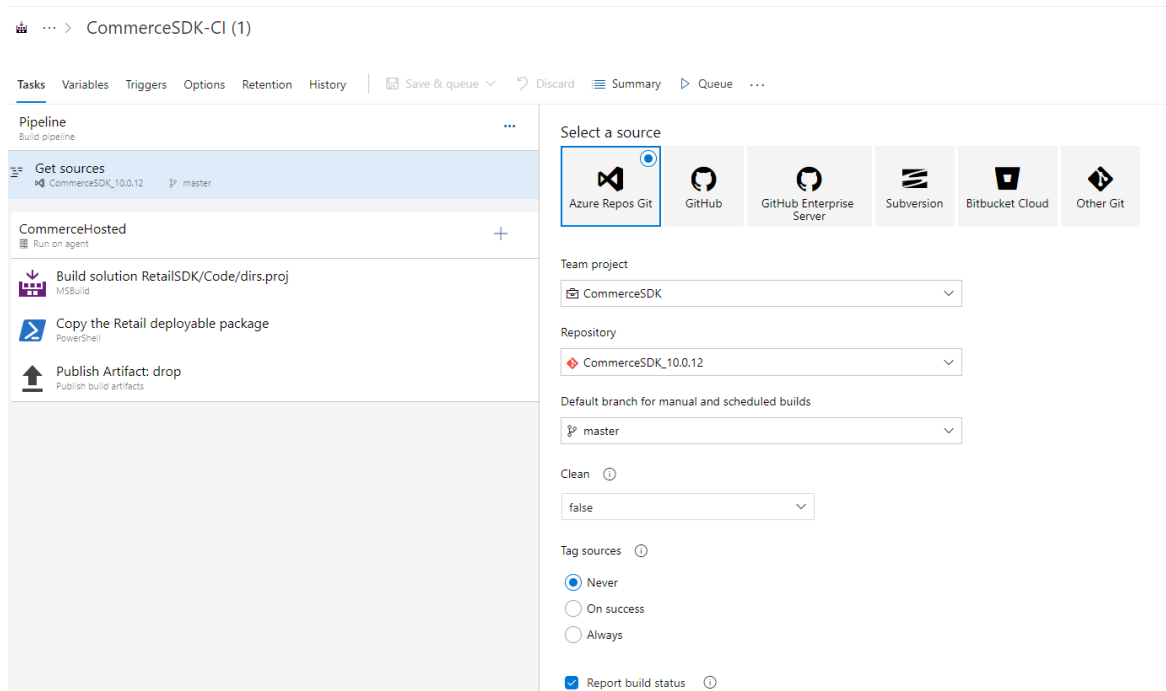
This topic shows how to set up the Azure DevOps build pipeline for the Retail SDK, by using the Azure DevOps build agent (a Microsoft-hosted agent). To generate the build for Retail SDK, a dedicated build machine isn't required. The setup works with the Azure DevOps build agent. This topic applies to Retail SDK version 10.0.11 or greater.

## Prerequisites

The Retail SDK must be added to Azure Repos (Git repository), GitHub, or Azure DevOps. The Retail SDK is available in the service volume drive of the development virtual machine (VM) from Lifecycle Services (LCS). To provision a development VM in LCS, follow the steps in [Deploy and access development environments](#).

## Set up a build pipeline in Azure DevOps

1. Sign in to your Azure DevOps organization.
2. Select **Pipeline** and click **New pipeline**.
3. Select your source repo.
4. Click the pipeline and provide a name for your pipeline. For **Agent pool**, choose **Azure Pipelines**. For **Agent Specification**, enter **vs2017-win2016**.
5. On the **Get sources** tab, select your Retail SDK repo.



6. In the **Agent job** panel, provide a display name. For **Agent pool**, select **<inherit from pipeline>**.

Tasks Variables Triggers Options Retention History | Save & queue Discard Summary Queue ...

Pipeline Build pipeline

- Get sources CommerceSDK\_10.0.12 master
- CommerceHosted Run on agent
- Build solution RetailSDK/Code/dirs.proj MSBuild
- Copy the Retail deployable package PowerShell
- Publish Artifact: drop Publish build artifacts

Agent job CommerceHosted View YAML

Display name \* CommerceHosted

Agent selection <inherit from pipeline>

Agent pool | Manage

Demands

Name	Condition	Value
msbuild	exists	

+ Add

Execution plan

Parallelism

7. Add an MSBuild task. For **Project**, select **RetailSDK\dirs.proj** or **RetailSDK\Code\dirs.proj**. The choice depends on how the Retail SDK is structured. Set **MSBuild Version** to **MSBuild 15.0**. Set **MSBuild Architecture** to **MsBuild x86**.

CommerceSDK-CI (1)

Tasks Variables Triggers Options Retention History | Save & queue Discard Summary Queue ...

Pipeline Build pipeline

- Get sources CommerceSDK\_10.0.12 master
- CommerceHosted Run on agent
- Build solution RetailSDK/Code/dirs.proj MSBuild
- Copy the Retail deployable package PowerShell
- Publish Artifact: drop Publish build artifacts

MSBuild Link settings View YAML

Task version 1.\*

Display name \* Build solution RetailSDK/Code/dirs.proj

Project \* RetailSDK/Code/dirs.proj

MSBuild

Version  Specify Location

MSBuild Version MSBuild 15.0

MSBuild Architecture MSBuild x86

Platform

Configuration

8. Add a PowerShell task (**Run PowerShell script on Linux, macOS, or Windows**). Provide a name for the task. For **Type**, select **Inline**.
9. In the **Script** section, copy and paste this script.

```
# Script to copy the Retail deployable package with Build number.

Copy-Item
"${Build.SourcesDirectory}\RetailSDK\Packages\RetailDeployablePackage\RetailDeployablePackage.zip" -
Destination
"${Build.ArtifactStagingDirectory}\RetailDeployablePackage_${Build.BuildNumber}.zip"
```



CommerceSDK-CI (1)

Tasks Variables Triggers Options Retention History Save & queue Discard Summary Queue

Pipeline Build pipeline

- Get sources CommerceSDK\_10.0.12 master
- CommerceHosted Run on agent
- Build solution RetailSDK/Code/dirs.proj MSBuild
- Copy Retail deployable package PowerShell
- Publish Artifact: drop Publish build artifacts

PowerShell Link settings View YAML

Task version 2.\*

Display name \* Copy Retail deployable package

Type File Path  Inline

Script \* # Script to copy the Retail deployable package with Build number.  
Copy-Item "\$(Build.SourcesDirectory)\RetailSDK\Code\Packages\RetailDeployablePackage\RetailDeployablePackage.zip" -Destination "\$(Build.ArtifactStagingDirectory)\RetailDeployablePackage\_\$(Build.BuildNumber).zip"

ErrorActionPreference Stop

10. Add a Publish Artifact task. Provide the Display name and Artifact name (drop name). Set the Path to publish to \$(Build.ArtifactStagingDirectory).

CommerceSDK-CI (1)

Tasks Variables Triggers Options Retention History Save & queue Discard Summary Queue

Pipeline Build pipeline

- Get sources CommerceSDK\_10.0.12 master
- CommerceHosted Run on agent
- Build solution RetailSDK/Code/dirs.proj MSBuild
- Copy Retail deployable package PowerShell
- Publish Artifact: drop Publish build artifacts

Publish build artifacts Link settings View YAML

Task version 1.\*

Display name \* Publish Artifact: drop

Path to publish \* \$(Build.ArtifactStagingDirectory)

Artifact name \* drop

Artifact publish location \* Azure Pipelines

Advanced Control Options Output Variables

11. Save the changes and queue the build.

12. When the build is complete, you can download the deployable package from the Published Artifacts.

DynamicsCommerceSDK / CommerceSDK / Pipelines / CommerceSDK-CI (1) / 17 / Published artifacts Search

← Artifacts

Published Consumed

Name	Size
drop	305 MB
RetailDeployablePackage_17.zip	305 MB

## Sample YAML script for the pipeline

The following YAML script specifies the pipeline.

```

pool:
  name: Azure Pipelines
  demands: msbuild

steps:
- task: MSBuild@1
  displayName: 'Build solution RetailSDK/Code/dirs.proj'
  inputs:
    solution: RetailSDK/Code/dirs.proj
    msbuildVersion: 15.0

- powershell: |
  # Script to copy the Retail deployable package with Build number.

  Copy-Item
"$$(Build.SourcesDirectory)\RetailSDK\Packages\RetailDeployablePackage\RetailDeployablePackage.zip" -
Destination
"$$(Build.ArtifactStagingDirectory)\RetailDeployablePackage_$$$(Build.BuildNumber).zip"

  displayName: 'Copy the Retail deployable package'

- task: PublishBuildArtifacts@1
  displayName: 'Publish Artifact: drop'

```

## Set up an Azure DevOps build pipeline that uses a build machine

If the build machine uses MSBuild with the Azure DevOps pipeline, follow these steps in the build machine to enable it for the Retail SDK, version 10.0.11:

1. Install Visual Studio 2017 on the build machine.
2. Optional, run **msbuild** (version 15.0) from the developer command prompt for Visual Studio 2017 in the build machine. Open the developer command prompt for Visual Studio 2017, navigate to the Retail SDK root folder, and run `msbuild dirs.proj`. Make sure that the **msbuild** command finishes successfully.
3. In the build machine add an environment variable for **msbuild**, version 15.0. Open **System Properties > Environment Variables > System variables**. Select **Path**, click **New**, and add the path variable for **msbuild**. An example of the path is `C:\Program Files(x86)\Microsoft Visual Studio\2017\Enterprise\MSBuild\15.0\Bin\`. The path depends on where you installed Visual Studio 2017. To get the path for **msbuild**, open the **Developer Command Prompt for Visual Studio 2017**, and run the command `where msbuild`.
4. Restart the Azure DevOps build agent in the build machines.
5. In the Azure DevOps pipeline, set the **Agent Pool** to **Default** and change the msbuild version to 15.0 or latest.

## Troubleshooting

If the build from Azure DevOps fails with a NuGet error, check for these issues:

- The Azure pipeline isn't using **msbuild**, version 15.0, for NuGet restore.
- The extension projects haven't been upgraded to use the package reference model.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Download Retail SDK samples and reference packages from GitHub and NuGet

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains to how to download Retail software development kit (SDK) samples from GitHub and reference packages from NuGet. The Retail SDK includes the code samples, templates, and tools that are required to extend or customize Microsoft Dynamics 365 Commerce functionality. The SDK is published in different repositories (repos) in GitHub, depending on the extension components.

This topic applies to Retail SDK version 10.0.16 or later. For more information about how to download earlier versions of the Retail SDK, see [Retail software development kit \(SDK\)](#).

## Commerce.ScaleUnit repo

The **Commerce.ScaleUnit** repo contains the sample code for customizing the Commerce runtime (CRT), Retail Server, and the channel database.

Clone or download the repo from [Dynamics365 Commerce ScaleUnit Samples](#).

The repo contains multiple branches for each release. The release branch that you should use depends on your go-live version, as shown in the following table. The repo contains only samples, so cloning this repo is optional.

RELEASE BRANCH NAME	VERSION	APPLICATION RELEASE VERSION
<a href="#">Release/9.26</a>	9.26.*	10.0.16
<a href="#">Release/9.27</a>	9.27.*	10.0.17

To clone a single branch, use the following command.

```
git clone --single-branch --branch release/9.26
https://github.com/microsoft/Dynamics365Commerce.ScaleUnit.git
```

This will clone the release/9.26 into your current directory.

### Commerce.ScaleUnit repo folders and projects

FOLDER	PROJECT	CONTENTS	DESCRIPTION
Channel Database	ChannelDatabase.csproj	Contoso.ExampleTable.ChannelDatabase.sql	A sample database extension.

FOLDER	PROJECT	CONTENTS	DESCRIPTION
CommerceRuntime	CommerceRuntime.csproj	<ul style="list-style-type: none"> <li>• <b>Controller</b> – Sample code for implement new Retail Server APIs.</li> <li>• <b>Entities, messages, and request handlers</b> – Sample code for implementing the new CRT service.</li> </ul>	Sample CRT extensions.
ScaleUnit	ScaleUnit.csproj	The project that is required to generate the Commerce Scale Unit (CSU) package	The project that is required to generate the CSU package.

#### NOTE

Repos aren't currently available for in-store components such as Modern point of sale (POS), Cloud POS, Hardware station, Retail scale unit, and other samples. However, Microsoft plans to make them available in later releases.

## Download reference packages for creating APIs, and for consuming messages, request, entities, and contracts

Contracts, messages, entities, and request packages are published in the public NuGet feed. You can use the extension code to consume and customize existing functionalities, or to build new functionalities.

Consume the packages from [https://pkgs.dev.azure.com/commerce-partner/Registry/\\_packaging/dynamics365-commerce/nuget/v3/index.json](https://pkgs.dev.azure.com/commerce-partner/Registry/_packaging/dynamics365-commerce/nuget/v3/index.json). You can add the package source location in the nuget.config file of your extension project file.

```
<packageSources>
  <add key="dynamics365-commerce" value="https://pkgs.dev.azure.com/commerce-partner/Registry/_packaging/dynamics365-commerce/nuget/v3/index.json" />
  <add key="nuget.org" value="https://api.nuget.org/v3/index.json" />
</packageSources>
```

## Reference packages that are available in the public NuGet feed

### Microsoft.Dynamics.Commerce.Sdk.Runtime package

This meta package contains all the required packages for implementing the CRT and Retail Server extensions. All the required contracts, messages, requests/responses, and entities are included in this package.

#### Dependencies:

- Microsoft.Dynamics.Commerce.Diagnostics
- Microsoft.Dynamics.Commerce.Runtime.Data
- Microsoft.Dynamics.Commerce.Runtime.DataServices.Messages
- Microsoft.Dynamics.Commerce.Runtime.Entities
- Microsoft.Dynamics.Commerce.Runtime.Framework
- Microsoft.Dynamics.Commerce.Runtime.Hosting.Contracts
- Microsoft.Dynamics.Commerce.Runtime.Messages

- Microsoft.Dynamics.Commerce.Runtime.RealtimeServices.Messages
- Microsoft.Dynamics.Commerce.Runtime.Services.Messages

### Microsoft.Dynamics.Commerce.Sdk.ScaleUnit package

This package is required to generate the CSU package for deployment.

#### Dependencies:

- CSU packaging dependencies

### Microsoft.Dynamics.Commerce.Sdk.ChannelDatabase package

This package is required to generate the database packages with CSU.

#### Dependencies:

- Channel database packaging dependencies

### Package versioning

PACKAGE VERSION	APPLICATION RELEASE
9.26.x_Preview	10.016 PEAP release
9.26.x	10.0.16 Customer preview
9.26.x	10.016 GA

An extension project can consume the correct version if you add the package reference to the project and include the full version number. Alternatively, to make the extension project always get the latest version, use a wildcard character. We recommend that you use the full version number, and that you update the version, based on your go-live version. There are two options:

- Without a wildcard character:

```
<PackageReference Include="Microsoft.Dynamics.Commerce.Sdk.Runtime" Version="9.26" />;
```

- With a wildcard character:

```
<PackageReference Include="Microsoft.Dynamics.Commerce.Sdk.Runtime" Version="9.26.*" />;
```

For every hotfix and new application release, a new version of the package will be published in the same public feed. Consume the package version that is based on the version that is required for your go-live. If the version of the package that is consumed is higher than the version of your go-live application, runtime and deployment failures will occur.

## Best practices and branching strategies

For detailed information about the Git branching strategy, see [Git branching strategy](#).

Keep your branch strategy simple, and follow these best practices:

- Use feature branches for all new features and software updates.
- Merge feature branches into the main branch by using pull requests.
- Keep a high-quality, up-to-date main branch.

The following branching strategies are based on the way that Microsoft uses Git. For more information, see [How](#)

we use Git at Microsoft.

## Create a new feature branch for development and software updates

Create a new feature branch that is based on the Dynamics 365 Commerce release/x.x.x branch. Clone the release/x.x.x branch, and then create a new branch. Be sure to use the correct naming convention for the new branch. For more information, see [Git branching doc for sample naming convention](#).

### Clone the release/x.x.x branch and create a new branch

1. Create the clone.

```
git clone --single-branch --branch release/9.26
https://github.com/microsoft/Dynamics365Commerce.ScaleUnit.git

git checkout -b private/{username}/{feature/description}
```

2. Add and commit new changes to the development branch.

```
git -add .
git commit -m"commit message"
```

3. After the development is completed, tested, and validated, push the changes to the main branch by using **git push <remote> <branch>**.

```
git push origin {private branch name}
```

## Create a release branch after development

1. After you push the development changes to the main branch, create a new release branch, and then create the deployable packages from it.

```
git checkout -b release/x.x.x
```

2. If any changes are made in the release branch, merge them from the release branch back to main branch.

```
git checkout master git merge release/x.x.x
```

## Create an extension hotfix branch

As you did for the release branch, create a hotfix branch for extension from the main branch, and then release the hotfix. Later, merge the changes back to the main branch.

## Merge the new SDK release branch to the main and development branches

After a new version of the SDK samples is released, you must merge it with your new branch. Because the SDK contains only samples, you don't have to get the updated changes from the new SDK release branch.

```
git checkout master git merge release/x.x.x
```

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail SDK FAQ

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic summarizes answers to questions that are frequently asked by users of the Retail SDK.

## How do I handle a runtime error thrown when using `ISupportedTypesAware` in SDK version (10.0.420.10003~10.0.420.10033)?

This error might occur when you use `ISupportedTypesAware`.

*Exception: System.MissingMethodException: Method not found:  
'System.Collections.Generic.IEnumerable`1<System.Type>  
Microsoft.Dynamics.Commerce.Runtime.ISupportedTypesAware.get\_SupportedRequestTypes()'*

When extending or implementing CRT handlers, use `IRequestHandlerAsync` instead of `ISupportedTypesAware`. `ISupportedTypesAware` is not supported and if used in extension it may cause this runtime error. Additionally, don't use explicit implementations.

The following example shows the recommended approach using `IRequestHandlerAsync`.

```
public class MyHandler : IRequestHandlerAsync
{
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            return new[]
            {
                typeof(MyRequest)
            };
        }
    }

    public Task<Response> Execute(Request request)
    {
        // Return response.
    }
}
```

The following example, which is not recommended, uses `ISupportedTypesAware`.

```
// NOT RECOMMENDED
public class MyHandler : IRequestHandlerAsync, ISupportedTypesAware
{
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            return new[]
            {
                typeof(MyRequest)
            };
        }
    }

    Task<Response> IRequestHandlerAsync.Execute(Request request)
    {
        return null;
    }
}
}
```

## How do I handle a ReadOnly CartValidationException error in SDK version (10.0.0 and later)?

This error will occur if the following read-only properties are updated in SDK version 10.0.9 or earlier. These properties are made read only in SDK version 10.0.0 to avoid miscalculations in the cart total:

- ExtendedPrice
- TaxAmount
- ItemTaxGroupID
- TaxAmountExclusive
- TaxAmountInclusive
- TotalAmount
- NetAmountWithoutTax
- IsInvoiceLine
- InvoiceAmount

To resolve this issue, remove the code in the client/server. Setting this value and OOB code will calculate these values or keep the logic in the read-only fields. You can follow the example shown below, however be sure to modify the code according to the best practice, as this option is not recommended.

In HQ Commerce **parameters > Configuration parameters > Create a new config** with a name/value like the following pattern:

**RetailReadOnlyExempt\_[EntityName] : [ColumnName]**

For example, in the error at the top of this page, ITEM TAXGROUPID is the read-only property with the error. You can search to see this is part of CartLine and CartLineData. The configuration name/value pair would look like this:

**RetailReadOnlyExempt\_CartLineData : ITEM TAXGROUPID**

Additionally, to exempt more than one column for a given entity, use a comma as the separator for column names. The configuration name/value pair in this case would look like this:

**RetailReadOnlyExempt\_CartLineData : TAXAMOUNT,TOTALAMOUNT, ExtendedPrice, ItemTaxGroupID, TaxAmountExclusive, TaxAmountInclusive, TotalAmount, NetAmountWithoutTax, IsInvoiceLine, InvoiceAmount**



#### NOTE

ITEMTAXGROUPID, TAXAMOUNT, TOTALAMOUNT are the column names of the properties in the examples above, not the actual property names. The entity name for the CartLine is CartLineData.

After adding the config run job 1110 to push this change.

## - Deployment failed with error - Could not load file or assembly Microsoft.Dynamics.Commerce.RetailServer.Extensibility.Contracts

Deployment of the new retail deployable package fails with the message:

*An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code. Could not load file or assembly "Microsoft.Dynamics.Commerce.RetailServer.Extensibility.Contracts, Version=7.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" or one of its dependencies. The system cannot find the file specified.*

This error may occur if the extension project upgraded from Retail SDK version 10.0.10 or 10.0.11 to Retail SDK version 10.0.12 or later, and there are extension projects referencing Microsoft.Dynamics.Commerce.RetailServer.Extensibility.Contracts.

To fix this issue, remove the reference to Microsoft.Dynamics.Commerce.RetailServer.Extensibility.Contracts in the extension project and regenerate the package and deploy again. Removing this reference library will not impact extension code/functionality because it's not needed for extension solutions. Starting in version 10.0.11, extensions can refer to Microsoft.Dynamics.Commerce.Hosting.Contracts for Retail server APIs.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

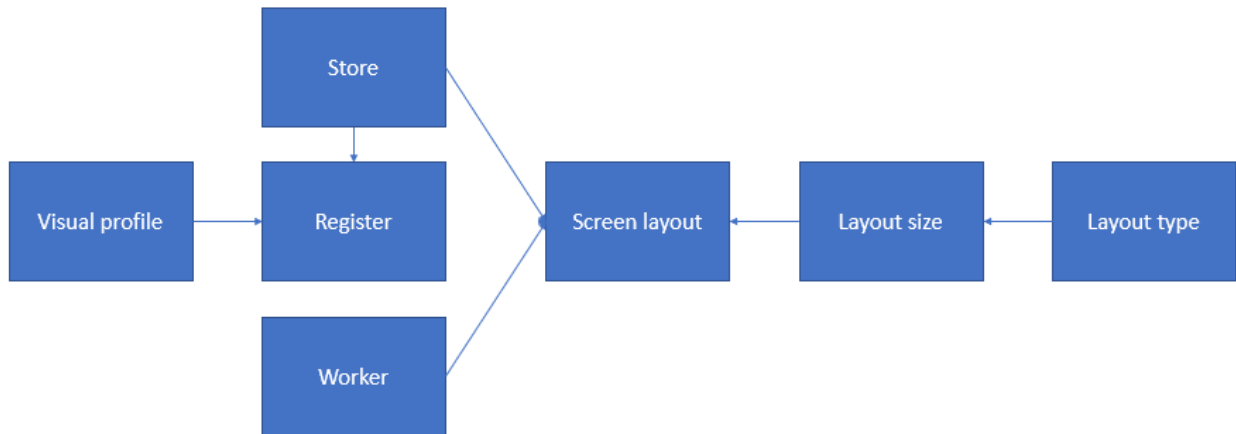
The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# POS user interface visual configurations

2/18/2021 • 12 minutes to read • [Edit Online](#)

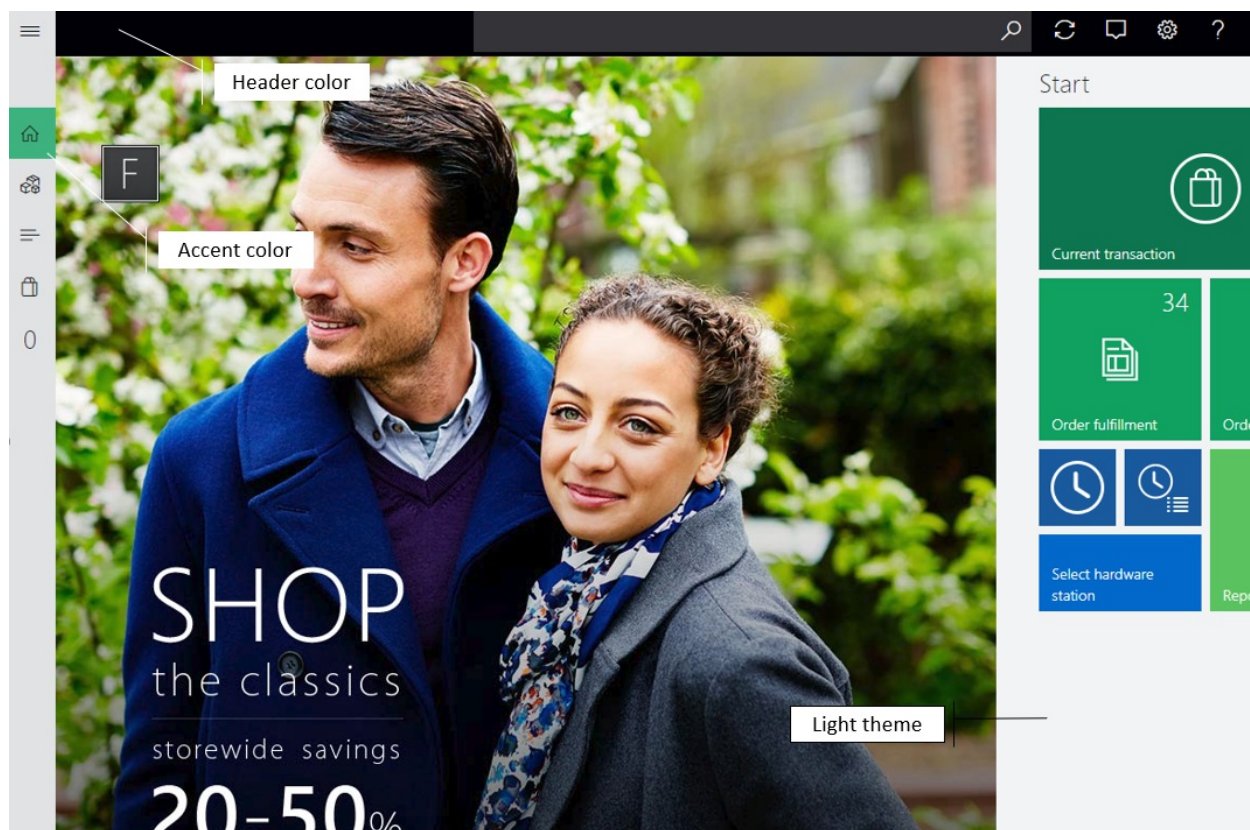
The user interface (UI) of the Microsoft Dynamics 365 Commerce point of sale (POS) can be configured by using a combination of visual profiles and screen layouts that are assigned to stores, registers, and users. This topic provides information about those configuration options.

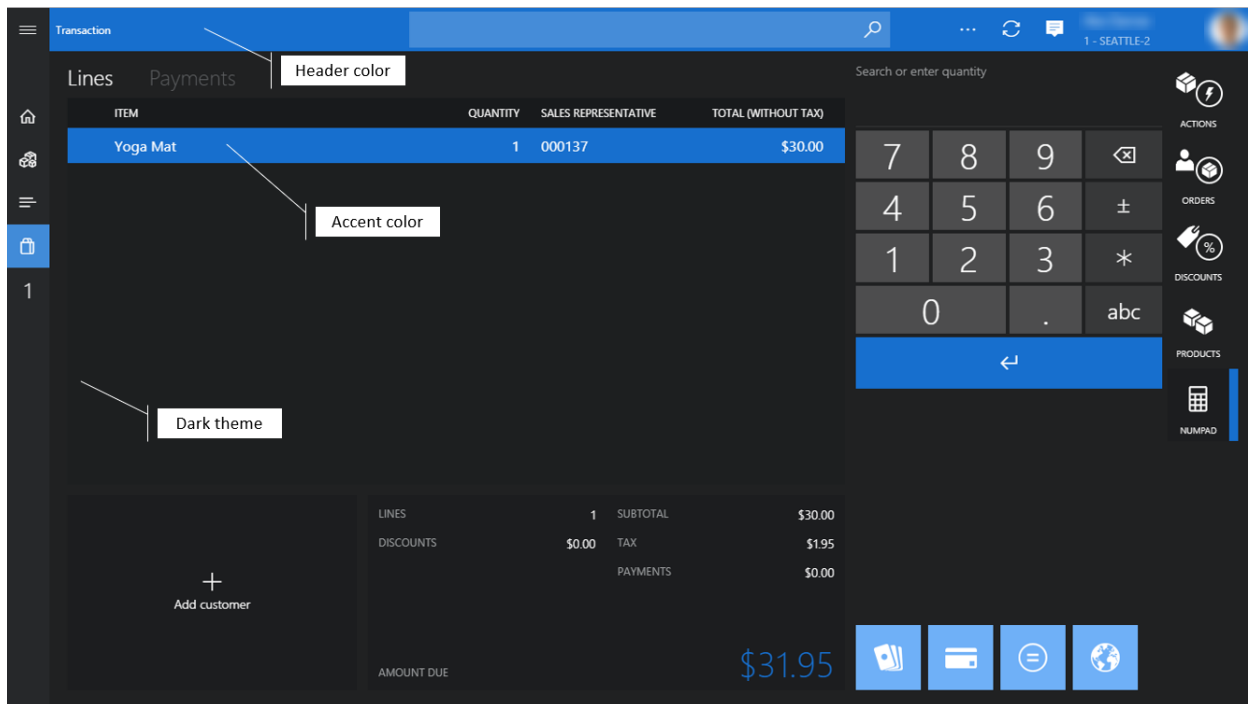
The following illustration shows the relationships among the various entities that make up the configurable aspects of the POS UI.



## Visual profile

Visual profiles are assigned to registers, and they specify the visual elements that are register-specific and shared across users. Every user who signs in to the register sees the same theme, layout, colors, and images.





- **Profile number** – The profile number is the unique identifier of the visual profile.
- **Description** – You can specify a meaningful name that will help identify the correct profile for your situation.
- **Theme** – You can select between the **Light** and **Dark** application themes. The theme affects the font and background colors throughout the application.
- **Accent color** – The accent color is used throughout the POS to differentiate or highlight specific visual elements, such as tiles, command buttons, and hyperlinks. Typically, these elements are actionable.
- **Header color** – You can configure the color of the page header to meet the retailer's branding requirements.
- **Font scheme** – You can select between the **Standard** and **Large** font schemes. The font scheme affects the font size throughout the application. The default selection is **Standard**.
- **Always show application bar labels** – When this option is turned on, the label text is always visible under the application bar buttons.
- **Layout** – You can select between the **Centered** and **Right** layouts. The layout affects the alignment of the sign-in box on the sign-in screen. The default selection is **Centered**.
- **Show date/time** – When this option is turned on, the current date and time are shown in the POS header and on the sign-in screen.
- **Keyboard** – You can select between **Default to OS keyboard** and **Show number pad** to specify the default keyboard that is used for input on the sign-in screen. The number pad is a virtual keyboard that is used primarily for touch-based devices. The default selection is **Default to OS keyboard**.
- **Logo image** – You can specify a logo image that is shown on the sign-in screen. We recommend that you use an image that has a transparent background. The file size should be kept as small as possible, because application behavior and performance can be affected when large files are stored and loaded.
- **Login background** – You can specify a background image for the sign-in screen. The file size of background images should be kept as small as possible.
- **Background** – You can specify a background image that is used instead of the solid theme color throughout the application. As for background images for the sign-in screen, the file size should be kept as small as possible.

#### NOTE

The **Right** layout and date/time display don't apply to the sign-in screen in compact view.

You need to run the **1090 (Registers)** distribution schedule job to synchronize the latest visual profile

configurations to the channel database.

## Screen layouts

Screen layout configurations determine the actions, content, and placement of UI controls on the POS **Welcome** screen and **Transaction** screen.

The screenshot shows the Dynamics 365 Retail POS setup interface for configuring screen layouts. The left sidebar lists various roles and their associated screen layouts, such as F3.1CSH (Fabrikam Cashier) and F3.MGR (Fabrikam Manager). The main area displays the 'Screen layouts' configuration page for the 'F3.MGR' role, showing a table of layout sizes and button grids.

Screen layout ID	Name	Default start screen
F3MGR	Fabrikam Manager	Welcome screen

Name	Layout type	Width	Height
1024x768 - Full	Modern POS - Full	1024	768
1280x720 - Full	Modern POS - Full	1280	720
1366x768 - Full	Modern POS - Full	1366	768
1440x960 - Full	Modern POS - Full	1440	960
480x853 - Compact	Modern POS - Compact	480	853

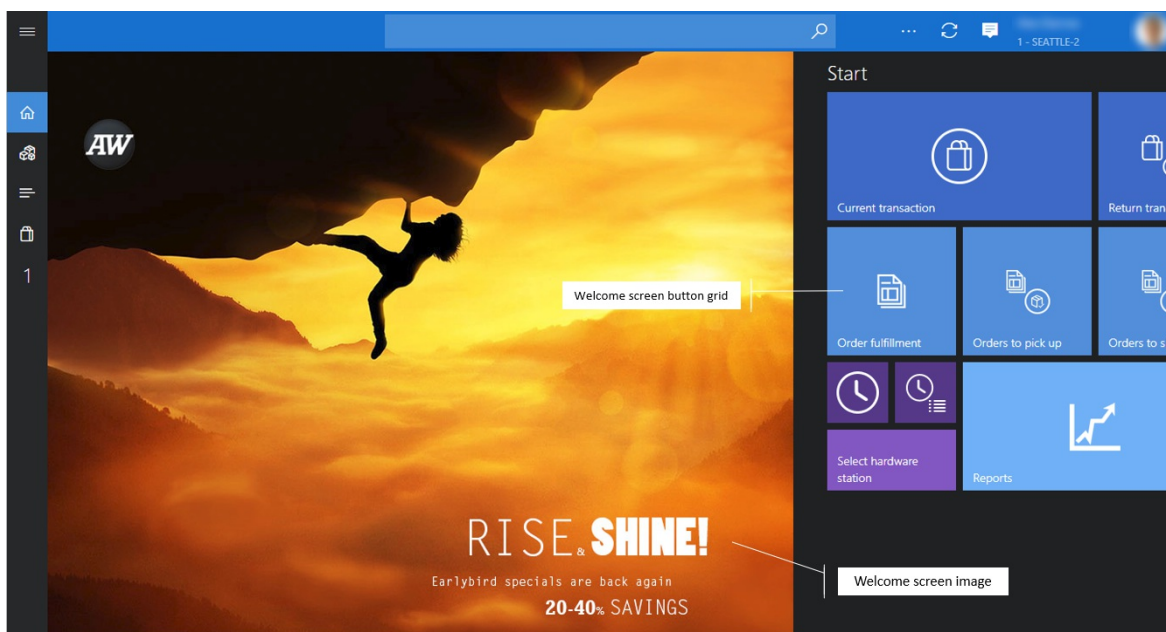
Layout zone	Button grid ID	Name
Welcome screen 1	F2W1M	Start
Welcome screen 2	F2W2	Products
Welcome screen 3	F2W3	Inventory
Welcome screen 4	F2W4M	Shift and drawer
Welcome screen 5	F2W5M	Operations
Transaction screen 1	F2T1M	Actions
Transaction screen 2	F2T2	Customer orders

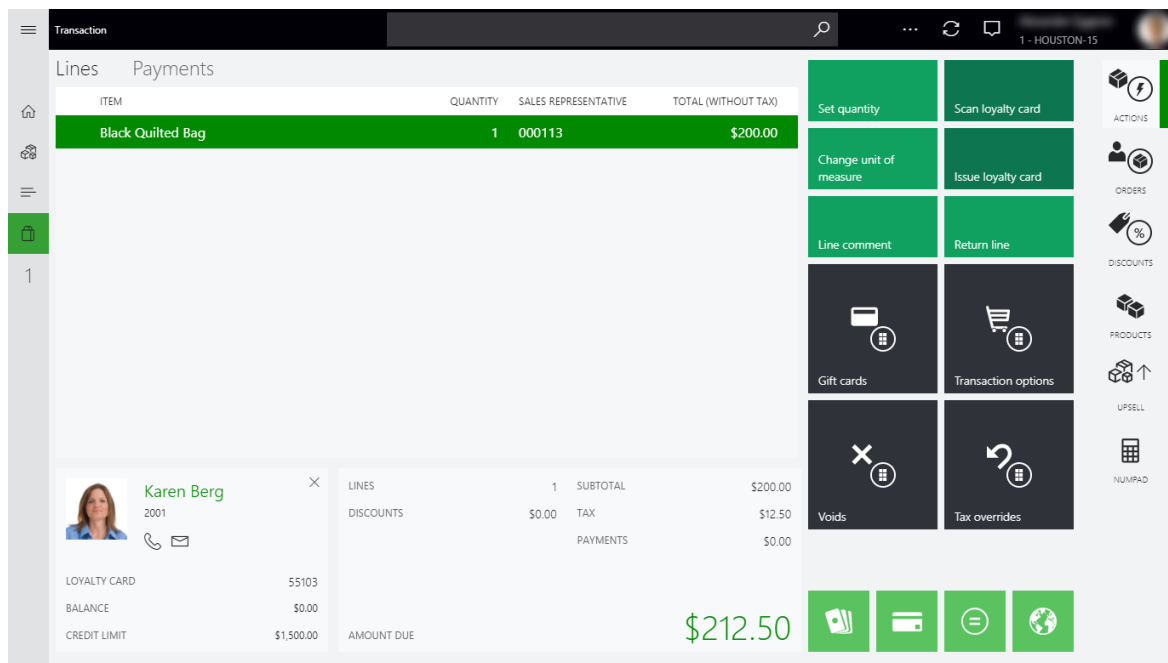
Layout zone	Image
Image 1	2506
Image 2	2507
Image 3	2508

Annotations in the image point to 'Screen layout' (the sidebar), 'Layout sizes' (the table), 'Button grids' (the table), and 'Images' (the table).

- **Welcome screen** – In most cases, the welcome screen is the page that users see when they first sign in to the POS. The welcome screen can consist of a branding image and button grids that provide access to POS operations. Typically, operations that aren't specific to the current transaction are put on this screen.



- **Transaction screen** – The **Transaction** screen is the main screen in the POS for processing sales transactions and orders. The content and layout are configured by using the screen layout designer.



- **Default start screen** – Some retailers prefer that cashiers go directly to the **Transaction** screen after sign-in. The **Default start screen** setting lets you specify the default screen that appears after sign-in for each screen layout.

### Assignment

Screen layouts can be assigned at the store, register, or user level. The user assignment overrides the register and store assignments, and the register assignment overrides the store assignment. In a simple scenario where all users use the same layout, regardless of register or role, the screen layout can be set only at the store level. In scenarios where specific registers or users require specialized layouts, those layouts can be assigned.

Depending on which level the screen layouts are assigned, you need to run the **1070 (Channel configuration)**, **1090 (Registers)**, and/or **1060 (Staff)** distribution schedule jobs to synchronize the latest screen layout configurations to the channel database.

### Layout sizes

Most aspects of the POS UI are responsive, and the layout is automatically resized and adjusted based on the screen size and orientation. However, the POS **Transaction** screen must be configured for every screen resolution that is expected.

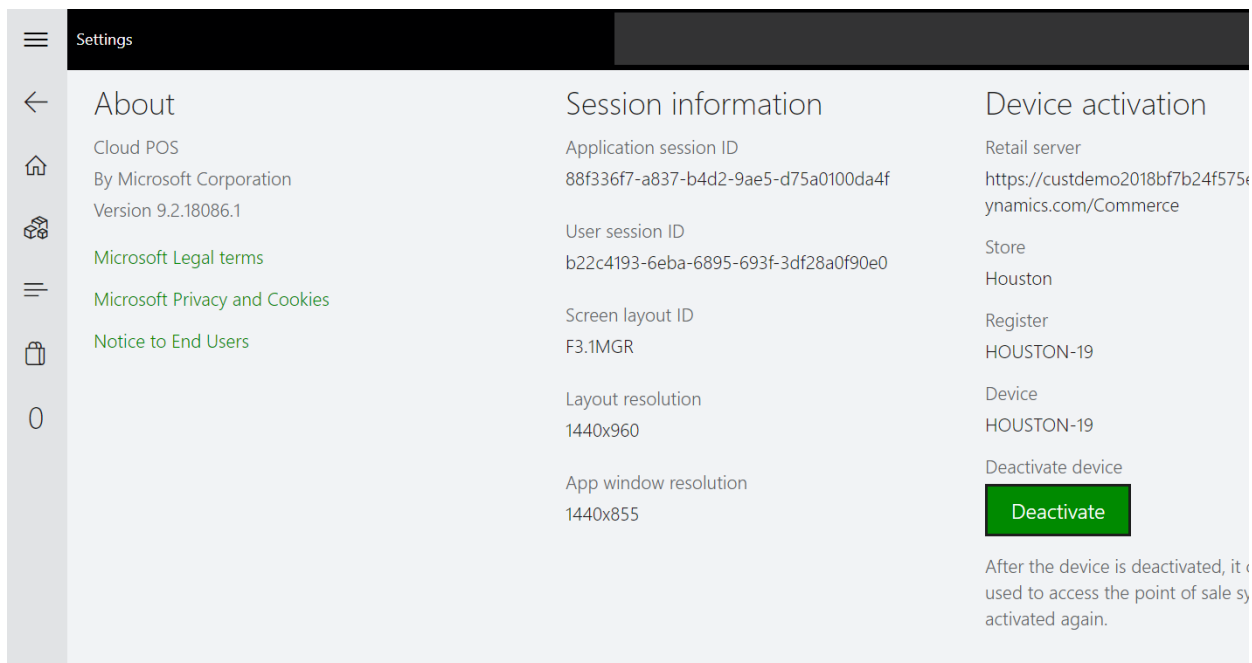
At startup, the POS application automatically selects the closest layout size that is configured for the device. A screen layout can also contain configurations for both landscape and portrait modes, and for both full-size and compact devices. Therefore, users can be assigned to a single screen layout that works across various sizes and form factors that are used in the store.



- **Name** – You can enter a meaningful name to identify the screen size.
- **Layout type** – The POS application can show its UI in various modes to provide the best user experience on a given device.
  - **Modern POS – Full** – Full layouts are typically best for larger displays, such as desktop monitors and tablets. You can select the UI elements to include, specify the size and placement of those elements, and configure their detailed properties. Full layouts support both portrait and landscape configurations.
  - **Modern POS – Compact** – Compact layouts are typically best for phones and small tablets. The design possibilities for compact devices are limited. You can configure the columns and fields for the receipt and totals panels.
- **Width/Height** – These values represent the effective screen size, in pixels, that is expected for the layout. Remember that some operating systems use scaling for high-resolution displays.

**TIP**

You can learn the layout size that is required for a POS screen by viewing the resolution in the app. Start the POS, and go to **Settings > Session information**. POS shows the screen layout that is currently loaded, the layout size, and the resolution of the app window.

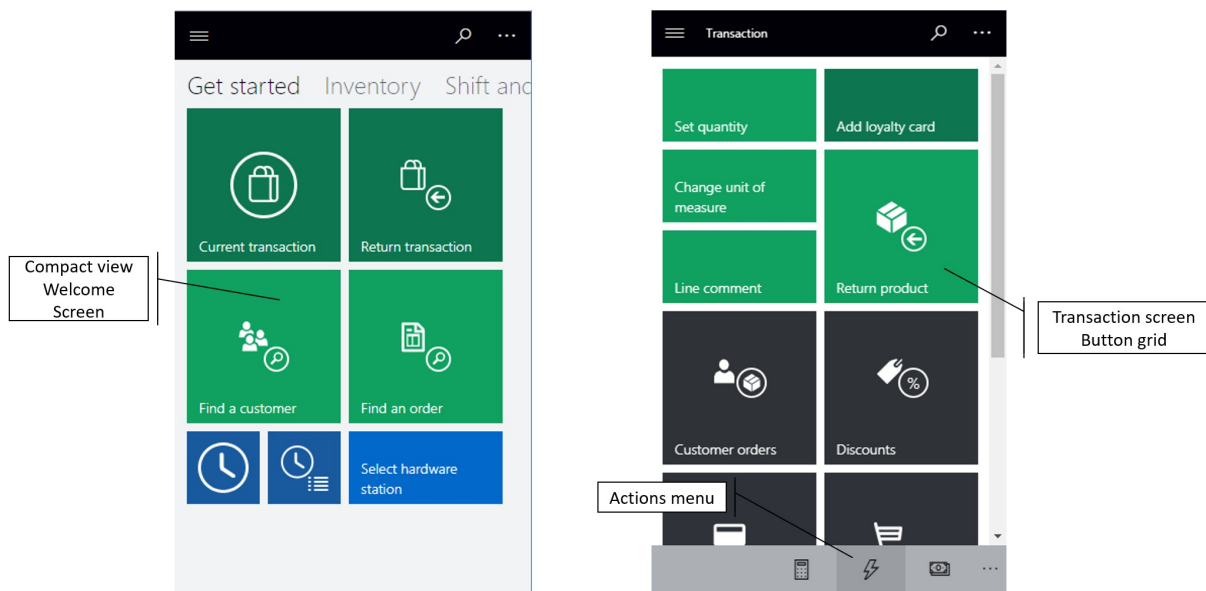


## Button grids

For each layout size in a screen layout, you can configure and assign button grids for the POS welcome screen and **Transaction** screen. Button grids for the welcome screen are automatically laid out from left to right, from the lowest number (Welcome screen 1) to the highest number.

In Full POS layouts, the placement of button grids is specified in the screen layout designer.

In Compact POS layouts, the button grids are automatically laid out from top to bottom, from the lowest number (Transaction screen 1) to the highest number. They can be accessed on the **Actions** menu.



### NOTE

The button sizes in the designer will scale to fit the size of the window, therefore they may not accurately reflect the actual buttons rendered in POS. To best simulate the button grid layout, adjust the designer windows to the same size as the POS.

## Images

For each layout size in a screen layout, you can specify images to include in the POS UI. For Full POS layouts, a

single image can be specified for the welcome screen. This image appears as the first UI element on the left. On the **Transaction** screen, images can be used as tab images or as a logo. Compact POS layouts don't use these images.

## Screen layout designer

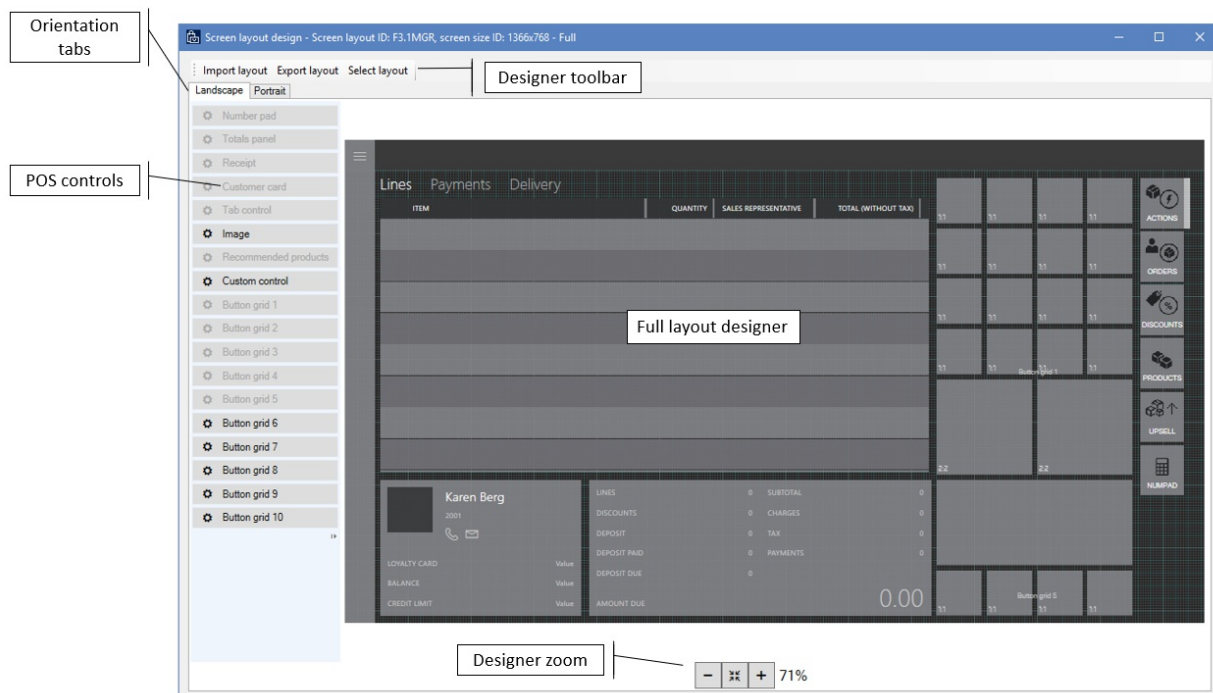
The screen layout designer lets you configure various aspects of the POS **Transaction** screen for each layout size, in both portrait and landscape modes, and for both Full and Compact layouts. The screen layout designer uses the ClickOnce deployment technology to download, install, and start the latest version of the application every time that users access it. Be sure to check the browser requirements for ClickOnce. Some browsers, such as Google Chrome, require extensions.

### IMPORTANT

You must configure a screen layout for each layout size that is defined and that is used by the POS.

## Full layout designer

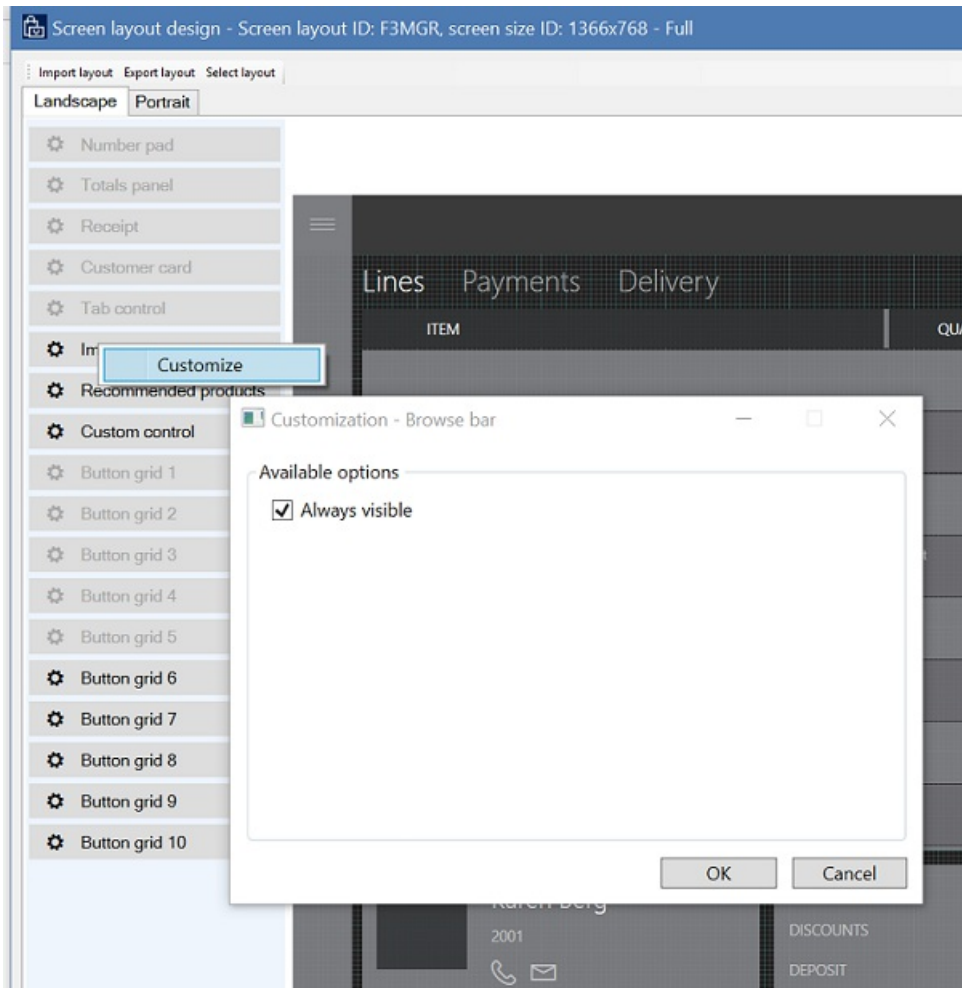
The Full layout designer lets users drag UI controls onto the POS **Transaction** screen and configure the settings of those controls.



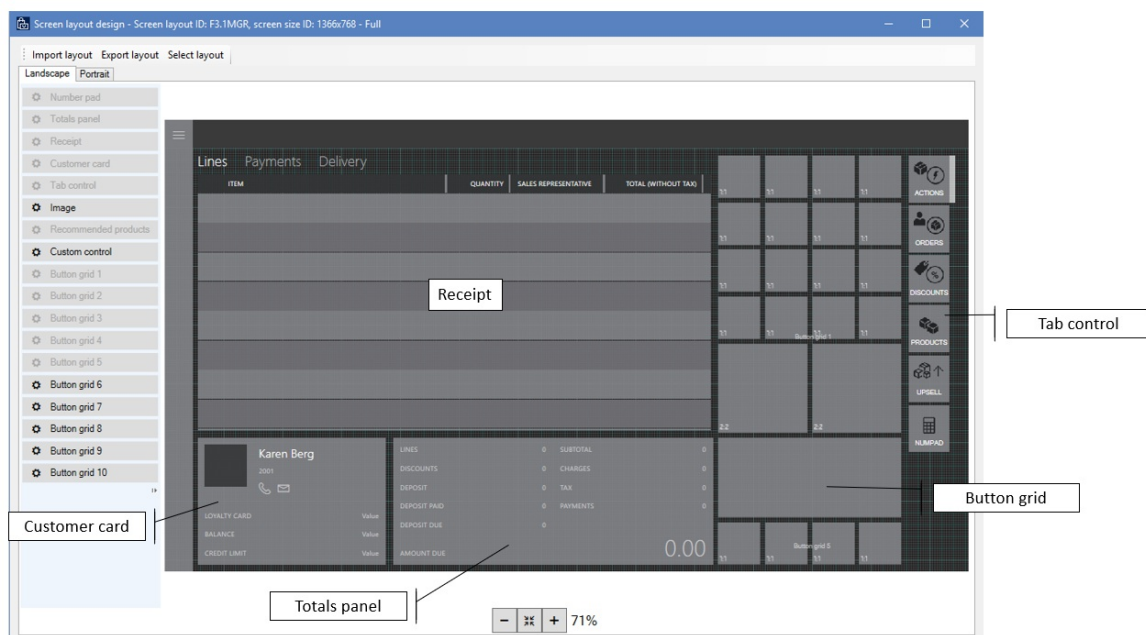
- **Import layout/Export layout** – You can export and import POS screen layout designs as XML files, so that you can easily reuse and share them across environments. It's important that you import layout designs for the correct layout sizes. Otherwise, UI elements might not fit correctly on the screen.
- **Landscape/Portrait** – If the POS device lets users switch between landscape and portrait modes, you must define a screen layout for each mode. The POS automatically detects screen rotation and shows the correct layout.
- **Layout grid** – The POS layout designer uses a 4-pixel grid. UI controls "snap" to the grid to help you correctly align the content.
- **Designer zoom** – You can zoom the designer view in and out to better view the content on the POS screen. This feature is useful when the screen resolution on the POS differs greatly from the resolution of the screen that is used in the designer.
- **Show/hide navigation bar** – For Full POS layouts, you can select whether the left navigation bar is visible on the **Transaction** screen. This feature is helpful for displays that have a lower resolution. To set



the visibility, right-click the navigation bar in the designer, and select or clear the **Always visible** check box. If the navigation bar is hidden, POS users can still access it by using the menu in the upper left.



- **POS controls** – The POS layout designer supports the following controls. You can configure many controls by right-clicking and using the shortcut menu.



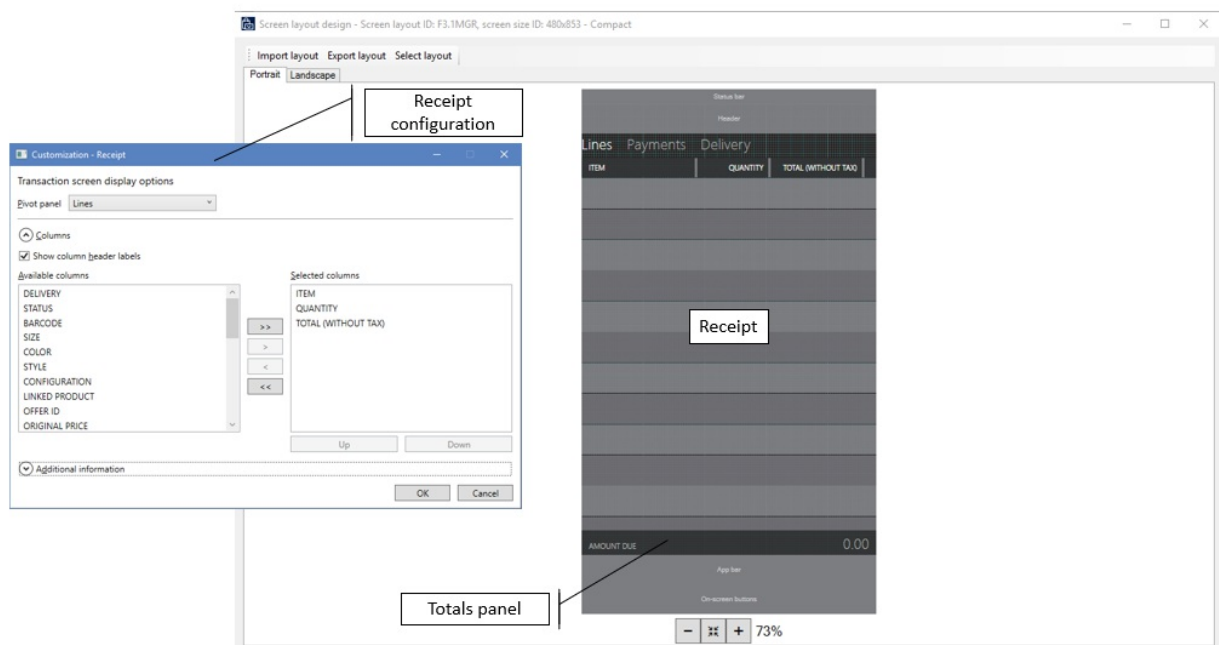
- **Number pad** – The number pad is the main mechanism for user input on the POS Transaction screen. You can configure the control so that the full number pad is shown. This option is ideal for touchscreen devices. Alternatively, you can configure it so that only the input field is shown. In this case, a physical keyboard is used for input. The number pad settings are available only for Full layouts.

For Compact layouts, the full number pad is always shown on the **Transaction** screen.

- **Totals panel** – You can configure the totals panel in either one column or two columns, to show values such as the line count, discount amount, charges, subtotal, and tax. Compact layouts support only a single column.
- **Receipt panel** – The receipt panel contains the sales lines, payment lines, and delivery information for the products and services that are processed in the POS. You can specify columns, widths, and placement. In Compact layouts, you can also configure additional information that appears in the row under the main line.
- **Customer card** – The customer card shows information about the customer who is associated with the current transaction. You can configure the customer card to hide or show additional information.
- **Tab control** – You can add the tab control to a screen layout, and then put other controls, such as the number pad, customer card, or button grids, in it. The tab control is a container that helps you fit more content on the screen. The tab control is available only for Full layouts.
- **Image** – You can use the image control to show the store's logo or another branding image on the **Transaction** screen. The image control is available only for Full layouts.
- **Recommended products** – If the recommended products control is configured for the environment, it shows product suggestions, based on machine learning.
- **Custom control** – The custom control acts as a placeholder in the screen layout and lets you reserve space for custom content. The custom control is available only for Full layouts.

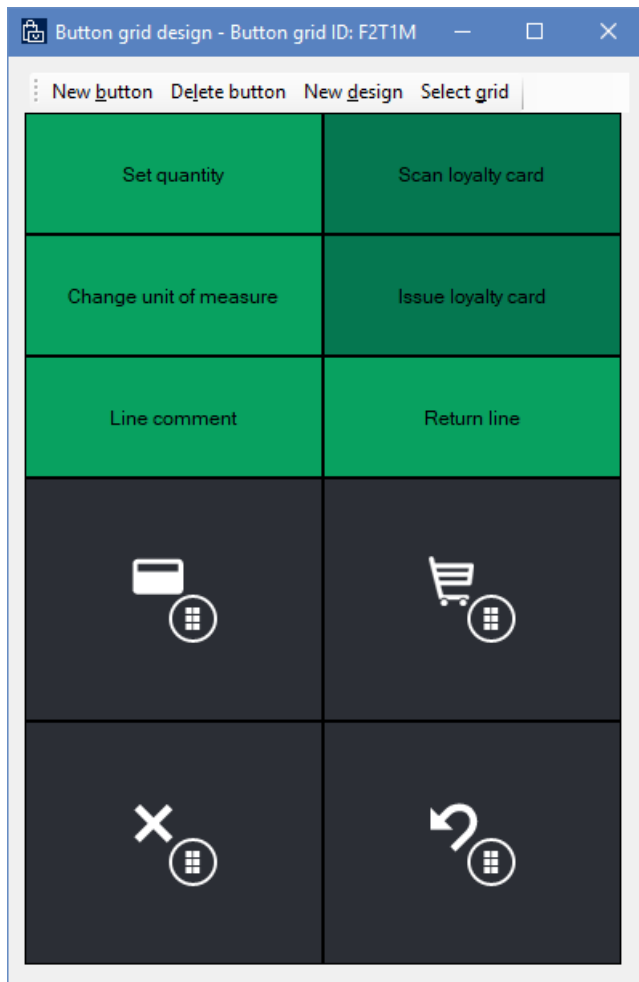
## Compact layout designer

Like the Full layout designer, the Compact layout designer lets you configure the POS screen layout for phones and small tablets. However, in this case, the layout itself is fixed. You can configure the controls in the layout by right-clicking and using the shortcut menu. However, you can't use drag-and-drop operations for additional content.



## Button grid designer

The button grid designer lets you configure button grids that can be used on the POS welcome screen and **Transaction** screen for both Full and Compact layouts. The same button grid can be used across layouts and layout types. Like the screen layout designer, the button grid designer uses the ClickOnce deployment technology to download, install, and start the latest version of the application every time that users access it. Be sure to check the browser requirements for ClickOnce. Some browsers, such as Google Chrome, require extensions.



- **New button** – Click to add a new button to the button grid. By default, new buttons appear in the upper-left corner of the grid. However, you can arrange buttons by dragging them in the layout.

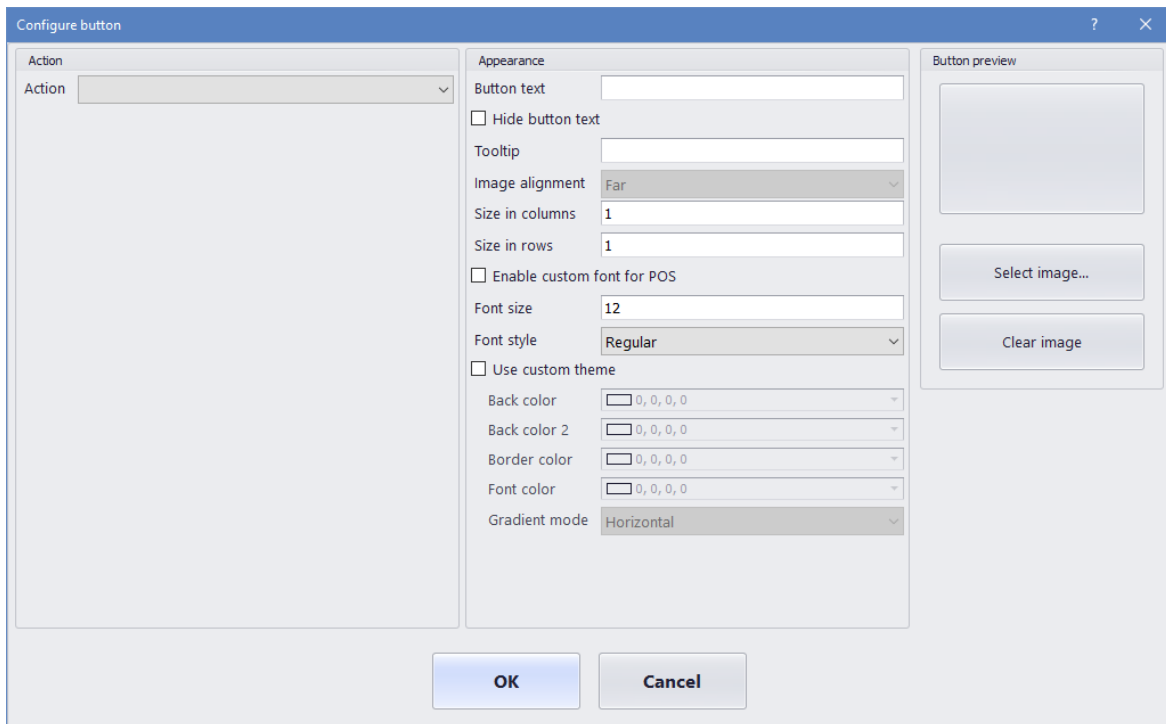
**IMPORTANT**

The contents of the button grid can overlap. When you arrange buttons, make sure that they don't hide other buttons.

- **New design** – Click to automatically set up a button grid layout by specifying the number of buttons per row and column.
- **Button properties** – You can configure button properties by right-clicking the button and using the shortcut menu.

**IMPORTANT**

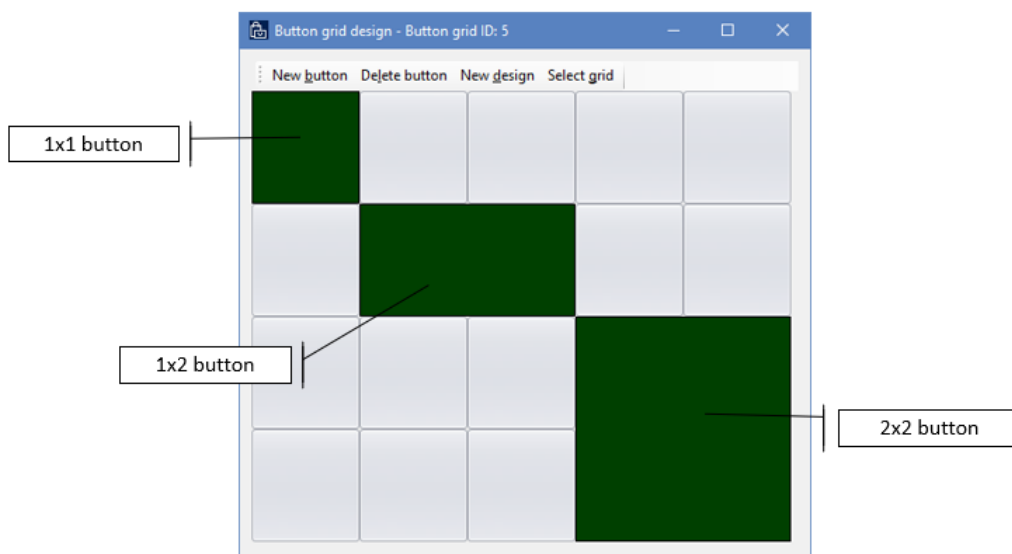
Some button grid settings apply only to Enterprise POS, not to Modern POS or Cloud POS.



- **Action** – In the list of applicable POS operations, select the operation that is invoked when the button is clicked in the POS.

For the list of supported POS operations, see [Online and offline point of sale \(POS\) operations](#).

- **Action parameters** – Some POS operations use additional parameters when they are invoked. For example, for the Add product operation, users can specify the product to add.
- **Button text** – Specify the text that appears on the button in the POS.
- **Hide button text** – Use this check box to hide or show the button text. Button text is often hidden for small buttons that show only an icon.
- **Tooltip** – Specify additional Help text that appears when users mouse over the button.
- **Size in columns/Size in rows** – You can specify how tall and wide the button is.



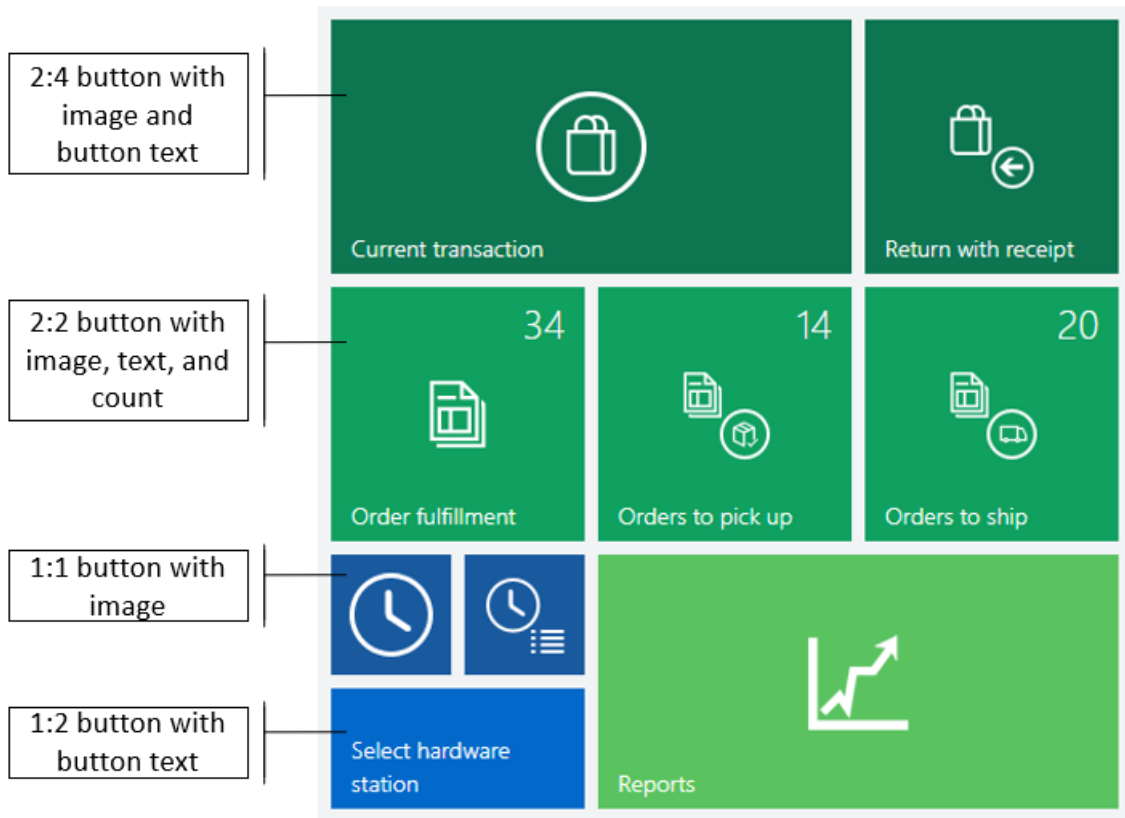
- **Custom font** – When you select the **Enable custom font for POS** check box, you can specify a font other than the default system font for the POS.

- **Custom theme** – By default, POS buttons use the accent color from the visual profile. When you select the **Use custom theme** check box, you can specify additional colors.

**NOTE**

Modern POS and Cloud POS use only the **Back color** and **Font color** values.

- **Button image** – Buttons can include images or icons. Select among the available images that are specified at **Retail and Commerce > Channel setup > POS setup > POS > Images**.



## Additional resources

[Install the Retail point of sale \(POS\) layout designer](#)

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Open URL in POS

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic describes how you can configure a button in Retail point of sale (POS) to open a URL. This feature does not require a code customization, and can be configured by someone in a non-developer role.

This feature allows configuration of a button in POS, using the button grid designer to open a URL. Currently, this is supported in the following configurations:

- Open in new window.
- Open within POS.
- Open a native app.

## Open in new window

This configuration defines whether to open the URL in a new window or within the app. When configured to open a web URL within the app, the side navigation panel and top bar of POS are visible and available for user interaction. When configured to open in a new window, the URL will open in a new app window on Modern POS for Windows, and in a new browser tab in all other POS clients. To enable this, you must configure the URL with the **Open in new window** option selected.

## Open within POS

Opening a web URL within POS is currently only supported for Modern POS on Windows. On other clients, this capability is under development and planned for release in future updates. To enable this, you must configure the URL with the **Open in new window** option not selected.

## Open a native app

This feature also allows you to specify non-web URLs to open a native app. For example, you can specify URL protocols such as MailTo, SIP, IM, or MSTEAMS, which can then be handled by respective native apps on the host device. To enable this, you must configure the URL with the **Open in new window** option selected.

- For Windows computers, see [Export or Import Default Application Associations](#) to set the default protocol associations if you are setting up your computer using Deployment Image Servicing and Management (DISM).
- If you are using MDM, such as Intune to manage your Windows computers, see [Policy CSP - ApplicationDefaults](#).
- If you are a developer building a custom website, see [Launch the default app for a URI](#).

## Open a native app seamlessly

Windows, iOS, and Android also allow opening of apps more seamlessly, based on app protocol association. If your app is not already configured to handle opening from a web browser, you may need a developer to configure this.

- For Windows, see [Enable apps for websites using app URI handlers](#).
- For iOS, see [Universal Links for Developers](#).
- For Android, see [Handling Android App Links](#).

CLIENT	OPEN IN NEW WINDOW	OPEN NATIVE APP	OPEN WITHIN POS	DETAILS
Modern POS on Windows	✓*	✓	✓	* Opens in new Modern POS window
Cloud POS	✓*	✓	X	* Opens in new browser tab
Modern POS on iOS	✓*	✓	X	* Opens in new browser tab
Modern POS on Android	✓*	✓	X	* Opens in new browser tab

## Before you begin

Before you begin, review how to configure [Screen layouts for the point of sale \(POS\)](#).

## Open URL in POS

To configure a URL to be opened in POS, perform the following steps.

1. In head office, go to **Retail and Commerce > Channel Setup > POS Setup > POS > Screen Layouts**.
2. Select **Button Grids > Designer**.
3. Create a new button.
4. Select **Button** properties.
5. Select **Open URL** as the action.
6. Enter the URL that you want to use.
7. Configure whether to open the URL in a new window.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Install the POS layout designer

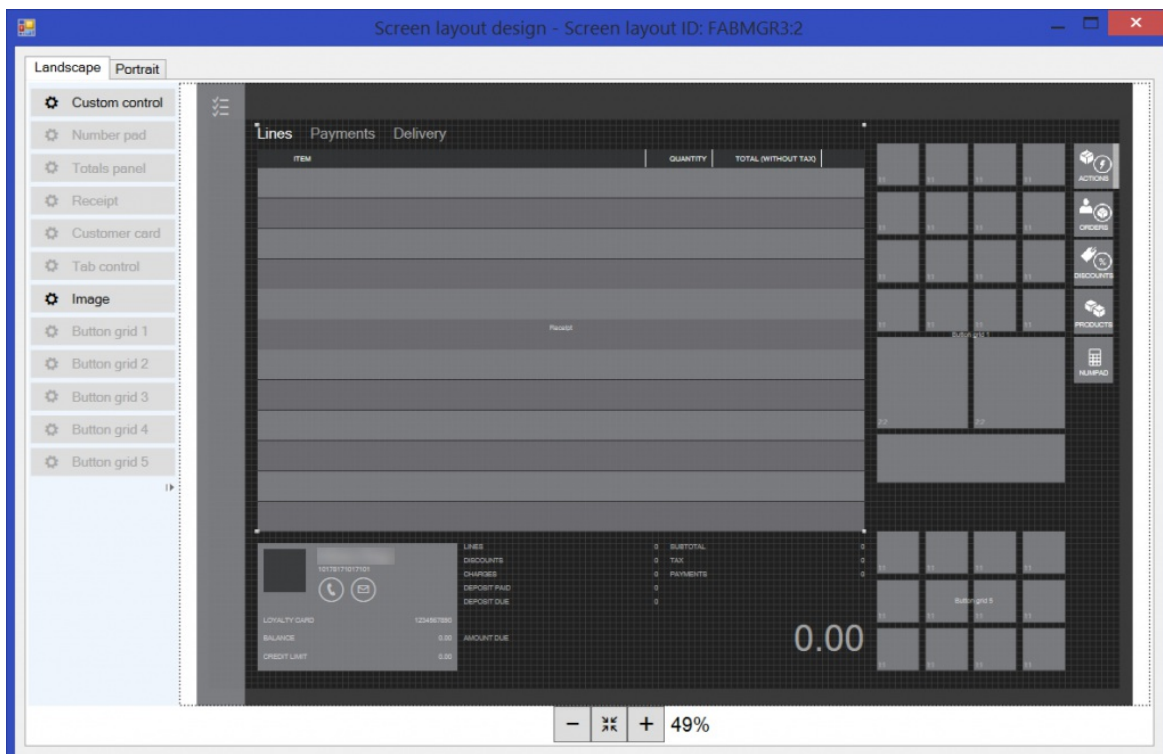
2/18/2021 • 2 minutes to read • [Edit Online](#)

You can use the one-click designer to design different Modern POS (MPOS) and Cloud POS layouts, in either Landscape mode or Portrait mode, for stores, registers, cashiers, and managers.

The graphical design interface for MPOS or Cloud POS is controlled by the till layout. A layout controls the position of various objects. Examples include the total layout, the item grid layout, the customer layout, the payment layout, and the layout of various menu buttons. Layouts also include the overall appearance of the sales interface that is presented to workers.

## Install the one-click designer

1. In Commerce, use the menu in the upper left to navigate to **Retail and Commerce > Channel setup > POS setup > POS > Screen layouts**.
2. Select any layout that has an application type of **Modern POS for Windows** or **Cloud POS**, and then click **Layout designer**.
3. On the notification bar that appears at the bottom of the Internet Explorer window, click **Open** to install the one-click designer. (The notification bar might appear in a different place in other browsers.)
4. In the **Application Run - Security Warning** message box that appears, click **Run** to install the Retail designer host. A progress indicator shows the progress of the installation.
5. After the installation is completed, on the **Sign in** page, enter your Commerce user name and password, and then click **Sign in** to start the designer.
6. After your credentials are validated and the designer starts, you can design your own layout or modify the existing layout.





# Troubleshoot the installation of the Layout designer

- When you click **Designer**, the prompt to download (or run) the installer doesn't appear, or your current security settings don't allow you to download the file.

## Solutions:

- In Internet Explorer, make sure that the pop-up blocker is disabled for this site. Click **Settings > Options > Privacy > Find Pop-up Blocker**, and change the setting, if a change is required.
- In Internet Explorer, add the Commerce URL to your trusted sites. Click **Settings > Options > Security > Trusted sites > Sites > Add**.
- The program doesn't start, and you're instructed to contact the vendor.

**Solution:** In Internet Explorer, add the Commerce URL to your trusted sites. Click **Setting > Options > Security > Trusted sites > Sites > Add**.

**Known issue:** The designer doesn't work correctly in the Google Chrome and Mozilla Firefox browsers. We are working to fix this issue.

## Additional resources

[Configure, install, and activate Retail Modern POS \(MPOS\)](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Extend the point of sale (POS) Dual display view

2/18/2021 • 10 minutes to read • [Edit Online](#)

This topic explains how to extend the point of sale (POS) Dual display view so that it shows custom information. This topic is applicable to Microsoft Dynamics 365 for Finance and Operations 7.2 or Microsoft Dynamics 365 Retail 7.2 with KB 4091080, and later versions.

You can extend the POS Dual display view by adding a custom control. In the custom control, you can add images, POS data lists, labels, and so on, to show custom information.

## NOTE

You can extend the POS Dual display view only by adding a custom control. The custom control will override the standard content that is shown in the POS Dual display view. The dual display custom control and other extension details information related to dual display will not be shown in the extension details view.

## Required steps

Here is an overview of the steps that are required in order to customize the POS Dual display view.

1. Configure the hardware profile to enable dual display.
2. Create a folder in the POS.Extensions project for extension of the POS Dual display view.
3. Add a new custom control that includes custom information.
4. Update the manifest.json and extensions.json files with the extension of the POS Dual display view.
5. Deploy the changes, and validate the customization.

## Scenario or business problem

You will add a custom control column in the POS Dual display view to show the cart details and information about the customer and the store employee.

## Configure the hardware profile to enable dual display

1. Sign in to the client.
2. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**.
3. Select the hardware profile that is linked to your register.
4. On the **Dual display** tab, set the **Dual display in use** option to **Yes**.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
6. Select the **Registers (1090)** job, and then select **Run now**.

## NOTE

You can find the end-to-end (E2E) sample in ...RetailSDK\POS\Extensions\DualDisplaySample.

## Add a new custom control for extension of the POS Dual display view

1. Start Microsoft Visual Studio 2015 in Administrator mode.
2. Open the **ModernPOS** solution from ...RetailSDK\POS.

3. Under the POS.Extensions project, create a folder that is named **DualDisplayExtension**.
4. Under the **DualDisplayExtension** folder, create a folder that is named **CustomControl**.
5. In the **CustomControl** folder, create a HTML file that is named **DualDisplayCustomControl.html**.

In the HTML file, you will add a data list control to show the cart details, and text controls to show the total amount, customer name, employee name, and sign-in status.

6. Copy the following code, and paste it into the **DualDisplayCustomControl.html** file.

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8" />
<title></title>
</head>
<body>
<!-- Note: The element ID is different from the ID generated by the POS extensibility framework. This
'template' ID is not used by the POS extensibility framework. -->
<script id="Microsoft_Pos_Extensibility_Samples_DualDisplay" type="text/html">
<div class="height100Percent width100Percent">
  <div class="height700 width100Percent no-shrink col marginBottom20">
    <div class="col grow marginBottom48">
      <div id="dualDisplayDataListSample" data-bind="msPosDataList: cartLinesDataList">
      </div>
    </div>
  </div>
  <div class="marginBottom20">
    <h2 class="marginLeft8" data-bind="text: cartTotalAmountLabel">Total Amount:</h2>
    <h2 class="marginLeft8" data-bind="text: cartTotalAmount"></h2>
    <h2 class="marginLeft8" data-bind="text: customerNameLabel">Customer Name:</h2>
    <h2 class="marginLeft8" data-bind="text: customerName"></h2>
    <h2 class="marginLeft8" data-bind="text: customerAccountNumberLabel">Customer Account Number:
  </h2>
    <h2 class="marginLeft8" data-bind="text: customerAccountNumber"></h2>
    <h2 class="marginLeft8" data-bind="text: isLoggedInOn() ? 'logged in' : 'logged out'"></h2>
    <h2 class="marginLeft8" data-bind="text: employeeNameLabel">Employee Name:</h2>
    <h2 class="marginLeft8" data-bind="text: employeeName"></h2>
  </div>
</div>
</script>
</body>
</html>
```

Next, you will add the resource file that is used to localize the field name.

7. Under the **DualDisplayExtension** folder, create a folder that is named **Resources**.
8. Under the **Resources** folder, create a folder that is named **Strings**.
9. Under the **Strings** folder, create a folder that is named **en-US**.
10. In the en-US folder add a new resources file and and change the file extension and name to resources.resjson and inside the resources.resjson file copy paste the below resource strings.

```

//=====
===
//===== Sample comment.
=====
//=====
===

{
  //===== extensions strings. =====

  "string_0" : "ID",
  "_string_0.comment" : "Item ID label text",

  "string_1" : "Name",
  "_string_1.comment" : "Item name label text",

  "string_2" : "Quantity",
  "_string_2.comment" : "Item cost label text",

  "string_3" : "Discount",
  "_string_3.comment" : "Total amount label text",

  "string_4" : "Cost",
  "_string_4.comment" : "Item cost label text",

  "string_5" : "Total Amount:",
  "_string_5.comment" : "Total amount label text",

  "string_6" : "Customer Name:",
  "_string_6.comment" : "Customer name label text",

  "string_7" : "Customer Account Number:",
  "_string_7.comment" : "Customer account number label text",

  "string_8" : "Employee Name:",
  "_string_8.comment" : "Employee name label text"
}

```

11. In the **CustomControl** folder, create a TypeScript file (.ts file) that is named **DualDisplayCustomControl.ts**.
12. Copy the following code, and paste it into the **DualDisplayCustomControl.ts** file. This code imports the relevant entities and context.

```

import {
  DualDisplayCustomControlBase,
  IDualDisplayCustomControlState,
  IDualDisplayCustomControlContext,
  CartChangedData,
  CustomerChangedData,
  LogOnStatusChangedData
} from "PosApi/Extend/DualDisplay";
import { DataList, IDataListState, SelectionMode } from "PosUISdk/Controls/DataList";
import { ObjectExtensions, StringExtensions } from "PosApi/TypeExtensions";
import { ProxyEntities } from "PosApi/Entities";

```

13. In the **DualDisplayCustomControl.ts** file, create a class that is named **DualDisplayCustomControl**, and extend it from the **DualDisplayCustomControlBase** class. You extend from the **DualDisplayCustomControlBase** class to get all the events that are exposed for dual display.

```

export default class DualDisplayCustomControl extends DualDisplayCustomControlBase { }

```

14. Inside the `DualDisplayCustomControl` class, add the following variables to get the cart, customer, and employee details.

```
private static readonly TEMPLATE_ID: string = "Microsoft_Pos_Extensibility_Samples_DualDisplay";

// The data list to bind against and display with cart line information.

public readonly cartLinesDataList: DataList<ProxyEntities.CartLine>;

// Computed values for binding against.

public readonly cartTotalAmount: Computed<number>;
public readonly customerName: Computed<string>;
public readonly customerAccountNumber: Computed<string>;
public readonly isLoggedInOn: Computed<boolean>;
public readonly employeeName: Computed<string>;

// Labels for binding against.

public readonly cartTotalAmountLabel: string;
public readonly customerNameLabel: string;
public readonly customerAccountNumberLabel: string;
public readonly employeeNameLabel: string;

// Observable values used for keeping track of state.

private readonly _cart: Observable<ProxyEntities.Cart>;
private readonly _cartLinesObservable: ObservableArray<ProxyEntities.CartLine>;
private readonly _customer: Observable<ProxyEntities.Customer>;
private readonly _loggedOn: Observable<boolean>;
private readonly _employee: Observable<ProxyEntities.Employee>; private _selectedTenderLines:
ProxyEntities.TenderLine[];
```

15. Create a class constructor method to initialize all the variables.

```
constructor(id: string, context: IDualDisplayCustomControlContext) {
    super(id, context);

    // Initializes labels.

    this.cartTotalAmountLabel = this.context.resources.getString("string_5");
    this.customerNameLabel = this.context.resources.getString("string_6");
    this.customerAccountNumberLabel = this.context.resources.getString("string_7");
    this.employeeNameLabel = this.context.resources.getString("string_8");

    // Initializes observable and computed values.

    this._cart = ko.observable(null);
    this._cartLinesObservable = ko.observableArray([]);
    this._customer = ko.observable(null);
    this._loggedOn = ko.observable(false);
    this._employee = ko.observable(null);
    this.cartTotalAmount = ko.computed(() => {
        return ObjectExtensions.isNullOrUndefined(this._cart()) ? 0.00 : this._cart().TotalAmount;
    });
    this.customerName = ko.computed(() => {
        return ObjectExtensions.isNullOrUndefined(this._customer()) ? StringExtensions.EMPTY :
this._customer().Name;
    });
    this.customerAccountNumber = ko.computed(() => {
        return ObjectExtensions.isNullOrUndefined(this._customer()) ? StringExtensions.EMPTY :
this._customer().AccountNumber;
    });
    this.isLoggedInOn = ko.computed(() => {
        return this._loggedOn();
    });
}
```

```

    },
    this.employeeName = ko.computed(() => {
        return ObjectExtensions.isNullOrUndefined(this._employee()) ? StringExtensions.EMPTY :
this._employee().Name;
    });
    this.cartChangedHandler = (data: CartChangedData) => {
        this._cart(data.cart);
        this._cartLinesObservable(ObjectExtensions.isNullOrUndefined(data.cart) ? [] :
data.cart.CartLines)
    };
    this.customerChangedHandler = (data: CustomerChangedData) => {
        this._customer(data.customer);
    };
    this.logOnStatusChangedHandler = (data: LogOnStatusChangedData) => {

        // Displays the busy indicator here, even though it's not necessary, in order to showcase and
test the scenario.

        this.isProcessing = true;
        window.setTimeout(() => {
            this.isProcessing = false;
        }, 1000);
        this._loggedOn(data.loggedOn);
        this._employee(data.employee);
    }

    // Initializes the cart lines data list

    let cartLinesDataListOptions: IDataListState<ProxyEntities.CartLine> = {
        selectionMode: SelectionMode.None,
        itemDataSource: this._cartLinesObservable,
        columns:[
            {
                title: context.resources.getString("string_0"), // ID
                ratio: 20,
                collapseOrder: 2,
                minWidth: 50,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.ItemId) ?
StringExtensions.EMPTY : cartLine.ItemId;
                }
            },
            {
                title: context.resources.getString("string_1"), // Name
                ratio: 50,
                collapseOrder: 4,
                minWidth: 100,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.Description) ?
StringExtensions.EMPTY : cartLine.Description;
                }
            },
            {
                title: context.resources.getString("string_2"), // Quantity
                ratio: 10,
                collapseOrder: 3,
                minWidth: 50,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.Quantity) ?
StringExtensions.EMPTY : cartLine.Quantity.toString();
                }
            },
            {
                title: context.resources.getString("string_3"), // Discount
                ratio: 10,
                collapseOrder: 1,
                minWidth: 50,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.DiscountAmount) ?
StringExtensions.EMPTY : cartLine.DiscountAmount.toString();
                }
            }
        ]
    };

```

```

StringExtensions.EMPTY : cartLine.DiscountAmount.toString();
    }
    },
    {
        title: context.resources.getString("string_4"), // Cost
        ratio: 10,
        collapseOrder: 5,
        minWidth: 50,
        computeValue: (cartLine: ProxyEntities.CartLine): string => {
            return ObjectExtensions.IsNullOrEmpty(cartLine.TotalAmount) ?
StringExtensions.EMPTY : cartLine.TotalAmount.toString();
        }
    }
    ]
};
this.cartLinesDataList = new DataList(cartLinesDataListOptions);

// Logs the completion of constructing the DualDisplayCustomControl.

this.context.logger.logInformational("DualDisplayCustomControl constructed",
this.context.logger.getNewCorrelationId());
}

```

16. Add the **onReady** method to bind the control to the specified HTML element.

```

/**
 * Binds the control to the specified element.
 * @param {HTMLElement} element The element to which the control should be bound.
 */
public onReady(element: HTMLElement): void {
    ko.applyBindingsToNode(element, {
        template: {
            name: DualDisplayCustomControl.TEMPLATE_ID,
            data: this
        }
    });
}

```

17. Add the **init** method to initialize all the controls.

```

/**
 * Initializes the control.
 * @param {IDualDisplayCustomControlState} state The initial state of the page used to initialize the
control.
 */
public init(state: IDualDisplayCustomControlState): void {
    this._cart(state.cart);
    this._customer(state.customer);
    this._loggedOn(state.loggedOn);
    this._employee(state.employee)
}

```

Here is what the overall class should look like.

```

import {
    DualDisplayCustomControlBase,
    IDualDisplayCustomControlState,
    IDualDisplayCustomControlContext,
    CartChangedData,
    CustomerChangedData,
    LogOnStatusChangedData
} from "PosApi/Extend/DualDisplay";

```

```

import { DataList, IDataListState, SelectionMode } from "PosUISdk/Controls/DataList";
import { ObjectExtensions, StringExtensions } from "PosApi/TypeExtensions";
import { ProxyEntities } from "PosApi/Entities";
export default class DualDisplayCustomControl extends DualDisplayCustomControlBase {
    private static readonly TEMPLATE_ID: string = "Microsoft_Pos_Extensibility_Samples_DualDisplay";

    // The data list to bind against and display with cart line information.

    public readonly cartLinesDataList: DataList<ProxyEntities.CartLine>;

    // Computed values for binding against.

    public readonly cartTotalAmount: Computed<number>;
    public readonly customerName: Computed<string>;
    public readonly customerAccountNumber: Computed<string>;
    public readonly isLoggedInOn: Computed<boolean>;
    public readonly employeeName: Computed<string>;

    // Labels for binding against.

    public readonly cartTotalAmountLabel: string;
    public readonly customerNameLabel: string;
    public readonly customerAccountNumberLabel: string;
    public readonly employeeNameLabel: string;

    // Observable values used for keeping track of state.

    private readonly _cart: Observable<ProxyEntities.Cart>;
    private readonly _cartLinesObservable: ObservableArray<ProxyEntities.CartLine>;
    private readonly _customer: Observable<ProxyEntities.Customer>;
    private readonly _loggedOn: Observable<boolean>;
    private readonly _employee: Observable<ProxyEntities.Employee>;
    constructor(id: string, context: IDualDisplayCustomControlContext) {
        super(id, context);

        // Initializes labels.

        this.cartTotalAmountLabel = this.context.resources.getString("string_5");
        this.customerNameLabel = this.context.resources.getString("string_6");
        this.customerAccountNumberLabel = this.context.resources.getString("string_7");
        this.employeeNameLabel = this.context.resources.getString("string_8");

        // Initializes observable and computed values.

        this._cart = ko.observable(null);
        this._cartLinesObservable = ko.observableArray([]);
        this._customer = ko.observable(null);
        this._loggedOn = ko.observable(false);
        this._employee = ko.observable(null);
        this.cartTotalAmount = ko.computed(() => {
            return ObjectExtensions.isNullOrUndefined(this._cart()) ? 0.00 :
this._cart().TotalAmount;
        });
        this.customerName = ko.computed(() => {
            return ObjectExtensions.isNullOrUndefined(this._customer()) ? StringExtensions.EMPTY :
this._customer().Name;
        });
        this.customerAccountNumber = ko.computed(() => {
            return ObjectExtensions.isNullOrUndefined(this._customer()) ? StringExtensions.EMPTY :
this._customer().AccountNumber;
        });
        this.isLoggedInOn = ko.computed(() => {
            return this._loggedOn();
        });
        this.employeeName = ko.computed(() => {
            return ObjectExtensions.isNullOrUndefined(this._employee()) ? StringExtensions.EMPTY :
this._employee().Name;
        });
        this.cartChangedHandler = (data: CartChangedData) => {

```



```

        this._cart(data.cart);
        this._cartLinesObservable(ObjectExtensions.isNullOrUndefined(data.cart) ?[] :
data.cart.CartLines)
    };
    this.customerChangedHandler = (data: CustomerChangedData) => {
        this._customer(data.customer);
    };
    this.logOnStatusChangedHandler = (data: LogOnStatusChangedData) => {

        // Displays the busy indicator here, even though it's not necessary, in order to showcase
and test the scenario.

        this.isProcessing = true;
        window.setTimeout(() => {
            this.isProcessing = false;
        }, 1000);
        this._loggedOn(data.loggedOn);
        this._employee(data.employee);
    }

    // Initializes the cart lines data list

    let cartLinesDataListOptions: IDataListState<ProxyEntities.CartLine> = {
        selectionMode: SelectionMode.None,
        itemDataSource: this._cartLinesObservable,
        columns:[
            {
                title: context.resources.getString("string_0"), // ID
                ratio: 20,
                collapseOrder: 2,
                minWidth: 50,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.ItemId) ?
StringExtensions.EMPTY : cartLine.ItemId;
                }
            },
            {
                title: context.resources.getString("string_1"), // Name
                ratio: 50,
                collapseOrder: 4,
                minWidth: 100,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.Description) ?
StringExtensions.EMPTY : cartLine.Description;
                }
            },
            {
                title: context.resources.getString("string_2"), // Quantity
                ratio: 10,
                collapseOrder: 3,
                minWidth: 50,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.Quantity) ?
StringExtensions.EMPTY : cartLine.Quantity.toString();
                }
            },
            {
                title: context.resources.getString("string_3"), // Discount
                ratio: 10,
                collapseOrder: 1,
                minWidth: 50,
                computeValue: (cartLine: ProxyEntities.CartLine): string => {
                    return ObjectExtensions.isNullOrUndefined(cartLine.DiscountAmount) ?
StringExtensions.EMPTY : cartLine.DiscountAmount.toString();
                }
            },
            {
                title: context.resources.getString("string_4"), // Cost
                ratio: 10.

```

```

        collapseOrder: 5,
        minWidth: 50,
        computeValue: (cartLine: ProxyEntities.CartLine): string => {
            return ObjectExtensions.isNullOrUndefined(cartLine.TotalAmount) ?
StringExtensions.EMPTY : cartLine.TotalAmount.toString();
        }
    }
}
];
};
this.cartLinesDataList = new DataList(cartLinesDataListOptions);

// Logs the completion of constructing the DualDisplayCustomControl.

this.context.logger.logInformational("DualDisplayCustomControl constructed",
this.context.logger.getNewCorrelationId());
}

/**
 * Binds the control to the specified element.
 * @param {HTMLElement} element The element to which the control should be bound.
 */

public onReady(element: HTMLElement): void {
    ko.applyBindingsToNode(element, {
        template: {
            name: DualDisplayCustomControl.TEMPLATE_ID,
            data: this
        }
    });
}

/**
 * Initializes the control.
 * @param {IDualDisplayCustomControlState} state The initial state of the page used to initialize
the control.
 */

public init(state: IDualDisplayCustomControlState): void {
    this._cart(state.cart);
    this._customer(state.customer);
    this._loggedOn(state.loggedOn);
    this._employee(state.employee)
}
}
}

```

18. In the **DualDisplayExtension** folder, create a JavaScript Object Notation (JSON) file (.json file) that is named **manifest.json**.
19. Copy the following code, and paste it into the **manifest.json** file. Delete the default generated code before you copy this code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_DualDisplaySample",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "resources": {
      "supportedUICultures": [ "en-US" ],
      "fallbackUICulture": "en-US",
      "culturesDirectoryPath": "Resources/Strings",
      "stringResourcesFileName": "resources.resjson"
    },
    "dualDisplay": {
      "customControl": {
        "controlName": "DualDisplayCustomControl",
        "htmlPath": "CustomControl/DualDisplayCustomControl.html",
        "modulePath": "CustomControl/DualDisplayCustomControl"
      }
    }
  }
}

```

20. Open the **extensions.json** file under the **POS.Extensions** project, and update it with the **DualDisplayExtension** samples. In that way, the POS will include this extension during runtime.

```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions"
    },
    {
      "baseUrl": " SampleExtensions2"
    },
    {
      "baseUrl": " DualDisplayExtension"
    }
  ]
}

```

21. Open the **tsconfig.json** file, and comment out the extension package folders in the exclude list. The POS will use this file to include or exclude the extension. By default, the list contains all the excluded extensions. If you want to include an extension as part of the POS, add the name of the extension folder, and comment out the extension in the extension list, as shown here.

```

"exclude": [
  "AuditEventExtensionSample",
  "B2BSample",
  "CustomerSearchWithAttributesSample",
  "FiscalRegisterSample",
  "PaymentSample",
  "PromotionsSample",
  "SalesTransactionSignatureSample",
  //"SampleExtensions2",
  "SampleExtensions",
  "StoreHoursSample",
  "SuspendTransactionReceiptSample"
  //"SampleExtensions",
  //"DualDisplayExtension"
],

```

22. Compile and rebuild the project.

## Validate the customization

1. Sign in to Retail Modern POS by using **000160** as the operator ID and **123** as the password.
2. On the welcome screen, select the **Current transaction** button.
3. Add any item to the transaction. For example, add item number **0005**.
4. Add any customer to transaction. For example, add **Karen Berg**.
5. The dual display should show the cart, total, employee, and customer details.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

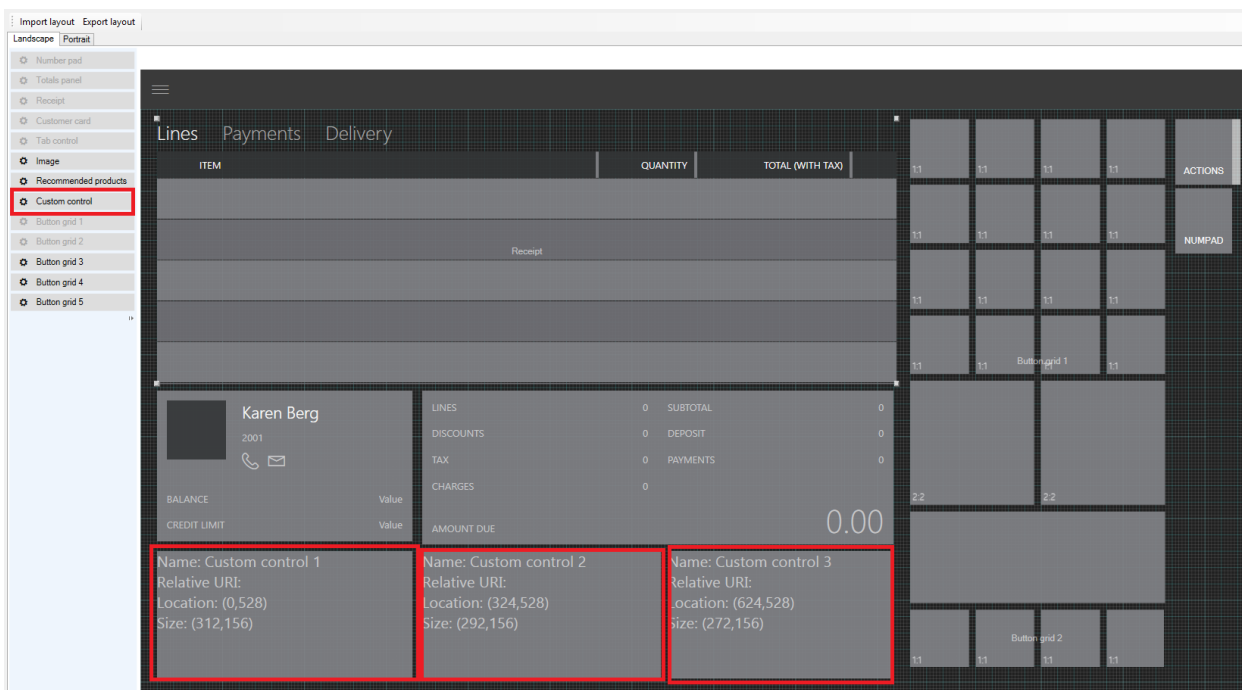
# Add custom controls to POS views

2/18/2021 • 5 minutes to read • [Edit Online](#)

To enhance the information that appears in the views in Microsoft Dynamics 365 Commerce POS, you can add custom controls. A custom control lets you add your own custom information to the existing POS views. Custom controls can be implemented by using the POS extension framework.

In Cart view, you can add custom controls by using the POS screen layout designer. In this case, you can drag a custom control to a location of your choice, and you can also set the height and width of the control. You then write the extension logic in the extension project.

For example, in the following illustration, three custom controls were added by using the screen layout designer.



Currently, only Cart view lets you use the screen layout designer to add custom controls. For all other screens, you should do the layout in the extension project. One advantage of using the screen layout designer is that you can drag the custom control wherever you want on the screen. On other screens, the position is fixed, but you can modify the position by specifying the height and width.

## Custom control matrix

The following table shows the views that support custom controls in POS.

POS VIEW	SUPPORTS CUSTOM CONTROLS	SUPPORTS SCREEN LAYOUT DESIGNER
Cart view/Transaction page	Yes	Yes
Customer details view	Yes	No
Product details view	Yes	No
Customer Add/Edit view	Yes	No

POS VIEW	SUPPORTS CUSTOM CONTROLS	SUPPORTS SCREEN LAYOUT DESIGNER
Address Add/Edit view	Yes	No

#### NOTE

Custom controls are supported only in the following product versions:

- **For views that aren't based on the screen layout designer:** Microsoft Dynamics 365 for Finance and Operations App update 3 and Microsoft Dynamics 365 Retail App update 3
- **For views that are based on the screen layout designer:** Microsoft Dynamics 365 for Finance and Operations App update 4 and Microsoft Dynamics 365 Retail App update 4

## Create a custom control

The following example shows how you can use extension to add custom controls to one of the existing POS views. For this example, we want information about product availability to appear in the product details view. To show this information, we will add a custom data list that has four columns: **Location**, **Inventory**, **Reserved**, and **Ordered**. You can use the same procedure to show other custom information in the POS views.

1. On the developer virtual machine (VM), start Microsoft Visual Studio 2015.
2. Open the **ModernPos.sln** file from RetailSDK\POS.
3. In the **POS.Extensions** project, add a new folder, and name it **SampleExtensions**.
4. In the new **SampleExtensions** folder, add another folder, and name it **ViewExtensions**.
5. In the **ViewExtensions** folder, add another folder, and name it **SimpleProductDetails**.

#### NOTE

If you're extending a view, you should give the folder the same name as the view, to make navigation and code maintenance easier.

6. In the **SimpleProductDetails** folder, add a new .html file, and name it **ProductAvailabilityPanel.html**. Also add a new .ts file, and name it **ProductAvailabilityPanel.ts**. In the .html file, you can add whatever information you want the custom control to show. In the .ts file, you add the corresponding logic.

A custom control is a simple HTML page that has your custom information.

7. Open the **ProductAvailabilityPanel.html** file, and paste the following code into it.

```

<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
    <!-- Note: The element ID differs from the ID that is generated by the POS extensibility
    framework. This 'template' ID isn't used by the POS extensibility framework. -->
    <script id="Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel"
    type="text/html">
      <h2 class="marginTop8 marginBottom8" data-bind="text: title"></h2>
      <div class="width400 grow col">
        <div id="Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel_DataList"
        data-bind="msPosDataList: dataList"></div>
      </div>
    </script>
  </body>
</html>

```

In the file, you add the POS data list control to show the product availability information. You also specify the width of the control.

You can copy the full code from

RetailSDK\Code\POS\Extensions\SampleExtensions\ViewExtensions\SimpleProductDetails\ProductAvailabilityPanel.html.

8. Open the **ProductAvailabilityPanel.ts** file, and paste the following code into it.

```

import {
  SimpleProductDetailsCustomControlBase,
  ISimpleProductDetailsCustomControlState,
  ISimpleProductDetailsCustomControlContext
} from "PosApi/Extend/Views/SimpleProductDetailsView";
import { ProxyEntities } from "PosApi/Entities";
import { ArrayExtensions } from "PosApi/TypeExtensions";
import { DataList, SelectionMode } from "PosUISdk/Controls/DataList";

```

To write our custom logic, we imported the list of controls and other data objects from the POS application programming interface (API).

## Add constructor and initialize data list

Next, you must add the constructor and initialize the data list with the product availability information. In this way, when you navigate to the page, the product availability information is loaded.

### NOTE

We didn't copy the source code here, but you can copy the full code from RetailSDK\Code\POS\Extensions\SampleExtensions\ViewExtensions\SimpleProductDetails\ProductAvailabilityPanel.ts.

1. In your **SampleExtensions** folder, add a new .json file, name it **manifest.json**, and paste the following code into it.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "resources": {
      "supportedUICultures": [ "en-US" ],
      "fallbackUICulture": "en-US",
      "culturesDirectoryPath": "Resources/Strings",
      "stringResourcesFileName": "resources.resjson",
      "cultureInfoOverridesFilePath": "Resources/cultureInfoOverrides.json"
    },
    "extend": {
      "views": {
        "SimpleProductDetailsView": {
          "controlsConfig": {
            "customControls": [
              {
                "controlName": "productAvailabilityPanel",
                "htmlPath":
"ViewExtensions/SimpleProductDetails/ProductAvailabilityPanel.html",
                "modulePath":
"ViewExtensions/SimpleProductDetails/ProductAvailabilityPanel"
              }
            ]
          }
        }
      }
    }
  }
}

```

During runtime, the manifest informs the POS that a custom control has been added in **SimpleProductDetailsView**. In the preceding code example, we included all the required metadata that the POS requires in order to load the control.

- **Extend** – Inform the POS that there is an extension for an existing POS feature.
- **Views** – Specify that an existing POS view is being extended.
- **View Name** – Specify the view that is being extended.
- **Controls config** – Specify the control that you are adding, such as **Custom control**.
- **Control metadata** – Specify the name, the path of the .html file path, and the path of the typescript module (that is, the .ts file).

2. Open the **extensions.json** file, and paste the following code into it.

```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions"
    }
  ]
}

```

In the **extensions.json** file, you specify the various extensions that you have. In this case, we added a new extension folder. Therefore, we must specify that folder.



## NOTE

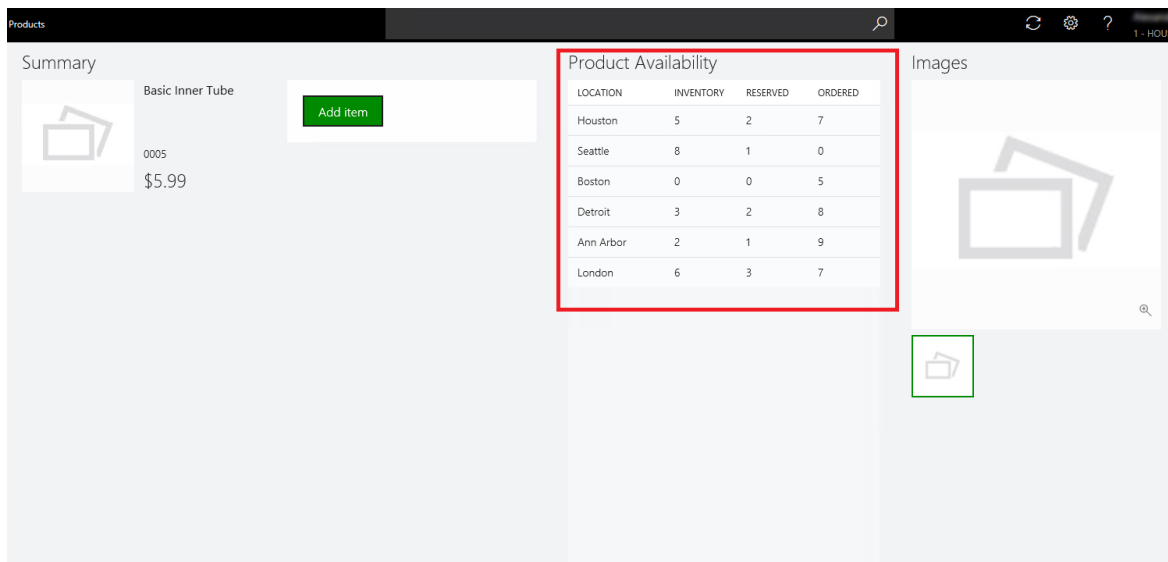
Each extension folder or package that you specify here should have a manifest.

3. Open the `tsconfig.json` file, and include your extension. Paste the following code into the file.

```
"extends": "../tsconfigs/tsmodulesconfig",  
"exclude": [  
  // "SampleExtensions"  
],
```

## Test the extension

1. Press F5, and deploy the POS to test your customization.
2. After the POS is started, sign in. Then search for any product, and open to product details view. You should now see the custom control that you added. Here is an example.



You can copy the full code for this sample from `RetailSDK\Code\POS\Extensions\SampleExtensions\ViewExtensions\SimpleProductDetails`.

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add POS operations to POS layouts by using Button grid designer

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic explains how to create a new point of sale (POS) operation and add it to the POS layout by using Button grid designer. This topic applies to the following applications where Platform update 8 and the Application update 4 hotfix are installed:

- Finance
- Dynamics 365 Commerce

If you want your business logic to be run in the POS when users click a button, you should create POS operations. POS operations can run multiple activities or workflows. For example, they can open a new view, ask for user input, or run business logic. All standard and custom POS operations support pre-triggers and post-triggers.

## NOTE

If logic should be run as part of another workflow, or if no button click is required, create POS request/response application programming interfaces (APIs). POS operations aren't required in these scenarios.

Every operation should implement the following elements:

- **Operation request** – The operation request is extended from the **ExtensionOperationRequestBase** class. It contains all the input that is required in order to run the operation.
- **Operation response** – The operation response is extended from the **Response** class. It contains the whole response, based on execution of the operation.
- **Operation factory** – The operation factory links the button click for the operation with the operation handler.
- **Operation handler** – The operation handler is extended from the **ExtensionOperationRequestHandlerBase** class. It contains core logic for the operation. All the business logic should be written in the handler, and it should return the operation response after the operation is run.

## Create a POS operation

This section explains how to create a sample operation that does simplified end-of-day (EOD) processing. This operation calls the standard Tender removal, Safe drop, Tender declaration, and Close shift operations in a sequence. Therefore, this one operation combines multiples steps. It is run based on the conditions that you define.

## NOTE

You can create a new operation and run your own custom logic, you can call existing POS operations, such as Add item to cart and Apply line discount, or you can call existing APIs, such as Get current cart and Set extension properties.

1. Start Microsoft Visual Studio 2015 in administrator mode.
2. From ...\**RetailSDK\POSOpen**, open the **ModernPOS** solution.
3. Under the **POS.Extensions** project, create a folder that is named **EODSample**.

4. Under the **EODSample** folder, create a folder that is named **Operations**.

### Create the operation request class

1. In the **Operations** folder, create a typescript (.ts) file that is named **EndOfDayOperationRequest.ts**.
2. Open the **EndOfDayOperationRequest.ts** file, and add the following **import** statements to import the relevant entities and context.

```
import { ExtensionOperationRequestBase } from "PosApi/Create/Operations";
import EndOfDayOperationResponse from "../EndOfDayOperationResponse";
```

3. Add a class that is named **EndOfDayOperationRequest**, and extend it from the **ExtensionOperationRequestBase** class. In this example, the operation ID in the **super** method is initialized to **5001**. However, you can use any operation ID starting from 4001. Operation IDs 0 through 4000 are reserved for internal POS operations, and no two operations should have the same operation ID. Additionally, the custom parameters field appears in Button grid designer properties only if the operation ID is 4001 or higher. (You can use custom parameters field to pass parameters to the POS operation from Retail headquarters).

```
/**
 * (Sample) Operation request for executing end of day operations.
 */
export default class EndOfDayOperationRequest<TResponse> extends EndOfDayOperationResponse<TResponse> extends ExtensionOperationRequestBase<TResponse> {
  constructor(correlationId: string) {
    super(5001, correlationId);
  }
}
```

### Create the operation response class

1. In the **Operations** folder, create a typescript (.ts) file that is named **EndOfDayOperationResponse.ts**.
2. Open the **EndOfDayOperationResponse.ts** file, and add the following **import** statement to import the relevant entities and context.

```
import { Response } from "PosApi/Create/RequestHandlers";
```

3. Add a class that is named **EndOfDayOperationResponse**, and extend it from the **Response** class.

```
/**
 * (Sample) Operation response of executing end of day operations.
 */
export default class EndOfDayOperationResponse extends Response { }
```

### Create the operation handler class

1. In the **Operations** folder, create a typescript (.ts) file that is named **EndOfDayOperationRequestHandler.ts**.
2. Open the **EndOfDayOperationRequestHandler.ts** file, and add the following **import** statements to import the relevant entities and context.

```
import { ExtensionOperationRequestType, ExtensionOperationRequestHandlerBase } from
"PosApi/Create/Operations";
import { CloseShiftOperationRequest, CloseShiftOperationResponse } from "PosApi/Consume/Shifts";
import { SafeDropOperationRequest, SafeDropOperationResponse } from "PosApi/Consume/StoreOperations";
import { TenderDeclarationOperationRequest, TenderDeclarationOperationResponse } from
"PosApi/Consume/StoreOperations";
import { TenderRemovalOperationRequest, TenderRemovalOperationResponse } from
"PosApi/Consume/StoreOperations";
import EndOfDayOperationResponse from "./EndOfDayOperationResponse";
import EndOfDayOperationRequest from "./EndOfDayOperationRequest";
import { ClientEntities } from "PosApi/Entities";|
```

3. Add a class that is named **EndOfDayOperationRequestHandler**, and extend it from the **ExtensionOperationRequestHandlerBase** class.

Each handler should implement two methods:

- supportedRequestType
- executeAsync

```
export default class EndOfDayOperationRequestHandler<TResponse extends EndOfDayOperationResponse>
extends ExtensionOperationRequestHandlerBase<TResponse> {}
```

4. Add the supported request type in the class.

```
/**
 * Gets the supported request type.
 * @return {RequestType<TResponse>} The supported request type.
 */
public supportedRequestType(): ExtensionOperationRequestType<TResponse> {
    return EndOfDayOperationRequest;
}
```

5. Implement the **executeAsync** method.

```
/**
 * Executes the request handler asynchronously.
 * @param {EndOfDayOperationRequest<TResponse>} request The request.
 * @return {Promise<ICancelableDataResult<TResponse>>} The cancelable async result containing the
response.
 */
public executeAsync(printRequest: EndOfDayOperationRequest<TResponse>):
Promise<ClientEntities.ICancelableDataResult<TResponse>> {
    this.context.logger.logInformational("Log message from PrintOperationRequestHandler
executeAsync().", this.context.logger.getNewCorrelationId());
    // Tender Removal
    let tenderRemovalRequest: TenderRemovalOperationRequest<TenderRemovalOperationResponse> =
new TenderRemovalOperationRequest(this.context.logger.getNewCorrelationId());
    return this.context.runtime.executeAsync(tenderRemovalRequest).then((result:
ClientEntities.ICancelableDataResult<TenderRemovalOperationResponse>)
: Promise<Commerce.Client.Entities.ICancelableDataResult<SafeDropOperationResponse>> => {
        // Safe Drop
        if (!result.canceled) {
            let safeDropRequest: SafeDropOperationRequest<SafeDropOperationResponse> =
new SafeDropOperationRequest(this.context.logger.getNewCorrelationId());
            return this.context.runtime.executeAsync(safeDropRequest);
        } else {
            return Promise.resolve({
                canceled: true,
                data: null
            });
        }
    });
}
```

```

    }).then((result: Commerce.Client.Entities.ICancelableDataResult<SafeDropOperationResponse>)
    : Promise<ClientEntities.ICancelableDataResult<TenderDeclarationOperationResponse>> => {
        // Tender Declaration
        if (!result.canceled) {
            let tenderDeclarationRequest:
TenderDeclarationOperationRequest<TenderDeclarationOperationResponse> =
            new TenderDeclarationOperationRequest(this.context.logger.getNewCorrelationId());
            return this.context.runtime.executeAsync(tenderDeclarationRequest);
        } else {
            return Promise.resolve({
                canceled: true,
                data: null
            });
        }
    }).then((result: ClientEntities.ICancelableDataResult<TenderDeclarationOperationResponse>)
    : Promise<ClientEntities.ICancelableDataResult<CloseShiftOperationResponse>> => {
        // Close Shift
        if (!result.canceled) {
            return new Promise(
                (resolve: (value?: ClientEntities.ICancelableDataResult<CloseShiftOperationResponse>)
=> void, reject: (reason?: any) => void) => {
                    // A delay of ten seconds is added here as a work-around for issues with printing
a second receipt to the windows driver
                    // printer before the first dialog is closed. A ten second delay gives the user a
chance to close the first dialog before
                    // the issue occurs.
                    setTimeout(() => { resolve(null); }, 10000);
                }).then(() => {
                    let closeShiftOperationRequest:
CloseShiftOperationRequest<CloseShiftOperationResponse> =
                    new CloseShiftOperationRequest(this.context.logger.getNewCorrelationId());
                    return this.context.runtime.executeAsync(closeShiftOperationRequest);
                });
        } else {
            return Promise.resolve({
                canceled: true,
                data: null
            });
        }
    }).then((result: ClientEntities.ICancelableDataResult<CloseShiftOperationResponse>)
    : ClientEntities.ICancelableDataResult<EndOfDayOperationResponse> => {
        return <ClientEntities.ICancelableDataResult<EndOfDayOperationResponse>>{
            canceled: result.canceled,
            data: result.canceled ? null : new EndOfDayOperationResponse()
        };
    });
}
}
}
}

```

The overall code should look like this.

```

import { ExtensionOperationRequestType, ExtensionOperationRequestHandlerBase } from
"PosApi/Create/Operations";
import { CloseShiftOperationRequest, CloseShiftOperationResponse } from "PosApi/Consume/Shifts";
import { SafeDropOperationRequest, SafeDropOperationResponse } from "PosApi/Consume/StoreOperations";
import { TenderDeclarationOperationRequest, TenderDeclarationOperationResponse } from
"PosApi/Consume/StoreOperations";
import { TenderRemovalOperationRequest, TenderRemovalOperationResponse } from
"PosApi/Consume/StoreOperations";
import EndOfDayOperationResponse from "./EndOfDayOperationResponse";
import EndOfDayOperationRequest from "./EndOfDayOperationRequest";
import { ClientEntities } from "PosApi/Entities";
/**
 * (Sample) Request handler for the EndOfDayOperationRequest class.
 */
export default class EndOfDayOperationRequestHandler<TResponse extends EndOfDayOperationResponse>
extends ExtensionOperationRequestHandlerBase<TResponse> {

```

```

/**
 * Gets the supported request type.
 * @return {RequestType<TResponse>} The supported request type.
 */
public supportedRequestType(): ExtensionOperationRequestType<TResponse> {
    return EndOfDayOperationRequest;
}
/**
 * Executes the request handler asynchronously.
 * @param {EndOfDayOperationRequest<TResponse>} request The request.
 * @return {Promise<ICancelableDataResult<TResponse>>} The cancelable async result containing the
response.
 */
public executeAsync(printRequest: EndOfDayOperationRequest<TResponse>):
Promise<ClientEntities.ICancelableDataResult<TResponse>> {
    this.context.logger.logInformational("Log message from PrintOperationRequestHandler
executeAsync().", this.context.logger.getNewCorrelationId());
    // Tender Removal
    let tenderRemovalRequest: TenderRemovalOperationRequest<TenderRemovalOperationResponse> =
    new TenderRemovalOperationRequest(this.context.logger.getNewCorrelationId());
    return this.context.runtime.executeAsync(tenderRemovalRequest).then((result:
ClientEntities.ICancelableDataResult<TenderRemovalOperationResponse>)
: Promise<Commerce.Client.Entities.ICancelableDataResult<SafeDropOperationResponse>> => {
        // Safe Drop
        if (!result.canceled) {
            let safeDropRequest: SafeDropOperationRequest<SafeDropOperationResponse> =
            new SafeDropOperationRequest(this.context.logger.getNewCorrelationId());
            return this.context.runtime.executeAsync(safeDropRequest);
        } else {
            return Promise.resolve({
                canceled: true,
                data: null
            });
        }
    }).then((result: Commerce.Client.Entities.ICancelableDataResult<SafeDropOperationResponse>)
: Promise<ClientEntities.ICancelableDataResult<TenderDeclarationOperationResponse>> => {
        // Tender Declaration
        if (!result.canceled) {
            let tenderDeclarationRequest:
TenderDeclarationOperationRequest<TenderDeclarationOperationResponse> =
            new TenderDeclarationOperationRequest(this.context.logger.getNewCorrelationId());
            return this.context.runtime.executeAsync(tenderDeclarationRequest);
        } else {
            return Promise.resolve({
                canceled: true,
                data: null
            });
        }
    }).then((result: ClientEntities.ICancelableDataResult<TenderDeclarationOperationResponse>)
: Promise<ClientEntities.ICancelableDataResult<CloseShiftOperationResponse>> => {
        // Close Shift
        if (!result.canceled) {
            return new Promise(
                (resolve: (value?:
ClientEntities.ICancelableDataResult<CloseShiftOperationResponse>) => void, reject: (reason?: any) =>
void) => {
                    // A delay of ten seconds is added here as a work-around for issues with
printing a second receipt to the windows driver
                    // printer before the first dialog is closed. A ten second delay gives the
user a chance to close the first dialog before
                    // the issue occurs.
                    setTimeout(() => { resolve(null); }, 10000);
                }).then(() => {
                    let closeShiftOperationRequest:
CloseShiftOperationRequest<CloseShiftOperationResponse> =
                    new CloseShiftOperationRequest(this.context.logger.getNewCorrelationId());
                    return this.context.runtime.executeAsync(closeShiftOperationRequest);
                });
            } else {

```

```

        } else {
            return Promise.resolve({
                canceled: true,
                data: null
            });
        }
    }).then((result: ClientEntities.ICancelableDataResult<CloseShiftOperationResponse>)
: ClientEntities.ICancelableDataResult<EndOfDayOperationResponse> => {
        return <ClientEntities.ICancelableDataResult<EndOfDayOperationResponse>>{
            canceled: result.canceled,
            data: result.canceled ? null : new EndOfDayOperationResponse()
        };
    });
}
}
}

```

### Create the operation factory class

1. In the **Operations** folder, create a typescript (.ts) file that is named **EndOfDayOperationRequestFactory.ts**.
2. Open the **EndOfDayOperationRequestFactory.ts** file, and add the following **import** statements to import the relevant entities and context.

```

import EndOfDayOperationResponse from "../EndOfDayOperationResponse";
import EndOfDayOperationRequest from "../EndOfDayOperationRequest";
import { ExtensionOperationRequestFactoryFunctionType, IOperationContext } from
"PosApi/Create/Operations";
import { ClientEntities } from "PosApi/Entities";

```

3. Add a function to link the operation handler and the operation button.

```

let getOperationRequest: ExtensionOperationRequestFactoryFunctionType<EndOfDayOperationResponse> =
/**
 * Gets an instance of EndOfDayOperationRequest.
 * @param {number} operationId The operation Id.
 * @param {string[]} actionParameters The action parameters.
 * @param {string} correlationId A telemetry correlation ID, used to group events logged from this
request together with the calling context.
 * @return {EndOfDayOperationRequest<TResponse>} Instance of EndOfDayOperationRequest.
 */
function (
    context: IOperationContext,
    operationId: number,
    actionParameters: string [],
    correlationId: string
): Promise<ClientEntities.ICancelableDataResult<EndOfDayOperationRequest<EndOfDayOperationResponse>>>
{
    let operationRequest: EndOfDayOperationRequest<EndOfDayOperationResponse> = new
EndOfDayOperationRequest<EndOfDayOperationResponse>(correlationId);
    return
Promise.resolve(<ClientEntities.ICancelableDataResult<EndOfDayOperationRequest<EndOfDayOperationRespo
nse>>>{
        canceled: false,
        data: operationRequest
    });
};
export default getOperationRequest;

```

The overall code should look like this.

```

import EndOfDayOperationResponse from "../EndOfDayOperationResponse";
import EndOfDayOperationRequest from "../EndOfDayOperationRequest";
import { ExtensionOperationRequestFactoryFunctionType, IOperationContext } from
"PosApi/Create/Operations";
import { ClientEntities } from "PosApi/Entities";
let getOperationRequest: ExtensionOperationRequestFactoryFunctionType<EndOfDayOperationResponse> =
/**
 * Gets an instance of EndOfDayOperationRequest.
 * @param {number} operationId The operation Id.
 * @param {string[]} actionParameters The action parameters.
 * @param {string} correlationId A telemetry correlation ID, used to group events logged from this
request together with the calling context.
 * @return {EndOfDayOperationRequest<TResponse>} Instance of EndOfDayOperationRequest.
 */
function (
  context: IOperationContext,
  operationId: number,
  actionParameters: string[],
  correlationId: string
): Promise<ClientEntities.ICancelableDataResult<EndOfDayOperationRequest<EndOfDayOperationResponse>>>
{
  let operationRequest: EndOfDayOperationRequest<EndOfDayOperationResponse> = new
EndOfDayOperationRequest<EndOfDayOperationResponse>(correlationId);
  return
Promise.resolve(<<ClientEntities.ICancelableDataResult<EndOfDayOperationRequest<EndOfDayOperationRespo
nse>>>{
    canceled: false,
    data: operationRequest
  });
};
export default getOperationRequest;

```

4. Open the **manifest.json** file, and paste in the following code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_EODSample",
  "publisher": "Microsoft",
  "version": "7.3.0",
  "minimumPosVersion": "7.3.0.0",
  "components": {
    "create": {
      "operations": [
        {
          "operationId": "5001",
          "operationRequestFactoryPath": "Operations/ EndOfDayOperationRequestFactory",
          "operationRequestHandlerPath": "Operations/ EndOfDayOperationRequestHandler"
        }
      ]
    }
  }
}

```

5. Open the **extensions.json** file under the **POS.Extensions** project, and update it with **EODSample**, so that the POS will include the extension at runtime.



```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions"
    },
    {
      "baseUrl": "EODSample"
    }
  ]
}

```

6. Open the `tsconfig.json` file, and comment out the extension package folders in the exclude list. The POS will use this file to include or exclude the extension. By default, the list contains the whole excluded extensions list. To include an extension as part of the POS, you must add the name of the extension folder and comment out the extension in the extension list, as shown here.

```

"extends": "../tsconfigs/tsmodulesconfig",
"exclude": [
  "AuditEventExtensionSample",
  "B2BSample",
  "CustomerSearchWithAttributesSample",
  "FiscalRegisterSample",
  // "EODSample",
  "PromotionsSample",
  "SalesTransactionSignatureSample",
  "SampleExtensions2",
  // "SampleExtensions",
  "StoreHoursSample",
  "SuspendTransactionReceiptSample"
],

```

7. Compile and rebuild the project.

## Add a custom operation button to the POS layout in Retail headquarters

1. In Retail go to **Retail and commerce > Channel setup > POS setup > POS > Operations**.
2. Create an operation that is named **EOD** and that has an operation ID of **5001**.
3. Go to **Retail and commerce > Channel setup > POS setup > POS > Button grids**.
4. Filter for **'F2W2'**.
5. Select the **Designer** button, and then follow the instructions to install the designer. If you're prompted for credentials, enter the Retail user name and password.
6. Right-click in the designer area, and then select **Add new row**.
7. Right-click the new button, and then select **Button properties**.
8. Set the **Action** property to **EOD**. Then select **OK**, and close the designer.
9. Go to **Retail and commerce > Retail IT > Distribution schedule**.
10. Select **1090**, and then select **Run now**.

**NOTE**

The preceding steps assume that you're using demo data. If you aren't using demo data, create and add the button according to your custom configurations.

## Validate your extension

1. Press F5, and deploy the POS to test your customization.
2. On the transaction screen, select the new EOD operation button, and follow the steps.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Point of sale (POS) APIs

2/18/2021 • 7 minutes to read • [Edit Online](#)

Retail POS APIs help you to easily build extensions or new features to the POS app. For example, if you are extending the Retail POS application to add new features in which to want to get product details, change prices, or add items to a cart. you can consume APIs that will do the work for you. To do this, you need to call the APIs to do the work. The POS API simplifies the extension pattern and provides continuous support to build the extensions.

Extension patterns have been unified across commerce runtime (CRT), POS, and Hardware station (HWS) by following the request/response pattern. All the POS APIs are exposed as request/response like CRT and HWS.

POS APIs are categorized into three different scenarios:

- **Consume** – Consume public APIs in your extension.
- **Extend** – Override public APIs to do some additional logic.
- **Create** – Create new APIs using the exposed POS interface, which can be used across extensions.

Many APIs can be consumed in extensions. For example, if you want to change the price of the item based on an external web service call, you can call `PriceOverrideOperationRequest` to change the price of the item. Within the consume, the APIs are sub categorized by module like cart, peripherals, store operations, etc.

## NOTE

A list of the all of the APIs is available in `Pos.Api.d.ts`, which is part of the Retail SDK (...Retail SDK\POS\Extensions\Pos.Api.d.ts). Extensions must consume only the exposed request and response from the `Pos.Api.d.ts` and no modification is allowed to `Pos.Api.d.ts`. Extensions should not consume or update any POS APIs, properties, methods, or handlers directly from POS commerce or sessions objects.

## How to consume APIs in your extension

Use the following steps to consume Retail APIs in your extensions.

1. Import the API in your extension file.

For example, if you want to consume the save attribute on cart API in your extension, then you need to add the following import statements.

The pattern is `import { api name } from "PosApi/Consume/Module name";`

```
import { SaveAttributesOnCartClientRequest, SaveAttributesOnCartClientResponse } from
"PosApi/Consume/Cart";
```

2. Import the client entities and proxy entities if necessary.

```
import { ClientEntities } from "PosApi/Entities";

import { ProxyEntities } from "PosApi/Entities";
```

3. Declare the API variable and execute it using the POS runtime, which you can access the runtime by

using: this.context.runtime.executeAsync("api name")

```
executeAsync<TResponse extends Response>(request: Request<TResponse>):  
Promise<Client.Entities.ICancelableDataResult<TResponse>>;
```

For example, if you want to execute the tender removal, use `SaveAttributesOnCartClientRequest` api, and refer to the following steps.

```
let attributeValue: ProxyEntities.AttributeTextValue = new ProxyEntities.AttributeTextValueClass();  
  
attributeValue.Name = PreEndTransactionTrigger.B2B_CART_ATTRIBUTE_NAME;  
  
attributeValue.TextValue = "Yes";  
  
let attributeValues: ProxyEntities.AttributeValueBase[] = [attributeValue];  
  
let saveAttributesOnCartRequest:  
SaveAttributesOnCartClientRequest<SaveAttributesOnCartClientResponse> =  
  
new SaveAttributesOnCartClientRequest(attributeValues);  
  
result = this.context.runtime.executeAsync(saveAttributesOnCartRequest);
```

## Samples showing how to access APIs

### Get Current cart

```
// Gets the current cart.  
  
let currentCart: ProxyEntities.Cart;  
  
return this.context.runtime.executeAsync<GetCurrentCartClientResponse>(new GetCurrentCartClientRequest())  
.then((getCurrentCartClientResponse: ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>):  
  
Promise<ClientEntities.ICancelableDataResult<GetCustomerClientResponse>> => {  
  
currentCart = getCurrentCartClientResponse.data.result;
```

### Get Current customer added to cart

```
// Gets the current customer.  
  
let result: Promise<ClientEntities.ICancelableDataResult<GetCustomerClientResponse>>;  
  
if (!ObjectExtensions.isNullOrUndefined(currentCart) &&  
!ObjectExtensions.isNullOrUndefined(currentCart.CustomerId)) {  
  
let getCurrentCustomerClientRequest: GetCustomerClientRequest<GetCustomerClientResponse> =  
  
new GetCustomerClientRequest(currentCart.CustomerId);  
  
result = this.context.runtime.executeAsync<GetCustomerClientResponse>(getCurrentCustomerClientRequest);  
  
} else {  
  
result = Promise.resolve({ canceled: true, data: new GetCustomerClientResponse(null) });  
  
}
```

## Force void transaction

```
// Force void transaction.
let forceVoidTransactionRequest: VoidTransactionOperationRequest<VoidTransactionOperationResponse> =

    new VoidTransactionOperationRequest<VoidTransactionOperationResponse>(false,
this.context.logger.getNewCorrelationId());

    this.context.runtime.executeAsync(forceVoidTransactionRequest).then((value:
ICancelableDataResult<VoidTransactionOperationResponse>) => {

    this.currentCart(JSON.stringify(value.data.cart));

    }).catch((err: any) => {

    this.currentCart(JSON.stringify(err));

    });
```

## Cart

The following table lists APIs exposed to perform cart-related functionality.

POS API	DESCRIPTION	RELEASE
AddPreprocessedTenderLineToCartClientRequest	Adds the pre-processed tender line to the cart.	10.0.14
AddTenderLineToCartClientRequest	Adds the tender line to the cart.	10.0.14
ConcludeTransactionClientRequest	Concludes the transaction.	10.0.14
GetCurrentCartClientRequest	Gets the current cart.	10.0.14
GetKeyedInPriceClientRequest	Gets the keyed in price.	10.0.14
GetPickupDateClientRequest	Gets the pickup date.	10.0.14
GetReasonCodeLinesClientRequest	Gets the reason code.	10.0.14
GetReceiptEmailAddressClientRequest	Gets the receipt email address.	10.0.14
GetShippingDateClientRequest	Gets the shipping date.	10.0.14
RefreshCartClientRequest	Refresh the current cart with the cart data from the server.	10.0.14
ResumeSuspendedCartClientRequest	Resumes the suspended transaction based on the ID passed.	10.0.14
SaveAttributesOnCartClientRequest	Saves the attributes on the cart.	10.0.14
SaveAttributesOnCartLinesClientRequest	Saves the attributes on the cart line.	10.0.14
SaveExtensionPropertiesOnCartClientRequest	Saves the extension properties on the cart.	10.0.14

POS API	DESCRIPTION	RELEASE
SaveExtensionPropertiesOnCartLinesClientRequest	Saves the extension properties on the cart line.	10.0.14
SaveReasonCodeLinesOnCartClientRequest	Saves the reason code lines on the cart.	10.0.14
SaveReasonCodeLinesOnCartLinesClientRequest	Saves the reason code lines on the cart line.	10.0.14
SelectSalesLinesForPickUpClientRequest	Select the sales lines for pickup.	10.0.14
SetCartAttributesClientRequest	Sets the cart attribute.	10.0.14
ShowChangeDueClientRequest	Shows the change due dialog.	10.0.14
AddAffiliationOperationRequest	Adds affiliation to the cart.	10.0.14
AddItemToCartOperationRequest	Add items to the cart.	10.0.14
CalculateTotalOperationRequest	Calculate the total for the cart.	10.0.14
ChangeCartLineUnitOfMeasureOperationRequest	Changes the cart line unit of measure.	10.0.14
CreateCustomerOrderOperationRequest	Creates the customer order.	10.0.14
CreateCustomerQuoteOperationRequest	Creates the customer quote.	10.0.14
CustomerAccountDepositOperationRequest		10.0.14
DepositOverrideOperationRequest	Overrides the deposit amount.	10.0.14
EditCustomerOrderOperationRequest	Edit the customer order.	10.0.14
LineDiscountAmountOperationRequest	Add line discount amount to the cart line.	10.0.14
LineDiscountPercentOperationRequest	Add line discount percent to the cart line.	10.0.14
OverrideLineTaxFromListOperationRequest	Override the cart line tax from the list.	10.0.14
OverrideLineTaxOperationRequest	Override the cart line tax.	10.0.14
OverrideTransactionTaxOperationRequest	Override the transaction tax.	10.0.14

POS API	DESCRIPTION	RELEASE
PickupAllOperationRequest	Picks up the order.	10.0.14
PriceOverrideOperationRequest	Override the price for the cart line.	10.0.14
SetCartLineCommentOperationRequest	Sets the cart line comment.	10.0.14
SetCartLineQuantityOperationRequest	Sets the cart line quantity.	10.0.14
SetCustomerOnCartOperationRequest	Sets the customer on the cart.	10.0.14
SetTransactionCommentOperationRequest	Sets the transaction comment.	10.0.14
SuspendCurrentCartOperationRequest	Suspends the current transaction.	10.0.14
TotalDiscountAmountOperationRequest	Add total discount amount to the transaction.	10.0.14
TotalDiscountPercentOperationRequest	Add total discount percent to the transaction.	10.0.14
VoidCartLineOperationRequest	voids the cart line.	10.0.14
VoidTenderLineOperationRequest	voids the tender line.	10.0.14
VoidTransactionOperationRequest	voids the transaction.	10.0.14
CreateEmptyCartServiceRequest	Creates empty cart.	10.0.14
GetTaxOverridesServiceRequest	Gets the tax override list.	10.0.14
UpdateTenderLineSignatureServiceRequest	Updates the tender line signature data.	10.0.14
CarryoutSelectedProductsOperationRequest	Marks the selected line as carry out.	10.0.14
AddCouponsOperationRequest	Add coupon to the transaction.	10.0.14
CreateNonSalesTransactionServiceRequest	Create non sales transaction cart.	10.0.14
ReturnTransactionOperationRequest	Returns the transaction.	10.0.14
AddLoyaltyCardToCartOperationRequest	Adds loyalty card to the transaction.	10.0.14
ReturnCartLineOperationRequest	Returns the cart line.	10.0.14
ReturnItemOperationRequest	Returns the item.	10.0.14

POS API	DESCRIPTION	RELEASE
AddExpenseAccountLineToCartOperationRequest	Add expense account line to the cart.	10.0.14
ShipAllCartLinesOperationRequest	Ships all the cart lines.	10.0.14
ShipSelectedCartLinesOperationRequest	Ships the selected cart line.	10.0.14
PickupSelectedOperationRequest	Marks the included lines for pickup	10.0.16

## Payments

The following table lists APIs exposed to perform payment-related functionality.

POS API	DESCRIPTION	RELEASE
GetGiftCardByIdServiceRequest	Gets the gift card ID.	10.0.12
GetPaymentCardTypeByBinRangeClientRequest	Get the card type bin range.	10.0.12
GetSignatureClientRequest	Shows the signature capture dialog in POS or sends the message to the signature capture device based on the configuration.	10.0.15

## Peripherals

The following table lists APIs exposed to perform peripheral-related functionality.

POS API
CardPaymentAuthorizePaymentRequest
CardPaymentBeginTransactionRequest
CardPaymentCapturePaymentRequest
CardPaymentEndTransactionRequest
CardPaymentEnquireGiftCardBalancePeripheralRequest
CardPaymentExecuteTaskRequest
CardPaymentRefundPaymentRequest
CardPaymentVoidPaymentRequest
CardPaymentAuthorizeCardTokenPeripheralRequest
CashDrawerIsOpenRequest
HardwareStationDeviceActionRequest



POS API
HardwareStationStatusRequest
LineDisplayDisplayLinesRequest
PaymentTerminalAuthorizePaymentActivityRequest
PaymentTerminalAuthorizePaymentRequest
PaymentTerminalBeginTransactionRequest
PaymentTerminalCancelOperationRequest
PaymentTerminalCapturePaymentRequest
PaymentTerminalEndTransactionRequest
PaymentTerminalEnquireGiftCardBalancePeripheralRequest
PaymentTerminalExecuteTaskRequest
PaymentTerminalRefundPaymentActivityRequest
PaymentTerminalRefundPaymentRequest
PaymentTerminalUpdateLinesRequest
PaymentTerminalVoidPaymentRequest
PaymentTerminalFetchTokenPeripheralRequest
PrinterPrintRequest
ScaleReadRequest

### ScanResults

The following table lists APIs exposed to perform scan results-related functionality.

POS API
GetScanResultClientRequest

### Customer

The following table lists APIs exposed to perform customer-related functionality.

POS API
GetCustomerClientRequest
CreateCustomerServiceRequest

POS API
UpdateCustomerServiceRequest
SelectCustomerClientRequest

### Authentication

The following table lists APIs exposed to perform authentication-related functionality.

POS API
LogOffOperationRequest
LockRegisterOperationRequest

### DataService

The following table lists APIs exposed to perform data service-related functionality.

POS API
DataServiceRequest

### Device

The following table lists APIs exposed to perform device-related functionality.

POS API
GetDeviceConfigurationClientRequest
GetExtensionProfileClientRequest
GetHardwareProfileClientRequest
GetAuthenticationTokenClientRequest
GetConnectionStatusClientRequest
GetActiveHardwareStationClientRequest
GetApplicationVersionClientRequest
GetChannelConfigurationClientRequest

### Diagnostics

The following table lists APIs exposed to perform diagnostics-related functionality.

POS API
GetSessionInfoClientRequest

### Dialog

The following table lists APIs exposed to perform dialog-related functionality.

POS API
ShowMessageDialogClientRequest
IAlphanumericInputDialogResult
ShowAlphanumericInputDialogClientRequest
ShowNumericInputDialogClientRequest
ShowListInputDialogClientRequest
ShowTextInputDialogClientRequest

## Employee

The following table lists APIs exposed to perform employee-related functionality.

POS API	DESCRIPTION	RELEASE
GetLoggedInEmployeeClientRequest	Gets the current logged in POS employee details.	10.0.14
SelectStoreEmployeeClientRequest	Gets the current store employee list for selection.	10.0.16

## Formatters

The following table lists APIs exposed to perform formatter-related functionality.

POS API
IBooleanFormatter
ICurrencyFormatter
IDateFormatter
ITransactionTypeFormatter
IPurchaseTransferOrderTypeFormatter

## OrgUnits

The following table lists APIs exposed to perform org units-related functionality.

POS API
GetOrgUnitConfigurationClientRequest
GetOrgUnitTenderTypesClientRequest
InventoryLookupOperationRequest

## Products

The following table lists APIs exposed to perform products-related functionality.

POS API
GetProductsByIdsClientRequest
GetCurrentProductCatalogStoreClientRequest
SelectProductVariantClientRequest
GetSerialNumberClientRequest
GetRefinerValuesByTextServiceRequest
SelectProductClientRequest
SelectProductVariantClientRequest
GetActivePricesServiceRequest

### Categories

The following table lists APIs exposed to perform categories-related functionality.

POS API
GetCategoriesServiceRequest

### SalesOrders

The following table lists APIs exposed to perform sales orders-related functionality.

POS API
GetReceiptsClientRequest
RegisterPrintReceiptCopyEventRequest
GetSalesOrderDetailsByTransactionIdClientRequest
GetGiftReceiptsClientRequest
RegisterPrintReceiptCopyEventRequest
MarkAsPickedServiceRequest
PrintPackingSlipClientRequest
PickUpCustomerOrderLinesClientRequest

### Shifts

The following table lists APIs exposed to perform shifts-related functionality.

POS API
CloseShiftOperationRequest
CloseShiftOperationRequest

### StockCountJournals

The following table lists APIs exposed to perform stock count journals-related functionality.

POS API
SyncAllStockCountJournalsClientRequest

### StoreOperations

The following table lists APIs exposed to perform store operations-related functionality.

POS API	DESCRIPTION	RELEASE
DeclareStartingAmountClientRequest	Declare start amount using this request.	10.0.14
GetSalesOrdersWithNoFiscalTransactionsRequest	Gets sales order with no fiscal transaction request.	10.0.14
RegisterCustomAuditEventClientRequest	Register custom audit event request.	10.0.14
GetOfflinePendingTransactionCountClientRequest	Gets the offline pending transaction count.	10.0.14
SaveFiscalTransactionClientRequest	Save fiscal transaction request.	10.0.14
SafeDropOperationRequest	Safe drop operation request.	10.0.14
TenderDeclarationOperationRequest	Tender declaration operation request.	10.0.14
TenderRemovalOperationRequest	Tender removal operation request.	10.0.14
CreateBankDropTransactionClientRequest	Bank drop transaction request.	10.0.14
CreateFloatEntryTransactionClientRequest	Float entry transaction request.	10.0.14
CreateStartingAmountTransactionClientRequest	Create start amount transaction request.	10.0.14
CreateTenderDeclarationTransactionClientRequest	Create tender declaration transaction request.	10.0.14
CreateTenderRemovalTransactionClientRequest	Remove tender declaration transaction request.	10.0.14

POS API	DESCRIPTION	RELEASE
GetDenominationTotalsClientRequest	Gets the denomination total request.	10.0.14
SelectZipCodeInfoClientRequest	Selects the Zip code information request.	10.0.14
CreateSafeDropTransactionClientRequest	Create safe drop transaction request.	10.0.14
GetTenderDetailsClientRequest	Gets the tender details.	10.0.14
LoyaltyCardPointsBalanceOperationRequest	Gets the loyalty card balance.	10.0.14
GetCommissionSalesGroupsServiceRequest	Gets the commission sales group.	10.0.14
GetCurrenciesServiceRequest	Gets the store currencies.	10.0.14
GetSrsReportDataSetServiceRequest	Gets the Srs report data.	10.0.14
SearchCommissionSalesGroupsServiceRequest	Search commission sales groups request.	10.0.14
IssueLoyaltyCardOperationRequest	Issues loyalty card.	10.0.14
GetPickingAndReceivingOrdersClientRequest	Gets the picking and receiving orders list.	10.0.14
BankDropOperationRequest	Bank drop request.	10.0.14
DeclareStartAmountOperationRequest	Declare start amount request.	10.0.14
GetAllDiscountsServiceRequest	Gets the discount applicable for the current cart.	10.0.16

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Point of sale (POS) payment extension

2/18/2021 • 7 minutes to read • [Edit Online](#)

With the extension points in point of sale (POS) to support payment extensibility, you can implement the core payment logic in the payment device or payment connector using the Hardware station APIs. Some scenarios where you might want to do this are:

- You need to pass additional information such as extension properties to your connector/device.
- You want to show custom messages in between the payment flow.
- You need input from the cashier to complete the flow and send an intermediate response back to the payment device/connector. For example, you might need the customer ID validation status or voice authorization.
- You want to show processing dialog messages, such as waiting for the customer pin input. You can override POS payment requests to implement these scenarios.

You can override the following request handlers from the POS side to customize the payment flow:

- `PaymentTerminalAuthorizePaymentRequestHandler`
- `PaymentTerminalCapturePaymentRequestHandler`
- `PaymentTerminalExecuteTaskRequestHandler`
- `PaymentTerminalRefundPaymentRequestHandler`
- `PaymentTerminalVoidPaymentRequestHandler`

The POS runtime checks the extension manifest to see if there are any extensions for these request handlers. If there are extensions, then the runtime loads the extended requests and executes the overridden requests. In the extension project, you can override these requests, add your own implementation to call the custom payment providers, and then update the response based on the status that is returned by your providers. When you override a request, you are overriding only the core logic. After all your custom logic has run, you send the updated response that you received from Hardware station (payment device/connector) to POS. All the standard workflow is handled by the POS, so that you do not need to worry about how to add, void, or decline the payment line and conclude the transaction based on the response.

## PaymentTerminalAuthorizePaymentRequestHandler

`PaymentTerminalAuthorizePaymentRequestHandler`, the authorization request, is the core payment request from POS that initiates and authorizes a card payment request. You can override this request if you want to change the authorize workflow. To override the request, you need to extend the `PaymentTerminalAuthorizePaymentRequestHandler` in POS.

```

import { PaymentTerminalAuthorizePaymentRequestHandler } from
"PosApi/Extend/RequestHandlers/PeripheralsRequestHandlers";
import { PaymentTerminalAuthorizePaymentRequest, PaymentTerminalAuthorizePaymentResponse } from
"PosApi/Consume/Peripherals";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { GetCurrentCartClientRequest, GetCurrentCartClientResponse } from "PosApi/Consume/Cart";
import { PaymentHandlerHelper } from "../PaymentHandlerHelper";
import { ObjectExtensions } from "PosApi/TypeExtensions";

/**
 * Override request handler class for the payment terminal authorize payment request.
 */
export default class PaymentTerminalAuthorizePaymentRequestHandlerExt extends
PaymentTerminalAuthorizePaymentRequestHandler {

    /**
     * Executes the request handler asynchronously.
     * @param {PaymentTerminalAuthorizePaymentRequest<PaymentTerminalAuthorizePaymentResponse>} request The
request.
     * @return {Promise<ICancelableDataResult<PaymentTerminalAuthorizePaymentResponse>>} The cancelable
promise containing
     * the response.
     */
    public executeAsync(request:
PaymentTerminalAuthorizePaymentRequest<PaymentTerminalAuthorizePaymentResponse>):
    Promise<ClientEntities.ICancelableDataResult<PaymentTerminalAuthorizePaymentResponse>> {
        let cart: ProxyEntities.Cart = null;
        let cartRequest: GetCurrentCartClientRequest<GetCurrentCartClientResponse> = new
GetCurrentCartClientRequest();

        // Get cart first and then build extension properties based on cart info.
        return this.context.runtime.executeAsync(cartRequest)
            .then((result: ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>): void => {
                if (!(result.canceled || ObjectExtensions.isNullOrUndefined(result.data))) {
                    cart = result.data.result;
                }
            }).then(() :
Promise<ClientEntities.ICancelableDataResult<PaymentTerminalAuthorizePaymentResponse>> => {
                let newRequest:
PaymentTerminalAuthorizePaymentRequest<PaymentTerminalAuthorizePaymentResponse> =
                    new PaymentTerminalAuthorizePaymentRequest<PaymentTerminalAuthorizePaymentResponse>(
                        request.paymentConnectorId,
                        request.amount,
                        request.tenderInfo,
                        request.voiceAuthorization,
                        request.isManualEntry,
                        PaymentHandlerHelper.FillExtensionProperties(cart,
request.extensionTransactionProperties));

                return this.defaultExecuteAsync(newRequest);
            });
    }
}

```

You need to make these changes in PaymentHandlerHelper.ts.



```

import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { ObjectExtensions, StringExtensions } from "PosApi/TypeExtensions";

/**
 * Override request handler class for the payment terminal authorize payment request.
 */
export class PaymentHandlerHelper {
  // Get extra properties for payment terminal
  public static FillExtensionProperties(
    cart: ProxyEntities.Cart,
    extensionProperties: ClientEntities.IExtensionTransaction): ClientEntities.IExtensionTransaction {

    let extraProperties: ClientEntities.IExtensionTransaction = null;
    // Build extra extension properties.
    if (!ObjectExtensions.isNullOrUndefined(cart)) {
      extraProperties = {
        ExtensionProperties: [
          <ProxyEntities.CommerceProperty>{
            Key: "CartId",
            Value: <ProxyEntities.CommercePropertyValue>{
              StringValue: !ObjectExtensions.isNullOrUndefined(cart) ? cart.Id : ""
            }
          }, {
            Key: "ChannelId",
            Value: <ProxyEntities.CommercePropertyValue>{
              StringValue: (ObjectExtensions.isNullOrUndefined(cart.ChannelId)) ? ""
                : cart.ChannelId.toString()
            }
          }, {
            Key: "TerminalId",
            Value: <ProxyEntities.CommercePropertyValue>{
              StringValue: cart.TerminalId
            }
          }, {
            Key: "StaffId",
            Value: <ProxyEntities.CommercePropertyValue>{ StringValue: cart.StaffId }
          }, {
            Key: "CustomerId",
            Value: <ProxyEntities.CommercePropertyValue>{
              StringValue: (StringExtensions.isNullOrWhitespace(cart.CustomerId)) ? ""
                : cart.CustomerId
            }
          }, {
            Key: "ShippingZipCode",
            Value: <ProxyEntities.CommercePropertyValue>{
              StringValue: !ObjectExtensions.isNullOrUndefined(cart.ShippingAddress) ?
                (StringExtensions.isNullOrWhitespace(cart.ShippingAddress.ZipCode) ? "" :
                  cart.ShippingAddress.ZipCode) : ""
            }
          }
        ]
      };
    }

    if (ObjectExtensions.isNullOrUndefined(extensionProperties)) {
      extensionProperties = extraProperties;
    } else {
      for (let i: number = 0; i < extraProperties.ExtensionProperties.length; i++) {
        extensionProperties.ExtensionProperties.push(extraProperties.ExtensionProperties[i]);
      }
    }

    return extensionProperties;
  }
}

```

After implementing the request logic, you need to update manifest.json with the extension information so that POS loads the extension.

```
{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Payment_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "extend": {
      "requestHandlers": [
        {
          "modulePath": "Peripherals/Handlers/PaymentTerminalAuthorizePaymentRequestHandlerExt"
        }
      ]
    }
  }
}
```

The full code sample, including how to pass extension properties, is available in Retail SDK app update 3 in the RetailSDK\Code\POS\Extensions\PaymentSample folder. If you check in the above code sample, only the calling portion has been overridden, and not the core logic on how to complete the payment or add payment line. The POS workflow manages that.

## PaymentTerminalCapturePaymentRequestHandler

**PaymentTerminalCapturePaymentRequestHandler**, the payment request, is a payment request from POS that initiates and captures the card payment request. Override this request if you want to change the capture workflow. To override the request, you need to extend the **PaymentTerminalCapturePaymentRequestHandler** in POS.

```

import { PaymentTerminalCapturePaymentRequestHandler } from
"PosApi/Extend/RequestHandlers/PeripheralsRequestHandlers";
import { PaymentTerminalCapturePaymentRequest, PaymentTerminalCapturePaymentResponse } from
"PosApi/Consume/Peripherals";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { GetCurrentCartClientRequest, GetCurrentCartClientResponse } from "PosApi/Consume/Cart";
import { PaymentHandlerHelper } from "./PaymentHandlerHelper";
import { ObjectExtensions } from "PosApi/TypeExtensions";

/**
 * Override request handler class for the payment terminal Capture payment request.
 */
export default class PaymentTerminalCapturePaymentRequestHandlerExt extends
PaymentTerminalCapturePaymentRequestHandler {
    /**
     * Executes the request handler asynchronously.
     * @param {PaymentTerminalCapturePaymentRequest<PaymentTerminalCapturePaymentResponse>} request The
request.
     * @return {Promise<ICancelableDataResult<PaymentTerminalCapturePaymentResponse>>}
     * The cancelable promise containing the response.
     */
    public executeAsync(request:
PaymentTerminalCapturePaymentRequest<PaymentTerminalCapturePaymentResponse>):
        Promise<ClientEntities.ICancelableDataResult<PaymentTerminalCapturePaymentResponse>> {
        let cart: ProxyEntities.Cart = null;
        let cartRequest: GetCurrentCartClientRequest<GetCurrentCartClientResponse> = new
GetCurrentCartClientRequest();

        // Get cart first and then build extension properties based on cart info.
        return this.context.runtime.executeAsync(cartRequest)
            .then((result: ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>): void => {
                if (!(result.canceled || ObjectExtensions.isNullOrUndefined(result.data))) {
                    cart = result.data.result;
                }
            }).then(() : Promise<ClientEntities.ICancelableDataResult<PaymentTerminalCapturePaymentResponse>>
=> {
                let newRequest: PaymentTerminalCapturePaymentRequest<PaymentTerminalCapturePaymentResponse>
=
                    new PaymentTerminalCapturePaymentRequest<PaymentTerminalCapturePaymentResponse>(
                        request.amount,
                        request.paymentProperties,
                        PaymentHandlerHelper.FillExtensionProperties(cart,
request.extensionTransactionProperties));

                return this.defaultExecuteAsync(newRequest);
            });
    }
}

```

After implementing the request logic, you need to update the manifest.json with the extension information so that POS loads the extension. Any requests that you override are specified in the manifest. If you didn't override any of the standard requests, then you do not need to specify anything in the manifest. The example of the manifest shows two overridden requests.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Payment_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "extend": {
      "requestHandlers": [
        {
          "modulePath": "Peripherals/Handlers/PaymentTerminalAuthorizePaymentRequestHandlerExt"
        },
        {
          "modulePath": "Peripherals/Handlers/PaymentTerminalCapturePaymentRequestHandlerExt"
        }
      ]
    }
  }
}

```

The full code sample, with how to pass extension properties, is available in Retail SDK app update in the RetailSDK\Code\POS\Extensions\PaymentSample folder.

## PaymentTerminalExecuteTaskRequestHandler

**PaymentTerminalExecuteTaskRequestHandler**, the execution request, is used from POS to initiate any custom payment device/connector operation from POS. You might use this to do a health check of the payment device from POS, to do batch processing, or for an end-of-day request to the payment device. You can override this request if you want to do a custom operation other than the standard authorize, capture, void, and refund. To override the request, you need to extend the **PaymentTerminalExecuteTaskRequestHandler** in POS.

```

import { PaymentTerminalExecuteTaskRequestHandler } from
"PosApi/Extend/RequestHandlers/PeripheralsRequestHandlers";
import { PaymentTerminalExecuteTaskRequest, PaymentTerminalExecuteTaskResponse } from
"PosApi/Consume/Peripherals";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { GetCurrentCartClientRequest, GetCurrentCartClientResponse } from "PosApi/Consume/Cart";
import { PaymentHandlerHelper } from "./PaymentHandlerHelper";
import { ObjectExtensions } from "PosApi/TypeExtensions";

/**
 * Override request handler class for the payment terminal ExecuteTask request.
 */
export default class PaymentTerminalExecuteTaskRequestHandlerExt extends
PaymentTerminalExecuteTaskRequestHandler {
    /**
     * Executes the request handler asynchronously.
     * @param {PaymentTerminalExecuteTaskRequest<PaymentTerminalExecuteTaskResponse>} request The request.
     * @return {Promise<ICancelableDataResult<PaymentTerminalExecuteTaskResponse>>}
     * The cancelable promise containing the response.
     */
    public executeAsync(request: PaymentTerminalExecuteTaskRequest<PaymentTerminalExecuteTaskResponse>):
    Promise<ClientEntities.ICancelableDataResult<PaymentTerminalExecuteTaskResponse>> {
        let cart: ProxyEntities.Cart = null;
        let cartRequest: GetCurrentCartClientRequest<GetCurrentCartClientResponse> = new
GetCurrentCartClientRequest();

        // Get cart first and then build extension properties based on cart info.
        return this.context.runtime.executeAsync(cartRequest)
            .then((result: ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>): void => {
                if (!(result.canceled || ObjectExtensions.isNullOrUndefined(result.data))) {
                    cart = result.data.result;
                }
            }).then(() : Promise<ClientEntities.ICancelableDataResult<PaymentTerminalExecuteTaskResponse>> =>
{
            let newRequest: PaymentTerminalExecuteTaskRequest<PaymentTerminalExecuteTaskResponse> =
                new PaymentTerminalExecuteTaskRequest<PaymentTerminalExecuteTaskResponse>(
                    request.task,
                    PaymentHandlerHelper.FillExtensionProperties(cart,
request.extensionTransactionProperties));

            return this.defaultExecuteAsync(newRequest);
        });
    }
}

```

After implementing the request logic, you need to update the manifest.json with the extension information so that POS loads the extension.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Payment_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "extend": {
      "requestHandlers": [
        {
          "modulePath": "Peripherals/Handlers/PaymentTerminalAuthorizePaymentRequestHandlerExt"
        },
        {
          "modulePath": "Peripherals/Handlers/PaymentTerminalCapturePaymentRequestHandlerExt"
        },
        {
          "modulePath": "Peripherals/Handlers/PaymentTerminalExecuteTaskRequestHandlerExt"
        }
      ]
    }
  }
}

```

The full code sample, with how to pass extension properties, is available in Retail SDK app update 3 in the RetailSDK\Code\POS\Extensions\PaymentSample folder

## PaymentTerminalRefundPaymentRequestHandler

**PaymentTerminalRefundPaymentRequestHandler**, the refund request, is a payment request from POS that initiates a refund or return of the card payment. You override this request if you want to change the refund workflow. To override the request, you need to extend the **PaymentTerminalRefundPaymentRequestHandler** in POS.

## PaymentTerminalVoidPaymentRequestHandler

**PaymentTerminalVoidPaymentRequestHandler**, the void request, is a payment request from POS that initiates the void card payment request. You override this request if you want to change the void workflow. To override the request, you need to extend the **PaymentTerminalVoidPaymentRequestHandler** in POS.

Extending the void and refund request code pattern is same as the authorize and capture request. The full code sample for the void and refund payment request, with how to pass extension properties, is available in Retail SDK app update 3 in the RetailSDK\Code\POS\Extensions\PaymentSample folder.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Support for tipping in the payments SDK

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how the payments software development kit (SDK) for Microsoft Dynamics 365 Commerce supports tip amounts that are entered through a payment terminal. The purpose of this feature is to add first-class support to the payments SDK by including a separate field for the tip amount in authorization responses.

This feature doesn't add support for tipping or tip reporting at the point of sale (POS). Implementation of full tipping capabilities requires POS extensions.

## Key terms

TERM	DESCRIPTION
Tips	Tips, which are also known as gratuities, are common in the quick service and hospitality industries. They enable a payment to be given directly to the store or restaurant employee who provides services.
Header-level charge	A charge that can be applied to a purchase, but that isn't for a specific line item.

## Overview of the tipping feature

Tipping is common in some locales and industries. For example, in the quick service and hospitality industries, tipping is now done almost everywhere in the United States. For many businesses, the ability to support tipping through payment terminals is becoming an important factor that helps attract employees. This feature adds a separate **TipAmount** field to the payments SDK, so that tip amounts that are selected on the payment terminal can be sent back to the POS as part of the authorization response.

Although this feature adds support for tipping at the level of the payments SDK, it doesn't include support for other important aspects of tipping functionality. For example, the feature doesn't support reporting of tip payouts at the end of shifts, the ability to pool tips, or the ability to report tips for payroll. To enable full tipping support, you must implement those capabilities through extensions.

For more information about how to create the payment terminal integrations and SDK references that are mentioned in this topic, see [Create an end-to-end payment integration for a payment terminal](#).

### Prerequisites

PREREQUISITE	DESCRIPTION
Tip support on devices	Customers must be able to select a tip amount on the payment terminal.
<b>isTippingEnabled</b> support	Payment connectors must support the <b>isTippingEnabled</b> variable for payment initialization.
<b>TipAmount</b> field	The tip amount must be returned from the payment connector in the <b>TipAmount</b> field of the authorization response.

PREREQUISITE	DESCRIPTION
POS tip support through extension	The tip amount that is returned from the payment connector should be added to the sale as a header-level charge. Tip reporting and management aren't provided out of the box.

### Tippping support by payment processors

This feature adds a new `isTippingEnabled` variable to the `AuthorizePaymentTerminalDeviceRequest` request. When a payment is requested from the POS, this variable indicates whether the authorization request is eligible to have a tip added. Payment connectors that receive this request can then request a tip-eligible authorization from the connected payment service or terminal.

When the `isTippingEnabled` variable is set to `True`, if a customer selects a tip amount on the payment terminal, that amount should be returned in the `TipAmount` field of the `AuthorizePaymentCardPaymentResponse` response that is returned from the payment connector. The tip amount is also included in the `ApprovedAmount` field of the authorization response.

### Tippping support for the Adyen connector

The Adyen connector includes tip support. Customization is required to set the `isTippingEnabled` variable to `True` when the authorization response is passed to the connector. When the `isTippingEnabled` variable is set to `True`, if a customer selects a tip amount on the device, that amount should be returned in the `TipAmount` field of the authorization response.

Before a customer can be prompted to select a tip amount on the Adyen terminal, the terminal must be configured for tipping. This configuration is done through the Adyen customer area. To enable tipping, select the **Point of Sale** tab in the Adyen portal. Tipping can be enabled by terminal or through the fleet-wide **Terminal settings** option. In both the settings for individual terminals and the fleet-wide terminal settings, tipping is enabled on the **Payment features** tab. After tipping is enabled on your Adyen device, settings on the device must be updated. The change won't take effect until this update is done.

### Suggested implementation

We recommend that a header-level charge be used to add the tip amount to the transaction after the authorization response is returned from the payment connector. To support this functionality, an extension should be created for the `PaymentTerminalAuthorizePaymentRequestHandler` handler. That extension should include logic to add a header-level charge to the transaction if a `TipAmount` value is returned in the authorization response.

## Additional resources

- [Adyen connector overview](#)
- [Create an end-to-end payment integration for a payment terminal](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Support for external gift cards

2/18/2021 • 16 minutes to read • [Edit Online](#)

This topic explains how to set up external gift cards in Retail Modern point of sale (MPOS), the call center, and the storefront.

Microsoft Dynamics 365 Commerce supports both *internal* and *external* gift cards. Internal gift cards are managed entirely in Dynamics 365 Commerce, whereas external gift cards are administered by a third party. If a retailer's operations are run entirely in Microsoft Dynamics, internal gift cards are sometimes the best solution. For complex enterprises that span multiple countries or regions, and multiple point of sale (POS) systems, it's often best to use a third party to manage gift card balances and enable gift cards to be used across those systems.

Like support for other card payment types, support for external gift cards must be built into the payment connector that is used. The out-of-box payment connector for Adyen supports external gift cards through SVS and Givex in POS, the call center, and the e-commerce storefront.

## External gift card setup

### NOTE

Some setup steps assume that demo data is used. The steps might vary, depending on the dataset that is used. The test connector is for sandbox purposes only. The test connector is not supported for use in UAT or production environments.

### Card types

1. Search for **Card Types**.
2. Select **New**, add the following values, and then select **Save**.

FIELD NAME	VALUE
Card ID	EXTGC
Card type name	External Gift Card
Card types	Gift card
Card issuer	Enter any description.

### Card numbers

1. On the **Card types** page, select the newly created gift card, and then select **Card numbers**.
2. Specify the range of card numbers that should be used for external gift cards, and then select **Save**.

In the following example, if the first four digits of a card number are **6036**, the card will be mapped to the gift card that you set up in the "Card types" section of this topic.

FIELD NAME	VALUE
Card number from	6000

FIELD NAME	VALUE
Card number to	6999
Digits to identify	4

## Payment methods

1. Search for **Payment methods** to open the **Payment methods** page.
2. Select **New**, and then follow these steps:
  - a. In the **Payment method** field, enter **12**.
  - b. In the **Payment method name** field, enter **External Gift Card**.
  - c. In the **Default function** field, select **Card**.
  - d. Select **Save**.

## Store setup

1. Search for **All stores** to open the **All stores** page.
2. Select the **San Francisco** store in the list.
3. On the Action Pane, on the **Set up** tab, in the **Set up** group, select **Payment methods**.
4. Select **New**.
5. In the **Payment method** field, enter **12**. The **Payment method name** and **Function** fields should then be set automatically.
6. On the **General** FastTab, set the following fields:
  - Set the **Operation name** field to **Pay gift card**.
  - Set the **Connector name** field to **TestConnector**.
7. On the **Posting** FastTab, set the **Gift card item number** field to **0010**.

The screenshot shows the 'Payment methods' configuration page in Microsoft Dynamics 365. The left sidebar lists various payment methods, with '12 External Gift Card Card' selected. The main area displays the 'Posting' tab with the following fields and values:

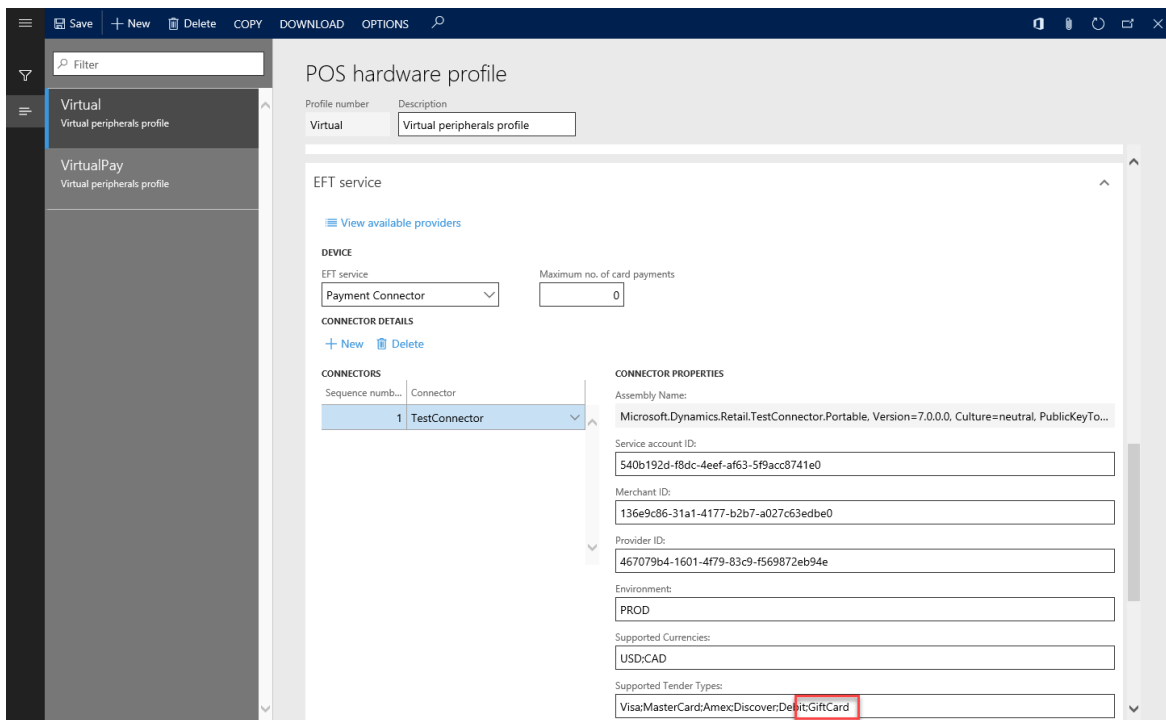
- Payment method:** 12
- Payment method name:** External Gift Card
- Function:** Card
- Posting:**
  - ACCOUNT:**
    - Account type: Ledger account
    - Account number: [empty]
  - GIFT CARD ACCOUNT:**
    - Company: USRT
    - Account type: Ledger account
    - Account number: [empty]
    - Gift card item number:** 0010 (highlighted with a red box)
  - DIFFERENCE ACCOUNT:**
    - Difference account: [empty]
    - Account name: [empty]
  - Maximum difference amount:** 0.00
  - Big difference account:** [empty]
  - Account name:** [empty]
  - TENDER DECLARATION OPTIONS:**
    - Compress payment entries: No
    - Counting required: No
    - Multiply in tender operations: No
    - Allow float: No
    - Taken to bank: No
    - Taken to safe: No
  - BANK TRANSACTION:**
    - Use bank account: No
    - Account type: Ledger account
    - Account number: [empty]
    - Bank transaction type: [empty]
  - SAFE TRANSACTION:**
    - Use safe account: No
    - Account type: Ledger account
    - Safe account: [empty]

8. Select **Save**.

9. Select **Card setup**, and then select **New** to map the gift card payment method to the newly created external gift card payment method for the San Francisco store.

## POS setup

1. In Dynamics 365 Commerce Headquarters, search for **Hardware profiles** to open the **POS hardware profile** page.
2. In the left pane, select **Virtual**.
3. Select **Edit**.
4. On the **EFT service** FastTab, in the **Connectors** grid, select the first entry, **TestConnector**.
5. In the **Supported Tender Types** field, add **GiftCard**.



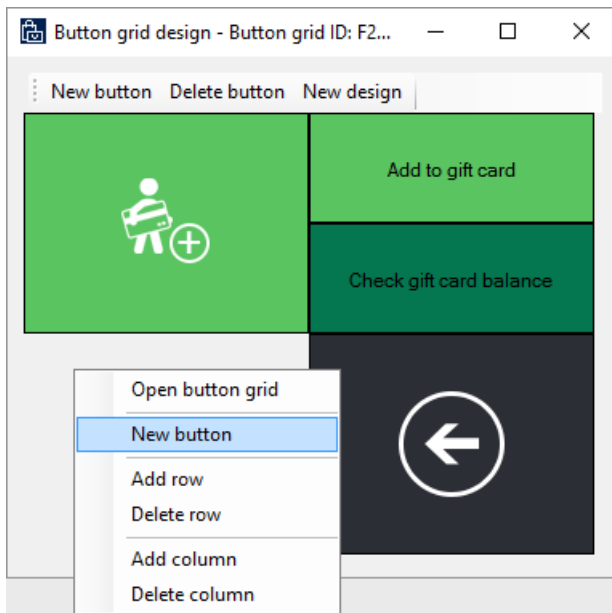
6. Select **Save**.

### NOTE

You can also use the **New** button to create multiple payment connectors. In this way, you can take advantage of the support for multiple connectors that has been added to the solution. You can then have different payment connectors for different payment methods. For example, all credit cards can be processed through one connector, but gift cards can be processed through a different connector.

## Update the button grid

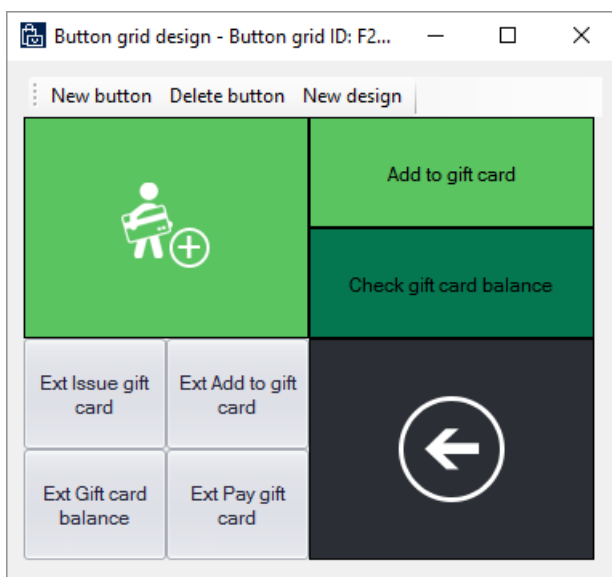
1. Go to the **Button grid** page.
2. In the navigation bar on the left side of the page, search for **F2S1M**, and select the filtered option.
3. On the Action Pane, select **Designer** to download the button designer application.
4. When the grid designer appears, right-click on an empty (gray) area, and then select **New button**.



5. Right-click the new button, and then select **Button properties**.
6. Set the **Action**, **Payment type**, and **Text on button** properties according to the following matrix.

ACTION	PAYMENT TYPE	TEXT ON BUTTON
Issue gift card	External Gift Card	Ext Issue gift card
Add to gift card	External Gift Card	Ext Add to gift card
Gift card balance	External Gift Card	Ext Gift card balance
Pay gift card	External Gift Card	Ext Pay gift card

When you've finished, your button layout should resemble the following illustration.



7. Close the designer.
8. Search for **Distribution Schedule**.
9. In the navigation bar on the left side of the page, search for 1090, 1115, and 1070.
10. On the Action Pane, select **Run now**.

11. Check the status of the job by searching for **Download sessions**.
12. Wait until **Applied** appears next to all the jobs, and then close the browser.

#### NOTE

If you are using Retail Commerce Scale Unit (RCSU) that is located in the store, you need to perform an IIS reset to clear the cache. You can either do this through the IIS application or open an admin Command Prompt window and enter `iisreset`. Otherwise, wait for the RCSU to be updated.

## Configure and test Modern POS

1. Start the Modern POS (MPOS) application.
2. Sign in by using the standard credentials.
3. When you're prompted, select **Perform a non-drawer operation**.
4. On the main screen, select **Select hardware station**.
5. On the bar on the right side of the page, select **Manage**.
6. Turn on **Virtual Peripherals**, and then select **OK**.
7. In the **Available paired stations** field, select **Virtual Peripherals**.
8. You're prompted to either open a new shift or perform non-drawer operations. You can now open a new shift.
9. On the main screen, select **Current transaction**.
10. Select **Gift cards**.
11. Select **Ext Issue gift card**.
12. Enter a number that starts with **9**, and then provide an amount.
13. After items are added to the cart, you can pay by using cash or a card.

When you use the test connector to demonstrate support for external gift cards, you should use card number **61234** to make payments. You won't be prompted for a personal identification number (PIN). The test connector should **never** be used in production.

## External gift cards for the call center and storefront

#### NOTE

External gift card support for call center and storefront is enabled in the **Feature management** workspace. Enable **Omni-channel payments**, then enable **Enable advanced external gift card**. For additional steps required to set up external gift cards in the storefront, please visit the docs article dedicated to [E-commerce digital gift cards](#).

### Tokenization

The out-of-box implementation and Payments software development kit (SDK) support for external gift cards in the call center and storefront requires tokenization. When external gift cards are processed, tokens are used to refer to the actual gift card number. This is important for 3rd party implementations because without tokens, external gift card processing may not function correctly. For example, if a gift card payment is captured when it's added to an order, but an issue occurs during order creation, the gift card payment will be reversed using references (tokens) to the transaction itself, not using the actual gift card number.

### Purchases and refunds

When an external gift card is used for a purchase, the tender line for the gift card is saved as a prepayment. Therefore, the funds for the purchase are captured when the order is created.

External gift cards aren't eligible for refunds. In part, this limitation is in place to prevent a refund from being

given for a gift card that the user has discarded. If an unprocessed order includes an external gift card as payment, and the customer wants to cancel the order, a new gift card or some other form of credit must be issued to the customer.

Gift cards lines that are issued as part of an order can be canceled before fulfillment.

### Issuing gift cards through fulfillment

Physical gift cards and virtual gift cards have distinct fulfillment methods.

Physical gift cards are gift cards that are mapped to a mode of delivery of the **Shipping** type. They must be issued directly through the gift card provider as part of order processing. The gift card number must then be mapped to the order line as part of the pick list registration process. Next, the masked gift card number is saved back to the order line. The gift card that is issued is then activated as part of order invoicing.

Virtual gift cards are issued as part of order invoicing. When a gift card line is marked as **Packed**, it becomes eligible to be issued. Virtual gift cards are issued as part of invoicing. When invoicing occurs, the gift card number is obtained from the provider through the payment connector. The number for the activated gift card is then sent to the gift card recipient via email. When invoicing occurs, the masked gift card number is then saved back to the order line.

### Using modes of delivery for gift card products in the call center and e-commerce

In the call center and storefront channels, unlike in POS, a dedication operation isn't used to issue gift cards. Instead, gift cards are issued by adding a line item to a transaction. Specifically, gift card products for e-commerce and the call center can be either mapped to product variants or modeled as standard products.

If product variants are used, the person who creates the gift card order is prompted to select the variant. The relevant mode of delivery will then be available for that product variant.

Modes of delivery must support the type of gift card. For example, a gift card product variant of the **Physical** style must be mapped to a mode of delivery that is related to shipping. A gift card product variant of the **Email** style must be mapped to an electronic mode of delivery. The electronic mode of delivery is defined on the **Customer orders** tab of the **Commerce parameters** page.

#### NOTE

Only virtual gift cards are currently supported in e-commerce. For more details about setting up virtual gift cards in e-commerce, see [E-commerce digital gift cards](#).

## Setup for the call center and storefront

### Payment services setup

In the back office, on the **Payment services** page, configure the payment services account for the call center. Each payment connector requires different setup steps. The payment service that the call center uses is marked as **Default**.

## Call center setup

1. Search for **All call centers** to open the call centers page.
2. Select the **Fashion call center** store in the list.
3. On the Action Pane, on the **Set up** tab, in the **Set up** group, select **Payment methods**.
4. Select **New**.
5. In the **Payment method** field, enter 12. The **Payment method name** and **Function** fields will then be

set automatically.

6. On the **General** FastTab, set the following fields:
  - In the **Operation name** field, select **Pay gift card**.
  - In the **Connector name** field, select **TestConnector**.
7. On the **Posting** FastTab, set the **Gift card item number** field to **0010**.
8. Select **Save**.
9. Select **Card setup**, and then select **New** to map the gift card payment method to the newly created external gift card payment method for the **Fashion call center**.

## Online store setup

1. Search for **Online stores** to open the online stores page.
2. Select the **Fabrikam extended online** store in the list.
3. On the Action Pane, on the **Set up** tab, in the **Set up** group, select **Payment methods**.
4. Select **New**.
5. In the **Payment method** field, enter **12**. The **Payment method name** and **Function** fields will then be set automatically.
6. On the **General** FastTab, set the following fields:
  - In the **Operation name** field, select **Pay gift card**.
  - In the **Connector name** field, select **TestConnector**.
7. On the **Posting** FastTab, set the **Gift card item number** field to **0010**.
8. Select **Save**.
9. Select **Card setup**, and then select **New** to map the gift card payment method to the newly created external gift card payment method for the **Fabrikam extended online store**.

## Online store payments setup

To configure the payment accounts for your online store to use Adyen for external gift card processing, please refer to the [e-Commerce setup section](#) of the documentation for the Adyen connector.

### Adyen external gift card setup

For an example that shows how to set up payment services, see the [documentation for the Adyen payment connector](#).

For the call center and storefront, the Adyen connector supports the following gift cards.

BRAND	GIFT CARD TYPE	SUPPORTED	ACTIVATION
SVS	Physical	Yes	Manually
SVS	Email	Yes	Programmatically
Givex	Physical	Yes	Manually

## NOTE

In the out-of-box Adyen connector, gift cards are not configured by default. To specify the gift card provider in the merchant properties of the payment connector, follow the instructions in the [documentation for the Adyen payment connector](#).

### Test connector external gift card setup

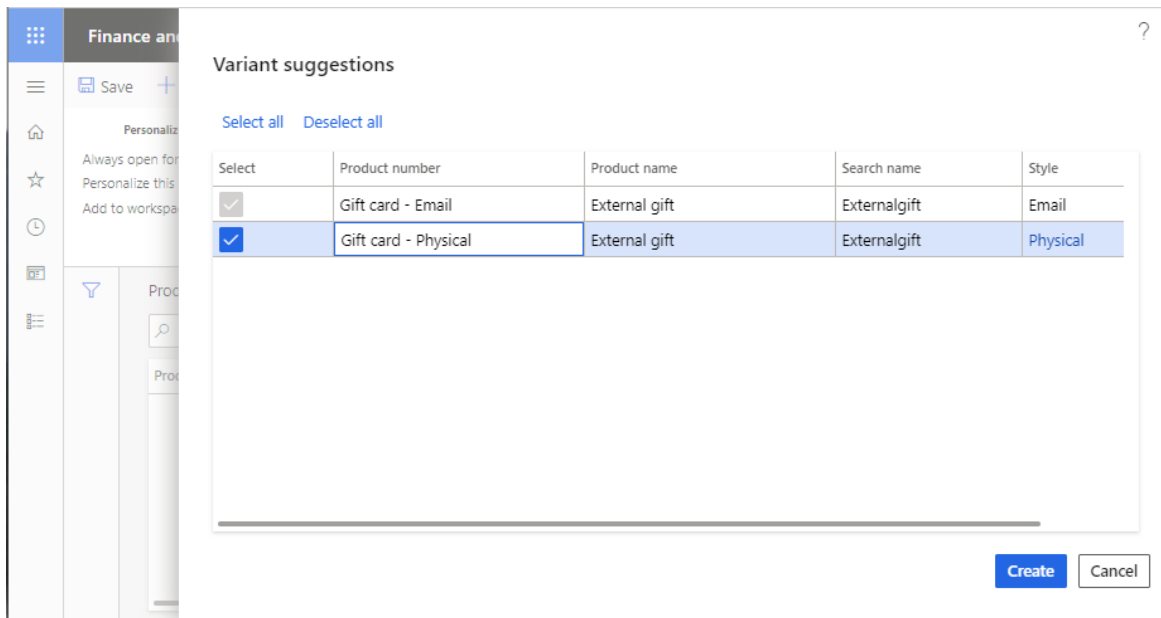
To set up external gift cards for the test connector, on the **Payment services** page, select **Dyn Online**, and then, in the **Supported Tender Types** field, add **;GiftCard** after **Debit**. Then select **Credit card types**, and assign a payment journal to the gift card payment method.

### Gift card product setup

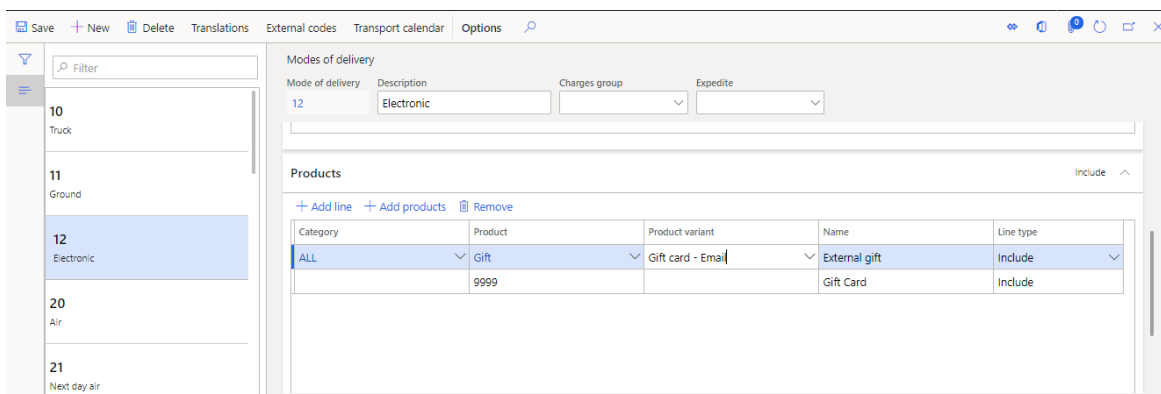
The following procedure shows how to set up an external gift card by using product masters. Product masters aren't required for external gift cards. However, they can be helpful when both physical and virtual gift cards are used.

1. Search for **Style groups** to open the **Style groups** page.
2. Select **New**.
3. Enter a name (for example, **Gift**) and a description (for example, **Gift card style group**).
4. Add styles to suit the types of gift cards that are available. For example, add **Physical** and **Email** styles.
5. Save your changes.
6. Search for **Product masters** to open the **Product masters** page.
7. Select **New**.
8. Enter a product number (for example, **Gift**), and assign a retail category.
9. In the **Product dimension group** field, select **Style**.
10. Select **OK**.
11. In the product master, select the style group that you created earlier (**Gift**).
12. On the Action Pane, on the **Product** tab, in the **Set up** group, select **Dimension groups**, and then assign a storage dimension group and a tracking dimension group.
13. Select **Save**.
14. Select **Product variants**, select **Variant suggestions**, and edit the gift card variant numbers as you require.
15. Select **Create**.





16. Select **Release products**, select **Next** two times, select a company (for example, **USRT**), and then select **Next**. Finally select **Next** to release the product master.
17. Search for **Modes of delivery** to open the **Modes of delivery** page.
18. Select the **Electronic** mode of delivery, and add the **Email** gift card variant. Make sure that applicable call centers and online channels are included.



19. Select a shipping mode of delivery, and add the **Physical** gift card variant.
20. Select **Save**.
21. Search for **Process delivery modes** to open the **Process delivery modes** dialog box.
22. Select **OK**.

**NOTE**  
Gift cards aren't currently supported for MPOS customer order creation or for in-store pickup.

23. Search for **Released products by category** to open the **Released product details** page.
24. Select the external gift card item.
25. Set the following values.

FASTTAB	FIELD	VALUE
General	Item model group	MA_Retail
Purchase	Purchase order unit	ea
Sell	Sales order unit	ea
Sell	Allow price adjust	Yes
Manage inventory	Inventory unit	ea
Manage costs	Posting item group	Any

26. Select **Save**.

For the storefront, the gift card must also be included in the storefront's assortment. For more information, see [Assortment management](#).

#### NOTE

The gift card product used for external gift card setup in POS should not use product masters with item variants. Gift cards based on item variants may still be used for payments, balance inquiries, and cash out in POS, but gift card products associated with the POS for issuance must be standard products.

#### Set up notification emails for virtual gift cards

For information about email setup, see [Configure email functionality](#).

For information about how to set up email notifications for Commerce, see [Set up an email notification profile](#).

For gift cards that are issued via email, the value of the **Retail email notification type** field is **Issue gift card**.

#### Call center setup

1. Search for **All call centers** to open the **Call center** page.
2. In the list, select a call center.
3. On the Action Pane, on the **Channel** tab, in the **Users** group, select **Channel users**.
4. Add a user, select **Save**, and then close the **Channel users** page.

#### IMPORTANT

Users must be call center users when they access the **Customer service** page and create orders. Otherwise, call center capabilities won't be available.

5. Back on the **Call center** page, on the Action Pane, on the **Set up** tab, in the **Set up** group, select **Payment methods**.
6. Select **New**.
7. In the **Payment method** field, enter **12**. The **Payment method name** and **Function** fields should then be set automatically.
8. On the **General** FastTab, select the connector that should be used for the external gift card.

9. Select **Card setup**, and then select **New** to map the gift card payment method to the external gift card payment method.
10. On the **Posting** FastTab, specify general ledger accounts for the external gift cards. In demo data, **112140** can be used as the general ledger account number.
11. Select **Save**.
12. Select **Card setup**.
13. Select **New**, select the external gift card that you created earlier, and then select **Save**.
14. Close the **Card setup** page, and refresh the **Payment methods** page.
15. Select the external gift card payment method, and then, on the **General** FastTab, in the **Gift card account** section, in the **Connector name** field, assign the **TestConnector** connector.
16. On the **Posting** FastTab, assign the gift card item number.
17. Select **Save**.
18. Select **Card setup**.
19. The **Card setup** page for the internal gift card now includes two additional configuration fields: **Check expiration date** and **PIN required**. Set these fields as required by the external gift card provider.

#### **Issue external gift cards in the call center**

1. As a call center user, search for **Customer service** to open the **Customer service** page.
2. Add a customer by using the **Search** function.
3. Select **New sales order**.
4. Select **Header**, and add a valid mode of delivery.
5. Select **Lines**.
6. In the **Item number** field, specify the gift card item number.
7. In the **Variant number** field, specify the variant number if you're using a product master.
8. In the **Unit price** field, specify the unit price for the gift card.

#### **NOTE**

If modes of delivery are mapped to variants, and if **Electronic** modes of delivery are specified for any gift cards, the gift card type should automatically be set to **Email** on the **Packing** tab.

9. On the **Line details** FastTab, on the **Packing** tab, follow one of these steps:
  - For virtual gift cards, set the **Buyer name**, **Buyer email**, **Recipient name**, **Recipient email**, and **Gift message** fields.
  - For physical gift cards, set the **Buyer name**, **Recipient name**, and **Gift message** fields.
10. On the **Price and discount** tab, in the **Reason code** field, specify the reason for the price override.
11. Select **Complete**, add a payment, and submit the order.

#### **Pay by using external gift cards in the call center**

1. As a call center user, create an order, and select **Complete**.
2. On the **Payments** FastTab, select **Add**.

3. Select the external gift card payment method, and enter the number and PIN, if applicable. For the test connector, 61234 can be used as the number, and the PIN isn't validated.
4. Use a percentage amount or a payment amount to define the payment amount.

The screenshot shows the Dynamics 365 Finance and Operations interface. On the left, a sidebar contains navigation options like 'New', 'Service order', 'Purchase order', and 'Direct delivery'. The main area displays a 'Sales order summary' with the following details:

- Order: Order (dropdown) / Hold (dropdown)
- Sales status: Open order
- Payment status: Not submitted
- Currency: USD
- TOTALS:
  - Sales total: 118.24
  - Payment total: 118.24
  - Balance: 0.00
- Payment method: 12
- Payment method name: External gift card

On the right, a dialog box titled 'Enter customer payment information' is open. It contains the following fields and options:

- Payment method: 12
- Payment method name: EXTERNAL (dropdown)
- Payment method name: External gift card
- Number: 61234
- Pin required: 222
- Check expiration date: [button]
- Available balance: 1,000.00 [refresh button]
- Percent amount: 100.00
- Payment amount: 118.24
- Prepay:  Yes

At the bottom right of the dialog box are 'OK' and 'Cancel' buttons.

5. Select OK.
6. Select Submit to complete the order.

## Troubleshooting

### Issue: An error occurs when you start the HardwareStationConfigurationUtility program

1. From an elevated command prompt, open the HardwareStationConfigurationUtility.exe.config file in Notepad.
2. In the file, follow these steps:
  - a. Replace the DataServiceUrl value with the correct Commerce Scale Unit URL.
  - b. Verify that the AADLogonUrl value is correct.
3. Save and close the file.
4. Restart the utility.

### Issue: A token error occurs when you try to pair virtual peripherals

1. Exit MPOS.
2. Go to C:\Program Files (x86)\Microsoft Dynamics 365\70\Retail Hardware Station\Package.
3. From an elevated command prompt, open the Web.config file in Notepad.
4. Replace the RetailServer value with the correct Commerce Scale Unit value.
5. Save and close the file.
6. Restart MPOS.

7. If the issue persists, exit MPOS, use Task Manager to end any instances of dllhost.exe that are running, and then do another reset of Internet Information Services (IIS).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create and apply branding to the Retail Experience app

2/18/2021 • 10 minutes to read • [Edit Online](#)

You can apply your branding to the Retail Experience app, and release it to Google Play and the Apple App Store. This topic explains how to build the app, connect, and apply your branding.

## Development tools

The Retail Experience app supports the Android and iOS phone platforms. The app is built by using Xamarin.Forms, and you must install Xamarin on your development computer. To build the iOS app, you must have a Mac that has Xamarin installed. Although you can do development for both Android and iOS on a computer that runs Microsoft Windows, you must use a Mac to complete the build for the iOS platform. If your Mac is a shared team resource, you might want to use a Mac just for the build process. You must install the Retail software development kit (SDK) on all the computers you use for development.

### Install Xamarin

You can download Xamarin from [Visual Studio Tools for Xamarin](#).

For a tutorial that shows how to install Xamarin on Windows, see [Installing Xamarin](#).

### Update Xamarin

After you've installed Xamarin, you must update it to the latest stable version.

- **Windows:** In Microsoft Visual Studio, click **Tools > Options > Environment > Xamarin > Other**.
- **Mac:** In Xamarin Studio, click **Check for Updates > Update channel**. For more information, see [Change the Updates Channel](#).

### Connecting to a Mac

If you're developing on Windows and using the Mac just for building the iOS app then you must connect the computer that runs Windows and the Mac. For instructions, see [Connecting to the Mac](#).

## Connect to an online channel

Open the Retail Experience app solution in Visual Studio for the next steps. The Retail Experience app uses an online channel to show the products. You can use any online channel. Depending on your requirements, you can use a different online channel for each app, or you can use the same online channel for both apps. Any released product that is assorted to the online channel will appear in the app.

### NOTE

The app can't be used to issue gift cards. Therefore, gift cards must be excluded from the assortment for the online channel that the apps use. Information about the Commerce Scale Unit endpoint and the online channel is added to the config.xml file that is present in each app project. You must make the following changes in the config.xml file:

- In the `<DataServiceUrl>` tag, add the URL of the Commerce Scale Unit.
- In the `<MediaBaseUrl>` tag, add the URL of the media server. Make sure that this URL has a slash (/) at the end.
- In the `<OperatingUnitNumber>` tag, add the operating unit number of the online channel that you want

to use together with the app.

#### NOTE

Enter the complete operating unit number. For example, in the demo data, the operating unit number for Contoso online store is 068.

## Connect to a payment connector

To enable payments to be accepted from the app, you must set up a payment connector on the online channel that is used for the app. The payment connector loads the card-related controls on a host HTML page. We have added a host page that is named `CardPaymentHost.html` to the Retail SDK. We have also verified that this host page works with the MasterCard connector. However, you might have to edit this page to meet the requirements of the other payment providers. Different connectors might provide pages. The host page can be hosted on any website that you choose. However, we recommend that you host it on your eCommerce site, if you have an eCommerce site. After the page has been hosted, add the URL of the host page in the `<CardPaymentHostPageUrl>` tag.

## Update the About page

Add the following links to the about page:

- In the `<TermsOfUseUrl>` tag, add a link to the license terms.
- In the `<PrivacyPolicyUrl>` tag, add a link to the privacy policy.
- In the `<ThirdPartyNoticesUrl>` tag, add a link to the third-party notices.

#### NOTE

If these links are missing, and the user clicks them, the app might stop responding. Therefore, be sure to add the links to these tags. The About page has an `<EvaluationModeEnabled>` tag that is used to enable Evaluation Mode view. Evaluation Mode lets retailers easily change the Commerce Scale Unit endpoint without having to change any code. This feature helps retailers evaluate the app for various channels. **Evaluation Mode must be disabled in the production version of the app.** By default, Evaluation Mode is disabled in the Retail SDK.

## Apply branding

After you've successfully built the Retail Experience app by using the Microsoft sample branding, you can change icons, labels, and colors to make the app meet your own branding requirements. The following list shows some of the icons, labels, and colors that retailers might want to change. This list isn't exhaustive.

### Icons

- **App** – The icon that users see after the app is installed on their phones.
- **Retailer's brand** – The icon that appears on the menu flyout on Android.
- **Shop** – The icon that appears on the menu flyout on Android, and on the tab bar on Apple iPhone.
- **Information** – The icon that appears on the menu flyout on Android, and on the tab bar on iPhone.
- **Shopping Cart** – The icon that appears in the upper-right corner on Android, and on the tab bar on iPhone.
- **Search** – The icon that appears in the upper-right corner on Android and iPhone.
- **Splash screen** – The icon that appears as the loading image on Android and iPhone.

### Labels/texts

- App name (the name that appears below the app icon when the app is installed on the phone).

- Shop icon name and the corresponding page title.
- Information icon name and the corresponding page title.
- Confirmation messages that appear as toast notifications.
- "Add to cart" button text and the corresponding page title.
- "Checkout" button text and the corresponding page title.
- "Place order" button text.
- About page label and the corresponding page title.

## Colors

- App bar (The app bar is also known as the action bar or toolbar.)
- Status bar color
- Accent color
- Primary action button colors. For example, the **Add to Cart**, **Checkout**, **Place Order**, and **Continue** buttons all have the same color.

## Applying branding to the Android app

### Icons

To change the icons for the Android app, open the Resource folder, and update the icons in each drawable folder. These drawable folders contain images for Android at different resolutions. You can either preserve the sizes of the images in these folders or change them to meet your requirements. For information about how Android handles different screen sizes, see [Screen compatibility overview](#). The splash screen is put in the drawable folder and is a special type of image that Android requires. For more information about the splash screen, see [Create resizable bitmaps](#).

### Label/text

To change the text labels on the Android app, open the TextResources.resx file, and edit the values of the available resources. The following table shows the name of the label/text that appears on the Android device and the corresponding property/key that must be changed in TextResources.resx. The app name is specified in the strings.xml file.

LABEL/TEXT	PROPERTY NAME/KEY
App name	<b>App_name</b> . This property can be found in the strings.xml file at ../ShoppingAppDroidResources/values.
Shop icon name and the corresponding page title	<b>MenuPage_Products</b> and <b>Pages_CategoryListPage_Products</b> , respectively
Information icon name and the corresponding page title	<b>MainTabs_Account</b> and <b>Pages_AccountPage_AccountInformation</b> , respectively
Confirmation messages that appear as toast notifications	Multiple messages in TextResources.resx
"Add to cart" button text and the corresponding page title	<b>Pages_ProductDetail_AddToCart</b> and <b>Pages_CartPage_Title</b> , respectively
"Checkout" button text and the corresponding page title	<b>Pages_CartPage_Checkout</b> and <b>Pages_CheckoutPage_Title</b> , respectively
"Place order" button text	<b>Pages_CheckoutPage_PlaceOrder</b>
About page label and the corresponding page title	<b>Pages_AccountPage_About</b> and <b>Pages_AboutPage_Title</b> , respectively



#### NOTE

Some of the resources in the TextResources.resx file aren't used by the Android app, but they are used by the iOS app. Examples include **MainTabs\_Cart** and **MainTabs\_Products**. You can change those resources from the Android app without affecting the iOS app.

#### Colors

To change the colors that appear in the Android app, open the config.xml file and colors.xml files (colors.xml is under Resources->values folder). Now change the values for **Primary** (the app bar color), **PrimaryDark** (the status bar color), **Accent**, and **ActionButtonBackground** in both the files and keep them consistent.

#### NOTE

The product price is shown on two pages namely products list page and the product details page. The price shown on the product list page is set to green (#008A00) and cannot be altered. However, the price shown on the product details page is governed by the value set for the "PrimaryDark" property.

### Applying branding to the iOS app

#### Icons

To change the icons on the iOS app, open the Resource folder, and update the icons. You can either preserve the sizes of the images in the folders or change them to meet your requirements.

#### Label/text

To change the text for the iOS app, open the TextResources.resx file, and edit the values of the available resources. The following table shows the name of the label/text that appears on the iOS device and the corresponding property/key that must be changed in TextResources.resx. The app name is specified in the info.plist file.

LABEL/TEXT	PROPERTY NAME/KEY
App name	<b>CFBundleDisplayName</b> . This property can be found in the info.plist file in the app's iOS project.
Shop icon name and the corresponding page title	<b>MainTabs_Products</b> and <b>Pages_CategoryListPage_Products</b> , respectively
Information icon name and the corresponding page title	<b>MainTabs_Account</b> and <b>Pages_AccountPage_AccountInformation</b> , respectively
Confirmation messages that appear as toast notifications	Multiple messages in TextResources.resx
"Add to cart" button text and the corresponding page title	<b>Pages_ProductDetail_AddToCart</b> and <b>Pages_CartPage_Title</b> , respectively
"Checkout" button text and the corresponding page title	<b>Pages_CartPage_Checkout</b> and <b>Pages_CheckoutPage_Title</b> , respectively
"Place order" button text	<b>Pages_CheckoutPage_PlaceOrder</b>
About page label and the corresponding page title	<b>Pages_AccountPage_About</b> and <b>Pages_AboutPage_Title</b> , respectively

#### NOTE

Some of the resources in the info.plist file aren't used by the iOS app, but they are used by the Android app. An example is `MenuPage_Products`. You can change those resources from the iOS app without affecting the Android app.

#### Colors

To change the colors that appear in the iOS app, open the config.xml file, and change the values for **Primary** (the app bar color), **PrimaryDark** (the status bar color), **Accent**, and **ActionButtonBackground**.

#### NOTE

The product price is shown on two pages namely products list page and the product details page. The price shown on the product list page is set to green (#008A00) and cannot be altered. However, the price shown on the product details page is governed by the value set for the "PrimaryDark" property.

## Build, test, and distribute apps

The next step is to build the app and test on devices. Testing will be followed by publishing the app to the public app stores. We have added the standard external links from Xamarin and Apple which explain the basics of Android and iOS app distribution.

### Android

To test the Android app on an emulator follow the below steps on Visual Studio

1. Select Droid as startup project, Marshmallow as Android device.
2. Build Release | Any CPU.
3. Run in Marshmallow emulator.

To learn how to use the Archive manager on Visual Studio to **create an .apk file** that can be signed by a signing certificate, see [Preparing an Application for Release](#).

For details about how to create a signing certificate and **sign the .apk file** for ad hoc and Google Play store distribution, see [Signing the Android Application Package](#).

For more information about the steps involved with the **public distribution of an application** created with Xamarin.Android via channels such as email, a private web server, Google Play, or the Amazon App Store for Android, see [Publishing an Application](#).

### iOS

To test the iOS app on an emulator, use the following steps in Visual Studio.

1. Select iOS as startup project.
2. Connect to Mac as explained above.
3. Build Release | iPhone.
4. Run the app in Visual Studio.

For detailed information about the distribution techniques that are available for Xamarin.iOS applications, see [Xamarin.iOS App Distribution overview](#).

To learn how to maintain your signing identities and certificates, refer to [What is app signing](#).

**NOTE**

If you are using the certificates as a part of Apple Developer Enterprise Program and you install an app manually (such as for testing), then you must also manually establish trust. For more information, see [Install custom enterprise apps on iOS](#).

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Set up POS hybrid app on Android and iOS

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic shows how to build and run the Retail POS hybrid app on Android and iOS devices.

## Overview

Retail hybrid app is shell built using [Xamarin](#). Inside the shell is a Web view controller that loads the cloud POS, which is based on the Commerce Scale Unit URL specified in the settings of this app. This is a Retail hybrid app shell for Android and iOS which will internally load the Cloud POS. For more information, see [Cloud POS](#).

## Development tools

The Retail hybrid app supports the Android and iOS phone platforms. The app is built by using Xamarin, which means that you must install Xamarin on your development computer. To build the iOS app, you must have a Mac that has Xamarin installed. Although you can do development for both Android and iOS on a computer that runs Microsoft Windows, you must use a Mac to complete the build for the iOS platform. If your Mac is a shared team resource, you might want to use a Mac just for the build process. You must copy the Retail software development kit (Retail SDK) on all the computers that you use for development. The Retail SDK is available in all developer VMs that are provisioned for using [Microsoft Dynamics Lifecycle Services \(LCS\)](#).

For more information about Xamarin, see the [Xamarin documentation](#).

## Set up and install Xamarin on Windows

To set up and install Xamarin on Windows, go to </xamarin/android/get-started/installation/windows>.

### Update Xamarin

#### NOTE

We recommend that you use Xamarin.Android SDK version < 10.0.

After you've installed Xamarin, you must update it to the latest stable version (Xamarin.Android SDK version must be < 10.0).

- **Windows** - In Microsoft Visual Studio, click **Tools > Options > Environment > Xamarin > Other**.
- **Mac** - In Xamarin Studio, click **Check for Updates > Update channel**. For more information about this step, see [Change the Updates Channel](#).

### Build the Android Retail hybrid app

1. When installation is complete, launch Visual Studio and sign in with your Microsoft account (this is the same account that you use with Windows). Check for Xamarin updates by clicking **Tools > Options > Xamarin** or **Tools > Options > Xamarin > Other**. Here you'll find a **Check Now** link. If you do not see an option for Xamarin in **Tools > Options**, review your installation, or try restarting Visual Studio. You can also search for Xamarin in the **Options** dialog box. If needed, download and install the latest version.
2. In the Retail SDK folder, open `SampleExtensions\HybridApp\Android\solution`. Build and deploy using the emulator and verify that everything appears as it should.

- Using the [Visual Studio Emulator for Android](#) or any emulator for Android, launch the POS hybrid app and enter the Commerce Scale Unit URL and save.
- You should be able to sign in and activate the device.

### Build the iOS Retail hybrid app

#### Connecting to a Mac

If you're developing on Windows and using the Mac just for building the iOS app then you must connect the computer that runs Windows and the Mac. For instructions, see [Connecting to the Mac](#).

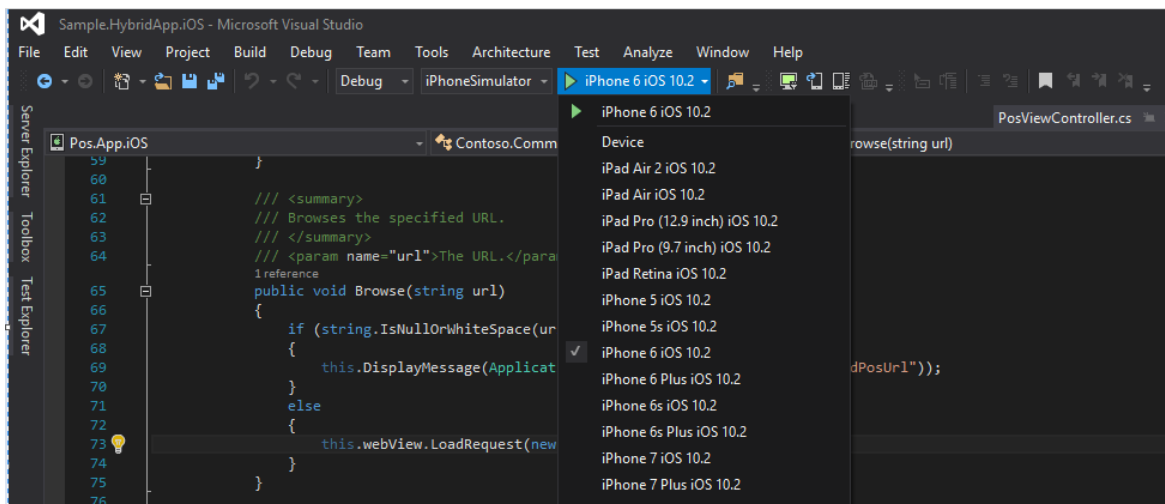
## Set up and install Xamarin on iOS

For more detailed steps on installing Xamarin on iOS, refer to [Xamarin.iOS installation](#).

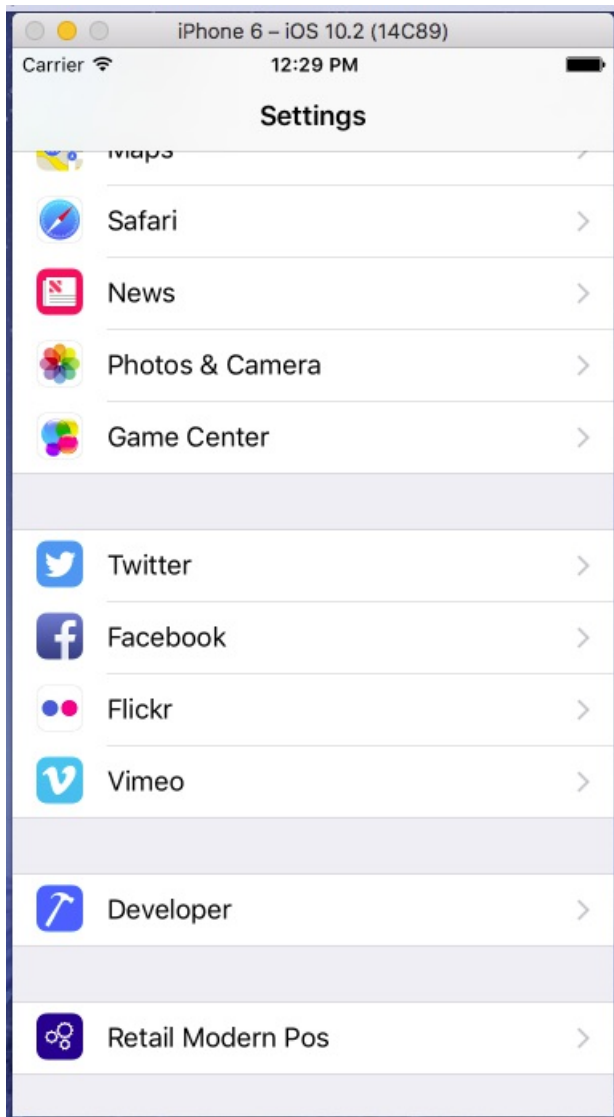
- Download and install Xcode from <https://developer.apple.com/xcode/>. Add your Apple ID using the instructions described in [Adding your account to Xcode](#) (apple.com).
- Download and install Xamarin by following the instructions in [Installing and configuring Xamarin.iOS](#) (xamarin.com).
- When you have completed installing Xamarin on both the Windows and Mac computers, follow the instructions in [Connecting to the Mac](#) (xamarin.com). After you do this, you can work with iOS and Mac from Visual Studio on the Windows computer.

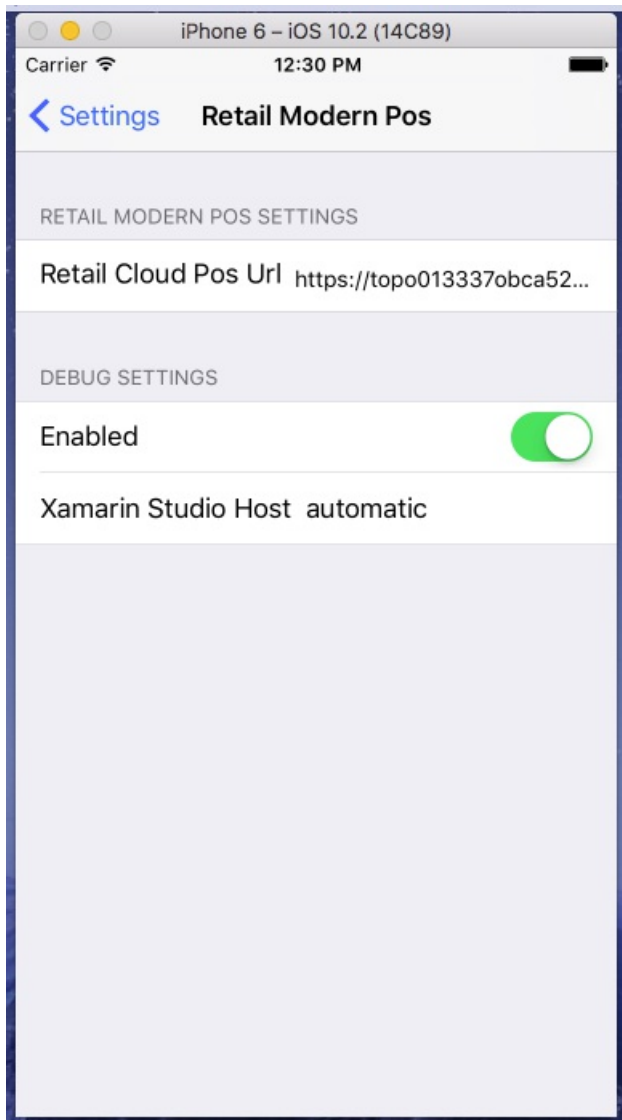
### Build the iOS Retail hybrid app

- In the Retail SDK folder, open SampleExtensions\HybridApp\iOS\solution. After connecting to the Mac and building the application in Visual Studio, select the iOS device type and deploy the app on the selected device.



- Using the Emulator, go to **Settings > RetailIMPOS**. Enter the Commerce Scale Unit URL.





3. Launch the MPOS app. You should be able to sign in and activate the device.

## Dedicated hardware station support for the hybrid Android app

Starting in release 8.1.3, dedicated hardware station support has been added to the hybrid Android app. In the same way that the Retail Modern POS has built-in support for peripheral devices, the Android app can also use the dedicated hardware station to connect to peripherals without needing to deploy an IIS-based hardware station. Out of the box, the hybrid Android app supports using payment terminals and receipt printers over network connections. Communicating with devices over a network typically requires adherence to a proprietary communication protocol specified by the manufacturer. For the hybrid Android app, out-of-the-box integrations are provided for the Dynamics 365 payment connector for Adyen and Epson receipt printers.

### Out-of-the-box supported devices

DEVICE	DESCRIPTION
Payment terminals	Any supported by the <a href="#">Adyen Payment Terminal API</a> through the Dynamics 365 Payment Connector for Adyen.
Receipt printer	Network-enabled Epson printers that support the Epson SOAP HTTP interface. Network-enabled Star Micronics printers.

DEVICE	DESCRIPTION
Cash drawer	Introduced in Dynamics 365 Commerce version 10.0.8: Cash drawers that are connected to network-enabled printers via the drawer kick (d/k) port.

Support for other payment processors and peripheral devices can be implemented by ISVs through the Payments and Hardware SDKs.

### Set up peripherals to work with the hybrid Android app

To enable direct hardware support for the hybrid Android app, set up a dedicated hardware station in the same way it would be set up for MPOS. Instructions for setting up the dedicated, or IPC, hardware station can be found in [Retail peripherals](#)

#### NOTE

The dedicated hardware station provided with demo data should not be used with the hybrid Android app. To test the hybrid Android app in an environment with demo data, delete the existing hardware stations and create a new dedicated hardware station.

To do this, go to **Retail and Commerce > Channels > Stores > All stores**. Select the store that will be used, typically "HOUSTON".

In the store details form, scroll down to the **Hardware stations** FastTab. Remove the existing dedicated hardware station, then select **Add** to add a new hardware station of type **Dedicated**. A description is optional. No other details are necessary for the hardware station.

To set up the payment connector, follow the standard setup steps noted in the [Dynamics 365 Payment Connector for Adyen](#). Skip the section labeled "Update the Modern POS or IIS Hardware Station configuration."

For details on setting up network connected peripherals the docs [Support for network peripherals](#).

## Additional resources

- [Payments FAQ](#)

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Override POS request handler

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains how to override POS request handler. We've introduced an extension pattern for overriding the POS business logic. If you have a scenario where you want to modify/add some business logic to the core POS business flow, then you can follow this pattern.

For example, when you sell a serial item, POS will display a dialog box where you can enter the serial number for that item after the scan. If you want to automate the serial number process by entering the serial number through code, then you can override this serial number request handler and use custom business logic. Most of the business logic in POS is implemented in request handler, however, you can override the relevant request handler and return the response according to your business flow.

## NOTE

Not all request handler logic is exposed for customization. If you want to customize any business logic and if that request handler is not overridable, then create a support ticket or log a request in the LCS extensibility tool.

## POS request handler logic exposed for overriding

This list is based on [Microsoft Dynamics 365 for Finance and Operations - Version 7.3.5](#).

In each monthly update we will be adding additional extension points, so check the Pos.api.d.ts file in the Retail SDK for the full list.

## Cart extension handlers

REQUEST NAME	DESCRIPTION
AddTenderLineToCartClientRequestHandler	This handler is executed when you add tender (payment) line to cart.
GetKeyedInPriceClientRequestHandler	This handler is executed when you add an item that has a configuration key in price during sale.
GetPickupDateClientRequestHandler	Executed when you select a pickup date during a customer order.
GetShippingDateClientRequestHandler	Executed when you select a shipping date during a customer order.
ShowChangeDueClientRequestHandler	Executed when the change due dialog box is shown at the end of transaction.
GetReceiptEmailAddressClientRequestHandler	Executed when you get a receipt email address.
DepositOverrideOperationRequestHandler	Executed when you override a deposit.
GetShippingChargeClientRequestHandler	Executed when get shipping charge workflow initiated during customer order flow.

REQUEST NAME	DESCRIPTION
GetKeyedInPriceClientRequestHandler	Executed when the key in product price dialog box is shown.

### Payment extension handler

REQUEST NAME	DESCRIPTION
GetGiftCardByIdServiceRequestHandler	This handler is executed when you receive the gift card ID.
GetPaymentCardTypeByBinRangeClientRequestHandler	This handler is executed when POS gets the card type, such as Visa or Master Card. This is based on the HQ configuration during the card tender line processing.

### Peripherals request handler

REQUEST NAME	DESCRIPTION
CardPaymentAuthorizePaymentRequestHandler	Executed when a card payment is authorized.
CardPaymentCapturePaymentRequestHandler	Executed when a card payment is captured.
CardPaymentExecuteTaskRequestHandler	Used to execute any custom task. This handler is mainly for extensions for custom functionality with payment connector, which is not supported.
CardPaymentRefundPaymentRequestHandler	Executed when a card payment is refunded.
CardPaymentVoidPaymentRequestHandler	Executed when a card payment is voided.
CardPaymentBeginTransactionRequestHandler	Executed when a card payment is initiated.
CardPaymentEndTransactionRequestHandler	Executed when a card payment is ended.
CardPaymentEnquireGiftCardBalancePeripheralRequestHandler	Executed when a gift card balance inquiry is made.
PaymentTerminalAuthorizePaymentActivityRequestHandler	Executed when a card payment is authorized using a payment. terminal/device.
PaymentTerminalAuthorizePaymentRequestHandler	Executed when a card payment is authorized using a payment. terminal/device.
PaymentTerminalEnquireGiftCardBalancePeripheralRequestHandler	Executed when a gift card balance inquiry is made using a payment. terminal/device.
PaymentTerminalExecuteTaskRequestHandler	Used to execute any custom task. This handler is mainly for extensions for custom functionality with payment terminal/device, which is not supported.
PaymentTerminalRefundPaymentRequestHandler	Executed when a card payment is refunded using a payment terminal/device.
PaymentTerminalUpdateLinesRequestHandler	Executed when POS sends line item details to a payment device for display purposes.

REQUEST NAME	DESCRIPTION
PaymentTerminalVoidPaymentRequestHandler	Executed when a card payment is voided using a payment terminal/device.
PaymentTerminalBeginTransactionRequestHandler	Executed when a card payment is initiated using a payment terminal/device.
PaymentTerminalCancelOperationRequestHandler	Executed when a card payment is canceled using a payment terminal/device.
PaymentTerminalEndTransactionRequestHandler	Executed when a card payment is ended using a payment terminal/device.
CashDrawerOpenRequestHandler	Executed when a cash drawer open request is initiated by POS.
PaymentTerminalActivateGiftCardPeripheralRequestHandler	Executed when activate gift card request is initiated by POS.
PaymentTerminalAddBalanceToGiftCardPeripheralRequestHandler	Executed when add balance to gift card request is initiated by POS.

### Scan request handler

REQUEST NAME	DESCRIPTION
GetScanResultClientRequestHandler	Executed when you scan or key in a POS transaction screen Numpad.

### Store fulfillment request handler

REQUEST NAME	DESCRIPTION
PrintPackingSlipClientRequestHandler	Executed when you print a packing slip from the store fulfillment view.
MarkAsPickedServiceRequestHandler	Executed when you mark a sales line as picked from the store fulfillment view.

### Store operations request handler

REQUEST NAME	DESCRIPTION
CreateTenderRemovalTransactionClientRequestHandler	Executed when you do a tender removal operation in POS.
CreateFloatEntryTransactionClientRequestHandler	Executed when you do a float entry operation in POS.
SelectZipCodeInfoClientRequestHandler	Executed when you key in zip code in address add/edit view in POS.
CreateStartingAmountTransactionClientRequestHandler	Executed when you do a start amount declaration in POS.
LoyaltyCardPointsBalanceOperationRequestHandler	Executed when you do a loyalty card balance operation in POS.

REQUEST NAME	DESCRIPTION
GetReportParametersClientRequestHandler	Executed when you use a report parameter. If your POS report needs an input parameter this dialog will be executed to capture the parameters.
GetPickingAndReceivingOrdersClientRequestHandler	Executed when orders fetched for picking and receiving processing.
GetStartingAmountClientRequestHandler	Executed when you do a start amount declaration in POS (before navigating to the view).

### Tender counting request handler

REQUEST NAME	DESCRIPTION
CreateSafeDropTransactionClientRequestHandler	Executed when you do a safe drop operation in POS.
GetTenderDetailsClientRequestHandler	Executed when you get tender declaration details in POS.
CreateBankDropTransactionClientRequestHandler	Executed when you do a bank drop operation in POS.
CreateTenderDeclarationTransactionClientRequestHandler	Executed when you do a tender declaration operation in POS.
GetCountedTenderDetailAmountClientRequestHandler	Executed when you do a tendercount detail in POS.
CreateBankDropTransactionClientRequestHandler	Executed when you do a bank drop operation in POS.

### Sales orders request handlers

REQUEST NAME	DESCRIPTION
GetGiftReceiptsClientRequestHandler	Executed when you print a gift receipt in POS.
SelectCustomerOrderTypeClientRequestHandler	Executed when you get a dialog box with options to choose between customer order or quote.
GetCancellationChargeClientRequestHandler	Executed when you get a dialog box to enter the cancellation shipping charge during the customer order workflow.

## How to override a handler in POS

If you want to override any of the above POS request handler logic, you to need to use the following steps:

1. Create a new class and extend it from the corresponding handler class. For example, if you are overriding GetSerialNumberClientRequestHandler, then extend your class from GetSerialNumberClientRequestHandler.
2. Implement the executeAsync method.
3. Either call the default handler or do your custom logic inside the executeAsync method and return the response.

## Step by step instructions

The following example shows how to override the `GetSerialNumberClientRequestHandler` to automate the serial number entry in POS. By default, POS will display a dialog box to enter the serial number if the item is configured to ask for serial number. We want to avoid showing this dialog box and enter serial number through code.

1. Open Visual Studio 2015 in administrator mode.
2. Open ModernPOS solution from ...\RetailSDK\POS.
3. Under the POS.Extensions project, create a new folder called `POSRequestHandlerExtension`.
4. Under the `POSRequestHandlerExtension` folder, create new folder called `Handlers`.
5. In the `Handlers` folder, add a new `.ts` (typescript) file and name it `GetSerialNumberClientRequestHandlerExt.ts`.
6. Add the following import statement to import the relevant entities and context in the `GetSerialNumberClientRequestHandlerExt.ts` file.

```
import { GetSerialNumberClientRequestHandler } from
"PosApi/Extend/RequestHandlers/ProductsRequestHandlers";
import { GetSerialNumberClientRequest, GetSerialNumberClientResponse } from
"PosApi/Consume/Products";
import { ClientEntities } from "PosApi/Entities";
```

7. In the `GetSerialNumberClientRequestHandlerExt.ts` file, create a new class called `GetSerialNumberClientRequestHandlerExtend` and extend it from `GetSerialNumberClientRequestHandler`.

```
export default class GetSerialNumberClientRequestHandlerExt extends
GetSerialNumberClientRequestHandler { }
```

8. Implement the `executeAsync` method inside the `GetSerialNumberClientRequestHandlerExt` class. In the `executeAsync` method, you can write your custom logic and return the response or call the default handler. When POS sells the serial item, it will look for `executeAsync` to execute the logic for the serial number, however because we are overriding it, POS will now execute this overridden `executeAsync` method instead of the standard method.

### Sample implementation of how to override the `executeAsync` method

```

public executeAsync(request: GetSerialNumberClientRequest<GetSerialNumberClientResponse>):
    Promise<ClientEntities.ICancelableDataResult<GetSerialNumberClientResponse>> {

    // User could implement new business logic here to process the serial number.
    // The following example sets serial number "112233" for product 82001.

    if (request.product.ItemId === "82001") {
    let response: GetSerialNumberClientResponse = new GetSerialNumberClientResponse("112233");
    return Promise.resolve(<ClientEntities.ICancelableDataResult<GetSerialNumberClientResponse>>{

    canceled: false,
    data: response

    });
    }

    // If you don't want to execute custom logic on some conditions, and you just want to call the
    standard logic, you can call the default request, as shown below.

    return this.defaultExecuteAsync(request);

}

```

Full sample code:

```

import { GetSerialNumberClientRequestHandler } from
"PosApi/Extend/RequestHandlers/ProductsRequestHandlers";
import { GetSerialNumberClientRequest, GetSerialNumberClientResponse } from
"PosApi/Consume/Products";
import { ClientEntities } from "PosApi/Entities";

/**
 * Override request handler class for getting serial number request.
 */

export default class GetSerialNumberClientRequestHandlerExt extends
GetSerialNumberClientRequestHandler {

/**
 * Executes the request handler asynchronously.
 * @param {GetSerialNumberClientRequest<GetSerialNumberClientResponse>} request The request containing
the response.
 * @return {Promise<ICancelableDataResult<GetSerialNumberClientResponse>>} The cancelable promise
containing the response.
 */

public executeAsync(request: GetSerialNumberClientRequest<GetSerialNumberClientResponse>):
    Promise<ClientEntities.ICancelableDataResult<GetSerialNumberClientResponse>> {

    // User could implement new business logic here to process the serial number.
    // The following example sets serial number "112233" for product 82001.

    if (request.product.ItemId === "82001") {
    let response: GetSerialNumberClientResponse = new GetSerialNumberClientResponse("112233");
    return Promise.resolve(<ClientEntities.ICancelableDataResult<GetSerialNumberClientResponse>>{
    canceled: false,
    data: response
    });
    }

    return this.defaultExecuteAsync(request);
    }
}

```

9. Create a new json file under the POSRequestHandlerExtension folder. Name it manifest.json.
10. In the manifest.json file, copy and paste the following code. Be sure to delete the default generated code before copying this code.

```
{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Microsoft",
  "version": "7.3.5.0",
  "minimumPosVersion": "7.3.5.0",
  "components": {
    "extend": {
      "requestHandlers": [
        {
          "modulePath": "Handlers/GetSerialNumberClientRequestHandlerExt"
        }
      ]
    }
  }
}
```

11. Open the extensions.json file under the POS.Extensions project. Update it with POSRequestHandlerExtension samples, so that POS during runtime will include this extension.

```
{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": " SampleExtensions"
    },
    {
      "baseUrl": "POSRequestHandlerExtension"
    }
  ]
}
```

#### NOTE

The extension.json file should always contain two extensions folder names, so be sure to keep the SampleExtensions folder name or your custom extension folder name. For production, don't use the sample extensions. You should add your own extension folders and remove all the samples.

12. Open the tsconfig.json file to comment out the extension package folders from the exclude list. POS will use this file to include or exclude the extension for compilation. By default, the list contains all the excluded extensions list. If you want to compile any extension part of the POS, then you need to add the extension folder name and comment the extension from the extension list, as shown below.

```
"exclude": [

  // "SampleExtensions",
  // "SampleExtensions2",
  // "POSRequestHandlerExtension"

],
```

13. Compile and rebuild the project.

# How to test your extension

1. Press F5 and deploy the POS to test your customization.
2. After POS launches, sign in to POS and add a serial item to a transaction.
3. Place a break point in the extension code. When you add the serial item you should be able to debug the extension code.

## **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# View POS extension package information

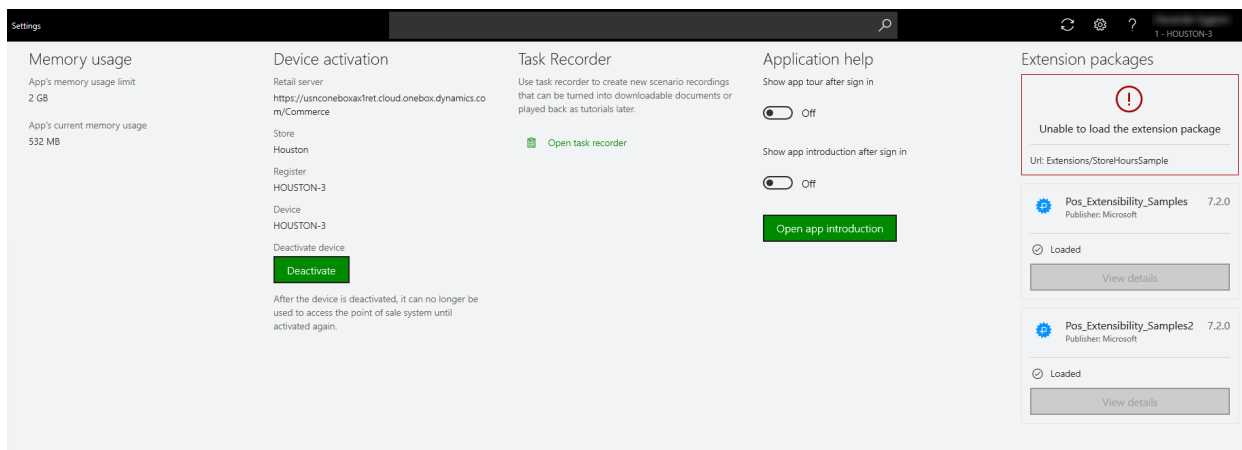
2/18/2021 • 2 minutes to read • [Edit Online](#)

In the **Settings** view in the point of sale (POS), the **Extension packages** section shows the list of POS extension packages that are included as part of the core POS. The tile for each package shows the status of that package. The package status indicates whether the extension was loaded, could not be loaded, or was skipped.

## Extension package status

This section describes what each package status means.

- **Loaded** – The extension package was successfully loaded.
- **Failed** – The extension package wasn't successfully loaded.
- **Skipped** – The package was skipped and wasn't loaded. In the extension manifest, you can specify that a package should be loaded for a specific locale, such as **en-fr**, but skipped for all the other locales.



### NOTE

Cloud POS will not display the extension version in the Customization.settings file under the **About** section on the **POS settings** page, it will only show the Microsoft app package version. Extension package versions can only be viewed from the **Extension details** section.

## Extension package details

If an issue occurs when an extension is loaded, or if there is a conflicting extension, you can use the details that are provided for each extension package to determine which extension file is causing the issue. In this way, you can troubleshoot the issue.

To view the details of an extension package, select **View details** on the tile for that package. The POS opens a new view, where you can see the details of all the individual extensions in the package. If any extensions weren't successfully loaded or were skipped, the details appear in the right pane.

The **Status** column shows the status of each extension in the package, the **Name** column shows the name of the extension type, and the **Path** column shows the path of the implementation file in the package. When you select a specific line item, the right pane also shows a description of the extension.

The information in this view is based on the manifest file that is included in the extension package. The POS extension loader loads all the extension packages and updates the status. The status information includes any

errors that have been logged.

**NOTE**

Dual display custom control and other extension details information related to dual display will not be shown in the extension details view.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Test recorder and Regression suite automation tool for Cloud POS

2/18/2021 • 21 minutes to read • [Edit Online](#)

This topic explains how to use the new test recorder tool in Cloud POS to record business scenarios for user acceptance testing (UAT) and user interface (UI) testing. It also explains how to automate test validation by using the Regression suite automation tool (RSAT). RSAT uses the Microsoft Azure DevOps test suite to download test cases. It then reports the results, together with the test execution status, back to Azure DevOps. The test cases can be manually created in Azure DevOps, or they can be synced from the Business process modeler (BPM) tool in Microsoft Dynamics Lifecycle Services (LCS) to Azure DevOps and then to RSAT.

This topic applies to Dynamics 365 Retail and Dynamics 365 Finance version 10.0.5 (October 2019) and later.

## NOTE

The test recorder is supported in Cloud POS only when the Google Chrome web browser is used. Support for other web browsers and device types will be added later.

## Test recorder

The test recorder in POS helps significantly reduce the time and cost of UAT. UAT is typically required before a Microsoft application update is applied, or before custom code and configurations are applied to your POS production environments.

The test recorder can record user actions in the client, and it provides exact fidelity for all controls and for all elements in the Document Object Model (DOM). In POS, the test recorder captures an event that has occurred and stores it, together with all relevant information about the corresponding user action, in real time. From this information, the test recorder can capture the type of user action (such as a button click, value entry, or navigation) and any data that is related to that user action (such as the value and type of input data, the view context, or the record context). However, password information isn't captured. During a recording session, the test recorder persists all the recorded information in memory. Then, at the end of the recording session, it generates an output file that includes enough detail so that RSAT can be used later to play back the actions just as the user performed them.

## IMPORTANT

The test recorder captures all the data that is entered during a recording session except POS user passwords. Don't record any personally identifiable information (PII), secrets, sensitive data, or user-specific data. All data that is entered during a recording session is stored in the Recording.xml file, and other users can see it in LCS and Azure DevOps, in the variables.xlsx and Recording.xml files, and during playback.

## Regression suite automation tool

RSAT lets functional power users run test cases in POS. It then updates the test execution results in Azure DevOps for reporting and investigation purposes.

RSAT provides options for investigating test failures. It also decouples the test parameters from test steps and stores the parameters in Microsoft Excel files. In that way, the values of test parameters can easily be edited.

# Prerequisites

- You must have a POS environment.
- Your test environment must be running binary update 10.0.5 or later.
- RSAT must have access to your test environment via a web browser.
- You must have Excel installed to generate and edit test parameters.
- You must have an Azure DevOps project to store and manage your test cases, test plans, and test case results.

# Enable test recording in the POS application

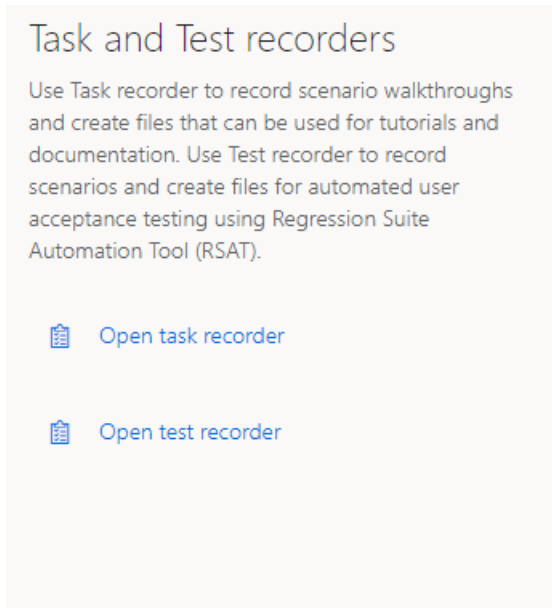
To turn on the test recording functionality in POS, follow these steps in Headquarters.

1. Go to **Retail and Commerce > Channel Setup > POS Setup > Registers**.
2. Select the register where the test recording functionality should be turned on.
3. On the **Register** tab, on the **General** FastTab, set the **Enable task and test recorder** option to **Yes**.
4. Select **Save**.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
6. Select the **Registers (1090)** job, and then select **Run now**.

# Controlling the test recorder

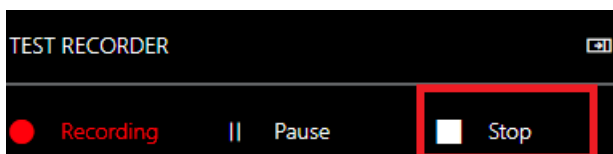
## Open the test recorder

To open the test recorder, sign in to Cloud POS, and then, on the **Settings** page, in the **Task and Test recorders** section, select **Open test recorder**.



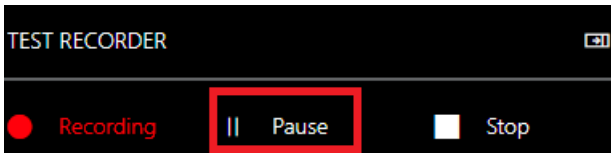
## Stop a recording session

To end a recording session, select **Stop**. Note that you can't restart a recording session after you end it. Therefore, make sure that the recording session is completed before you end it.



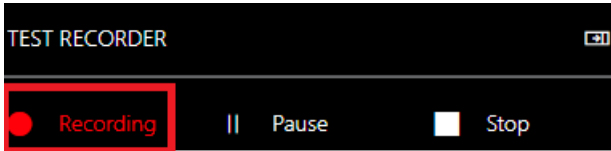
## Pause a recording session

To temporarily stop (pause) a recording session, select **Pause**. Steps that you perform after you select **Pause** aren't recorded.



### Continue a recording session

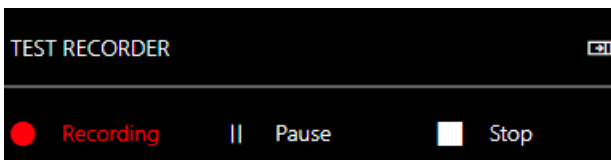
To resume a recording session after you've paused it, select **Recording**.



### Start and end a task

To help organize your procedures, you can group steps together into tasks. To specify the beginning and end of a set of grouped steps, use the **Start task** and **End task** buttons. Select **Start task** to add a "Start Task" step, and then perform the steps that should be included in the group. After you've finished performing the steps for the group, select **End task**.

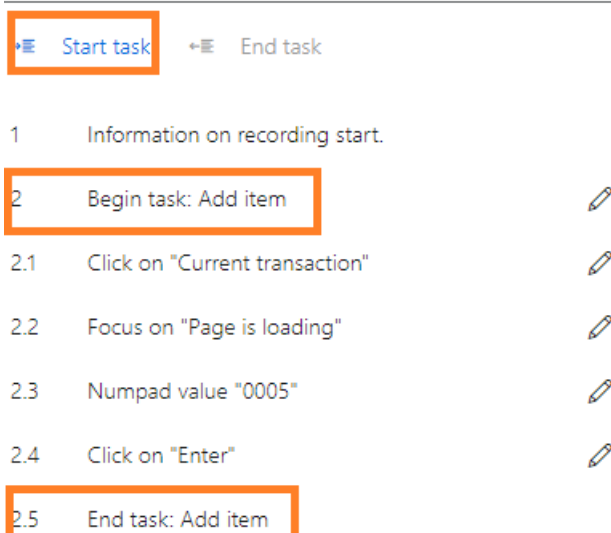
Tasks can be nested inside other tasks. In this way, you can better organize very long and complex business processes.



Enable validation mode



### Steps



### Add a new task

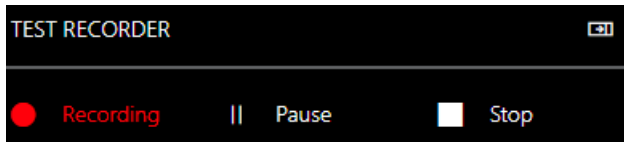
## New task

Name

Comment

### **Add an annotation**

An annotation is additional text that you add to a step in a recording. For example, you can use annotations to give the user more context or instructions. You can add an annotation to any step by selecting the **Edit** button (pencil symbol) to the right of the step.



Enable validation mode



## Steps

---

→ Start task   ← End task

- 1 Information on recording start.
- 2 Click on "Lines"



### Add text and notes

You can use the **Text** and **Notes** fields in the annotation dialog box to add text that should be associated with a step in a task guide.

- **Text** – Text that you enter in this field appears *above* the step text in the test steps.
- **Notes** – Text that you enter in this field appears *below* the step text.

### Change input values

You can change user input values that are entered during a recording session. For example, if you added product 0005 during the recording session, the product ID is stored by default in the Recording.xml file. If you want to specify a different product ID, you can change the value here. The value will be shown only if there is user input.



## Current step

Numpad value "0005"

Text

Numpad value "0005"

Notes

Value

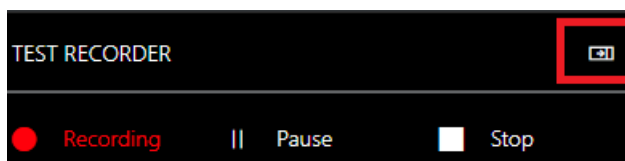
0005

Save

Cancel

### Hide the test recorder pane

To hide and show the test recorder pane during a recording session, select the collapse button.



Enable validation mode

No

### Test recorder floating control



The test recorder floating control is useful when the test recorder pane is hidden during a recording session. The test recorder pane overrides non-error dialog boxes and/or part of the POS view. Therefore, you must sometimes hide the pane to add validation in the dialog boxes or select controls. If the test recorder pane is hidden, but you must still be able to access test recording functionality (for example, you must turn on validation mode, or pause or continue the recording session), you can use this floating control.

Test Recorder



The following sections describe the controls on the floating control.

#### **Move control**

The move control lets you move the floating control within the POS app.

#### **Validation mode**

When you turn on validation mode, the recording session is paused. To continue the recording session, you must turn off validation mode.

#### **Pause**

To temporarily stop (pause) the recording session and continue the operation, select the **Pause** button. Steps that you perform after you select **Pause** aren't recorded.

#### **Recording**

To resume the recording session after you've paused it, select **Recording**.

## Record a test case in POS

### **Create a recording**

#### **IMPORTANT**


Before creating the recording or test execution/playback, turn off the Show app tour and Show app introduction after sign in. To do this, go to the **CPOS Settings > Application help** section (applicable only if Dynamics 365 Commerce demo data is used). Turn this off in the demo data by running the following script in your channel database: Update [ax]. [SYSSERVICECONFIGURATIONSETTING] SET VALUE = '0' WHERE NAME = 'APPTOUR'.


Follow these steps to create a new recording by using the test recorder:

1. Launch Cloud POS.
2. Select the hamburger icon on the left pane and select **Settings**. Don't sign in to Cloud POS. The sign in step must be recorded as part of the test recording flow, so you need to launch the recorder before sign in.)
3. On the **Settings** page, in the **Task and Test recorders** section, select **Open test recorder**.

## Task and Test recorders

Use Task recorder to record scenario walkthroughs and create files that can be used for tutorials and documentation. Use Test recorder to record scenarios and create files for automated user acceptance testing using Regression Suite Automation Tool (RSAT).

 [Open task recorder](#)

 [Open test recorder](#)

4. Select **Create a new recording**.

## What would you like to do?



Create a new recording

Privacy statement: Any information that you enter into the application while you are recording will be captured and included in the recording file, Training document and BPM package. If you decide to share the recording file, training document or BPM package others may be able to see the information that was captured.

Cancel

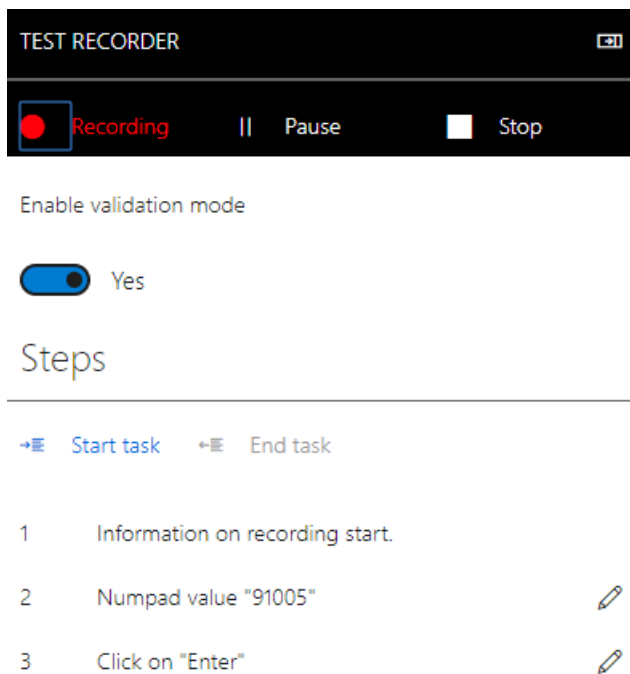
5. Enter a name and description for the recording, and then select **Start**.

The test recorder enters recording mode, and the recording session begins. The test recorder pane shows information and controls that are related to the recording session.

### NOTE

All test recording must begin from the Cloud POS sign in page. If the recording is started from the home page, any transaction or other page playback will fail and you will need to start the recording again from the sign in page.

6. Perform the needed actions in the POS user interface.



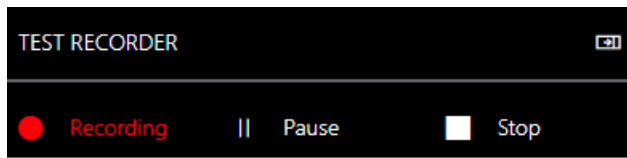
### Validation mode

When you use validation mode during a recording session, users can then validate values during test execution. For example, you can use the validation mode feature if you want to validate label text or an error message, or if you want to validate that the item price or tax is calculated correctly. To turn on validation mode during a recording session, use the **Enable validation mode** option.

1. Set the **Enable validation mode** option to **Yes**.
2. Select values or text in the POS to add a validation step. No validation is done for passwords, sensitive data, or fields where the test recorder can't get the field values. During test execution, the playback engine determines whether the value is same. The test case then either passes or fails, depending on the result of the validation.

#### NOTE

While validation mode is turned on, the test recorder will be in a paused state. It will just add validations steps, and POS won't respond to any user actions except the addition of validation steps. For example, in validation mode, you can't open a different POS view or use any POS functionality. To continue the recording session, you must turn off validation mode by setting the **Enable validation mode** option to **No**.





Enable validation mode



## Steps

→ Start task   ← End task

- 1 Information on recording start.
- 2 Numpad value "91005" 
- 3 Click on "Enter" 

3. To end the recording session, select **Stop**.

### Download options

After you end a recording session, you can download the recording by selecting **Save to this PC**.



## Value

Name

Sale transaction



Save to this PC

Privacy statement: Any information that you enter into the application while you are recording will be captured and included in the recording file, Training document and BPM package. If you decide to share the recording file, training document or BPM package others may be able to see the information that was captured.

The .axtr file is saved to the local file system. You must manually upload this file to LCS or Azure DevOps and then either delete it from the file system or secure it.

To upload to Azure DevOps directly:

1. Change the .axtr file extension to .zip.
2. Open the .zip package.
3. Inside the package there will be file with name Recording.xml. Upload the Recording.xml to the test case in Azure DevOps. Don't upload the entire .zip or .axtr package.

## Install RSAT

Download the Microsoft Windows Installer (MSI) package file for RSAT from [Regression Suite Automation Tool](#).

Double-click the MSI file to run it. After you install RSAT, you must install drivers for Selenium and the web browser.

#### NOTE

Before you run the test, you must set up Azure DevOps, and you must complete the required general settings and other required settings in RSAT. For detailed steps, see [Regression suite automation tool installation and configuration](#).

The following procedure describes the configuration that is required to run the POS test cases.

### Configure the POS settings

1. Open RSAT from your desktop.
2. Select the **Settings** button in the upper right to configure RSAT.
3. In the **Settings** dialog box, click the **Optional** tab and select the **Configure Retail POS** option to enable the Retail POS automation.
4. In the **Settings** dialog box, on the **Retail POS** tab, on the **Playback environment** tab, set the following fields:
  - **Cloud POS URL** – Enter the URL of the Cloud POS environment where you want to run the test.
  - **Commerce Scale Unit URL** – Enter the Commerce Scale Unit URL that should be used for device activation, if the device hasn't already been activated.

#### NOTE

The Cloud POS and Commerce Scale Unit URL can be obtained from Finance and Operations environment. Navigate to **Retail and Commerce > Channel setup > Channel profiles**. You can also obtain the URLs from the LCS environment page.

- **AAD user email** – Enter the email address of the Azure Active Directory (Azure AD) user that should be used for device activation. The Azure AD user must have permission to activate the device.
- **AAD password** – Enter the password of the Azure AD user that should be used for device activation.
- **Register number** – Enter the ID of the register number (channel) where the test should be run.
- **Device** – Enter the ID of the device where the test should be run.
- **Default wait time** – Enter the wait time, in seconds, before the test case fails if any element isn't found. During test execution, the playback engine keeps trying to find element until this default wait time has passed. It then fails the test case and notifies you that the element that was recorded wasn't found or loaded for playback.

Settings ×

General Optional Retail POS Preview Process

Playback environment POS login credentials

Cloud POS URL:

Commerce Scale Unit URL:

AAD user email:

AAD password:

Register Number:

Device:

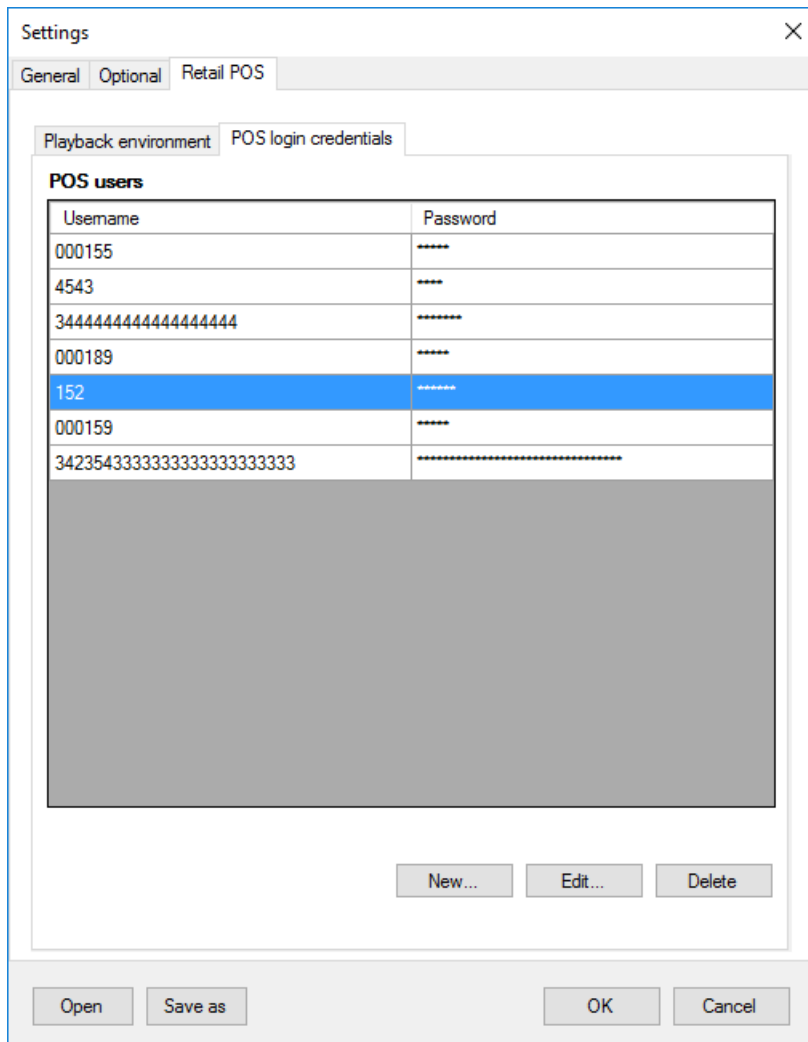
Default wait time (seconds):

Open Save as OK Cancel

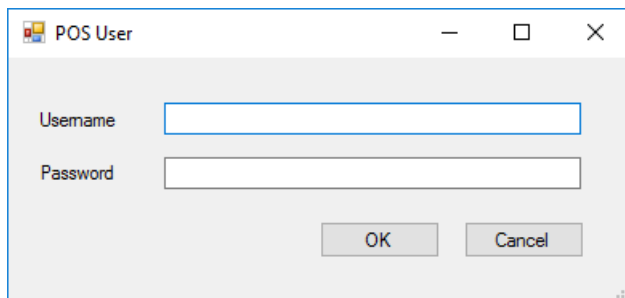
5. Select the **POS login credentials** tab.

During a recording session, the test recorder captures only the user name from the POS. It doesn't store any password. However, to run the test, you must have both the user name and the password that are used to sign in to POS. This tab captures the POS user name and password, so that the password information is securely stored outside the recording file. During test execution, the user ID is then mapped to the same user ID that is entered in RSAT, and the password is retrieved.

Therefore, on the **POS login credentials** tab, you must enter all the user name and password information that was used during the recording session, so that the password can be retrieved during test execution. Otherwise, test execution will fail, and you will be notified that sign-in details weren't found.



6. Select **New**.



7. In the **Username** field, enter the user name for sign in to POS.

8. In the **Password** field, enter the password for sign in to POS.

9. Repeat steps 6 through 8 to enter other user names and passwords.

10. To edit a set of POS sign-in credentials, select **Edit**.

11. To delete a set of POS sign-in credentials, select **Delete**.

## Run tests

This section explains how to load test cases from Azure DevOps, generate automation files, modify test parameters, run tests, investigate results, and save your work back to Azure DevOps.

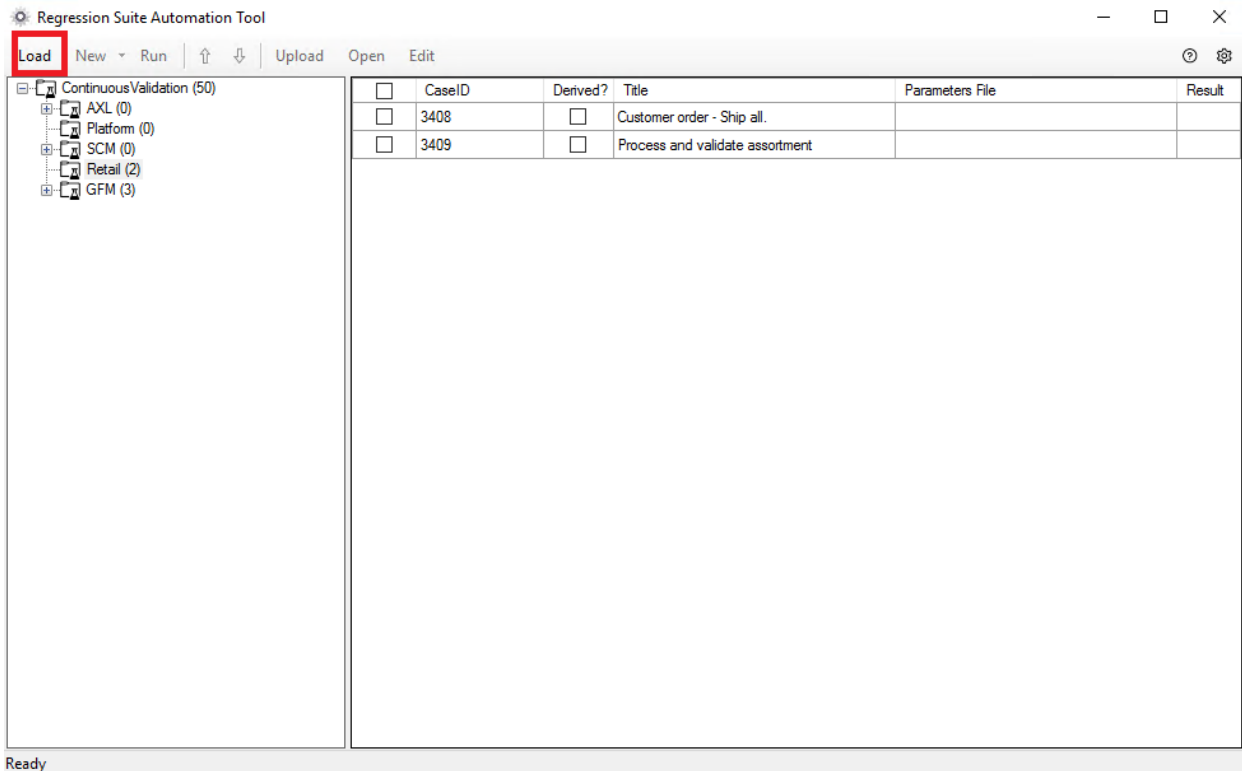


## NOTE

For detailed information about how to set up Azure DevOps and test cases, see [Regression suite automation tool installation and configuration](#). You must complete that setup before you start to run tests. Before test execution/playback, turn off the Show app tour and Show app introduction after sign in. Turn this off in the demo data by running the following script in your channel database: Update [ax].[SYSSERVICECONFIGURATIONSETTING] SET VALUE = '0' WHERE NAME = 'APPTOUR'.

## Load test cases and create parameter files

In RSAT, select **Load** to download test cases and test case automation files from Azure DevOps. All test cases that belong to the test plan that is specified in the **Settings** dialog box are downloaded.



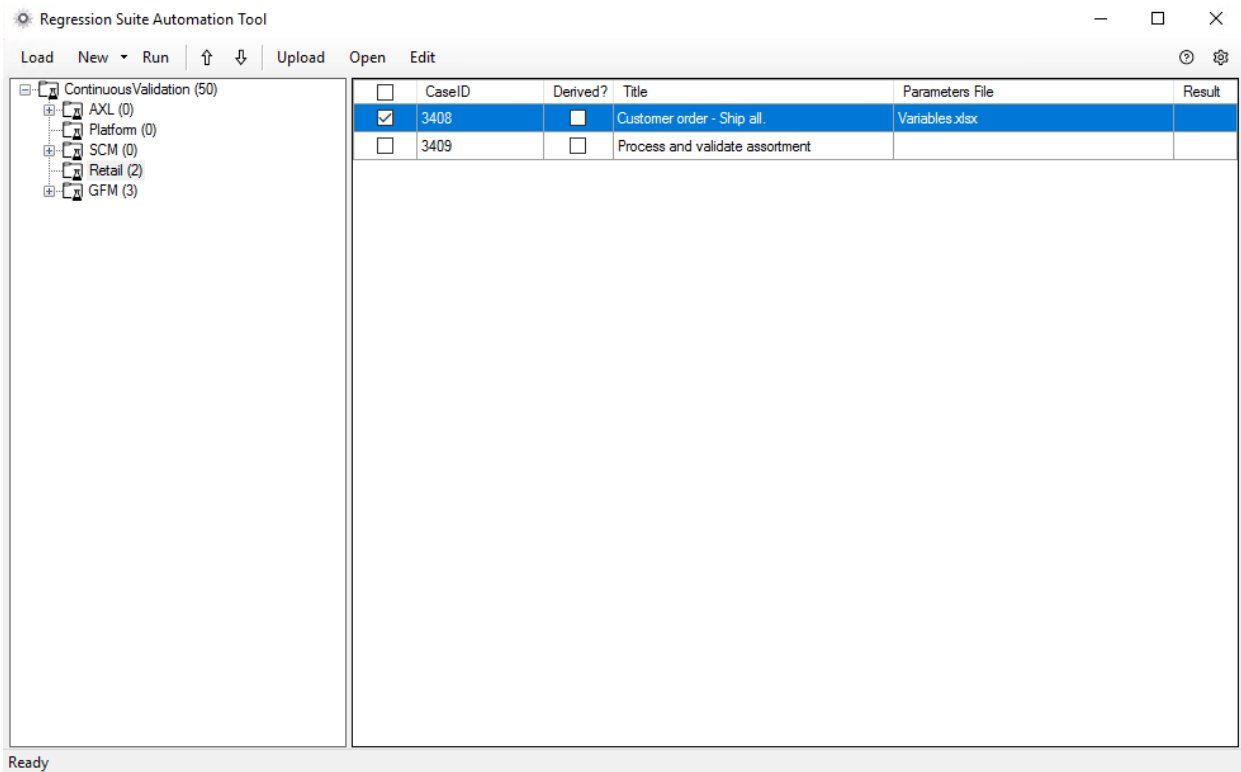
Test cases are organized by test suite under a common test plan. The test suites are the test suites that you created in your Azure DevOps project. By using RSAT, you can work with one test suite at a time. If RSAT can't load any test case, verify that your test plan was correctly created in Azure DevOps, and that it contains the required test suites and test cases.

If you're loading the test plan for the first time, the **Parameters File** column in the grid is blank, and you must generate test automation parameter files for your test cases.

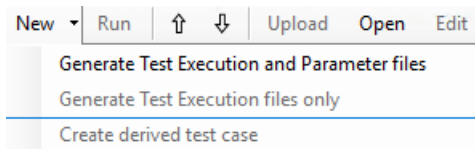
To run the tests, generate the following test automation files:

- Test parameter files (Excel files that contain test case parameters)
- XML files that are required to run the tests

When you select **New**, test automation files are generated in your working directory. The Excel test parameter files appear in the **Parameters File** column in the grid.



For the test recording files, the **Generate Test Execution files only** option is unavailable. Because Cloud POS uses Selenium WebDriver directly to do the playback, no additional script file must be generated.



### Modify test parameters and validation values

This section explains how to modify Excel files by specifying input and validation parameters for your test run.

In RSAT, select one or more test cases to modify, and then select **Edit**. An Excel window is opened for each test case that you selected. Alternatively, you can open the Excel files directly from the working directory.

In addition to a **Summary** tab, the Excel file includes a **Variables** tab that has the details of all the variables that were generated. POS automatically generates variables for all the input values that are entered during a recording session. You don't have to generate the variables separately. Each variable has a unique variable ID that you can pass, in order, to different test cases in a single instance of test execution. All the variables on the **Variables** tab appear in the order that they were entered in during the recording session.

To pass variables or values between POS test cases, select the test cases in the RSAT tool and open the Variables.xlsx file by selecting the Excel icon in the tool. Copy the Variable ID (Column C) value and paste it in the Variable value field (Column D). For example, to pass the Receipt ID from **Test case 1 - variable.xlsx** to **Test case 2 - variable.xlsx**, copy the variable ID value from column C: c8cc0571-9a27-b3c5-0749-c26c3cca6afe. Paste the value in the Variable value column D in to the test case 2 - variable.xlsx file, in braces: `{{ c8cc0571-9a27-b3c5-0749-c26c3cca6afe}}`

#### Test case 1 – Variable.xlsx

A (DESCRIPTION)	B (VIEW NAME)	C (VARIABLE ID)	D (VARIABLE VALUE)
Receipt Id "HOU123R456"	ShowJournalView	c8cc0571-9a27-b3c5-0749-c26c3cca6afe	HOU123R456

#### Test case 2 – Variable.xlsx

A (DESCRIPTION)	B (VIEW NAME)	C (VARIABLE ID)	D (VARIABLE VALUE)
Receipt Id "HOU123R456"	ShowJournalView	80f23afa-5b76-5442-d16a-6cc9b8b245cb	{{c8cc0571-9a27-b3c5-0749-c26c3cca6afe}}

### Validate expected values

Validation of expected values is an important component of a test case. When you create your test cases, you can define validation parameters by using the test recorder in validation mode.

During the recording session, turn on validation mode. Then, while the test recorder is recording, select all the fields that must be validated. This action becomes a validation step that you can use with RSAT. The validation values will appear, in the order that they were entered in, on the **Variables** tab in the Excel file. You can then modify the values in the Excel file before test execution, and the new values will be used for data entry and validation during test execution.

	A	B	C	D
1	Description	View Name	Variable ID	Variable Value
2	In "Operator ID" set v	LoginView	68ff7367-0df0-37ab	
3	In "Password" set val	LoginView	c15c9013-e586-2ad	
4	Numpad value "9100"	CartView	262d3277-81db-ed19	9100
5	Numpad value "91005"	CartView	66af3a4a-4cce-87ac	91005
6	Numpad value "2001"	CartView	c4e5b0ab-4181-2a7	2001
7	Validation on "AMOU"	CartView	61bd5f98-d6ec-a25	\$141.31
8	Validation on "\$8.31"	CartView	de1e5cb8-eead-92	\$8.31
9				

### Run

In RSAT, select **Run** to run the selected test cases. You can run only test cases that automation files have been generated for. RSAT opens POS and runs the tests by using the data that is entered in Excel. After the test run, the results are updated in the **Result** column in RSAT, and also in Azure DevOps.

To change the order that test cases are run in, use the up arrow and down arrow buttons.

### Investigate results

After test cases have finished running, the pass or fail status appears in the **Result** column in RSAT. More details are available in Azure DevOps, and you can use them to investigate the results. From your Azure DevOps project page, go to **Test > Runs**.

All error messages are also available locally at

C:\Users\%YourUserName%\AppData\Roaming\regressionTool\errormsg<TestCaselD>.txt.

## System and metadata files

The following table shows the files that are generated during recording sessions, test execution, and playback.

FILE NAME	DESCRIPTION	FILE GENERATION FLOW	FILE SAVE FLOW
Recording.xml	This file contains all the steps that are required to play back a recording. It includes all user-specific values for each step in the recording.	The file is generated by the user when that user records a test case and uploads it to Azure DevOps so that it can be used by RSAT.	The file is saved to disk when the test case is loaded into RSAT.

FILE NAME	DESCRIPTION	FILE GENERATION FLOW	FILE SAVE FLOW
Variables.xml	This file contains the values for all the variables that are used in the recording file. It's used by the playback tool.	The file is generated when the user selects <b>New &gt; Generate Test Execution and Parameter files</b> in RSAT.	The file is saved when the user selects <b>New &gt; Generate Test Execution and Parameter files</b> .
Variables.xlsx	This file contains the values for all the variables that are used in the recording file. It can be modified by the user and is used by the playback tool.	The file is generated when the user selects <b>New &gt; Generate Test Execution and Parameter files</b> in RSAT.	The file is saved when the user selects <b>New &gt; Generate Test Execution and Parameter files</b> .
OutputLog.txt	This file contains a log of execution of the playback process. It includes a description of each step that was run. It might also include an exception. Depending on the exception, it might contain data that is available in the Recording.xml file.	The file is generated after the playback tool in RSAT has finished running, regardless of whether the test case was successfully played back.	The file is saved when a test case is played back.
Time.xml	This file contains the list of steps, the user-edited description of each step, and the amount of time that the execution of each step required.	The file is generated after the playback tool in RSAT has finished running, but only if the test case was successfully played back.	The file is saved when a test case is successfully played back.
Out.xml	This file contains the values for all the variables that are used in the recording file. For each variable, this file uses the updated value from the Variables.xlsx file. The playback tool uses this file to support test cases that depend on variables from other test cases.	The file is generated after the playback tool in RSAT has finished running, regardless of whether the test case was successfully played back.	The file is saved when a test case is played back.
In.xml	This file contains the values for all the variables that are used in the recording file of every test case that was run before the current test case. It uses the updated values from each Variables.xlsx file. The playback tool uses this file to support test cases that depend on variables from other test cases.	The file is generated before the playback tool in RSAT is run, when multiple test cases are run.	The file is saved when the previous test case has finished running and before a new test case is run.

You must manually delete these files and secure them as you require. All these files are stored in the RSAT working directory.

## Best practices

## Creating test cases by using the test recorder

- Before creating the recording or test execution/playback, turn off the Show app tour and Show app introduction after sign in. To do this, go to the **CPOS Settings > Application help** section (applicable only if Dynamics 365 Commerce demo data is used). Turn this off in the demo data by running the following script in your channel database: Update [ax].[SYSSERVICECONFIGURATIONSETTING] SET VALUE = '0' WHERE NAME = 'APPTOUR'.
- Disable Chrome extension - If possible, disable the Chrome extension in the Chrome browser that used to record and play back. The Chrome extension may change the DOM element xpath, which can result in test case failure because the element is not found during the execution of steps (xpath is different from the recording).
- Keep individual recordings short, and focus on a business task that is performed by one user, such as the creation a sale transaction. This approach makes it easier to maintain and reuse test cases.
- Don't record any scenario that includes secrets.
- Recording and playback must be done in the same screen layout and at the same resolution. If recording and playback are done in different layouts and at different resolutions, playback will fail.
- You can't change the POS user name during playback of a recording. When you make a recording, always use the same user name that will be used later for playback.
- Recording the POS activation flow is not supported.
- Keystroke recording performance may be slow, so type slowly while recording so that all the events are captured property.
- Peripheral emulation is currently not supported, use a keyboard wedge-based device.
- Don't hold a key down during recording, as this could record multiple key press events.
- Extension controls should follow the best practice of proper XPath and if possible have a unique ID for each HTML element.

## Troubleshooting guides

### Chrome driver

If playback fails by flickering (opens and closes browser multiple times without starting playback), this could be related to the Chrome driver version. Check the error log in the RSAT tool. If the error states that the Chrome driver version is not supported, then download the supported chromedriver.exe version mentioned in the error message and paste it in the ...\\Regression Suite Automation Tool\\Common\\External\\Selenium folder. You can download the Chrome driver from [ChromeDriver](#).

### Disable Chrome extension

If possible, disable the Chrome extension in the Chrome browser that used to record and play back. The Chrome extension may change the DOM element xpath, which can result in test case failure because the element is not found during the execution of steps (xpath is different from the recording).

### .NET standard error

If you get the following 'netstandard' error, install .NET Framework 4.8 runtime. You can download the .NET Runtime from [Download SDKs](#).

Unhandled Exception: System.IO.FileNotFoundException: Could not load file or assembly 'netstandard, Version=2.0.0.0, Culture=neutral, PublicKeyToken=cc7b13ffcd2ddd51' or one of its dependencies. The system cannot find the file specified at Microsoft.Dynamics.Commerce.PosPlayback.RecordingsRunner.Program.Main(String[] args).Multifactor authentication:

### Multifactor authentication

If multifactor authentication is enabled for the device activation user, then playback may fail. If possible, disable the multifactor authentication for the activation user temporarily. After the activation is completed, then re-

enable the multifactor authentication. Activation will be required only for the first-time playback. We recommend that you discuss this approach with your security experts before making this change.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add custom fields to the point of sale (POS) Totals panel

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic explains how to add a new custom field to the **Totals** panel on the POS transaction screen by using the screen layout designer. This topic is applicable to Microsoft Dynamics 365 for Finance and Operations 7.3 and later, and to Microsoft Dynamics 365 Retail 7.3 and later.

Custom fields that you add for the **Totals** panel on the **Custom fields** page will appear in the designer. You can then select which custom fields should be in the left and right columns. The logic for the custom fields should be coded in the point of sale (POS) extensions.

## Scenario/business problem

You will add a custom field to the **Totals** panel on the POS transaction screen by using the screen layout designer. You will also code the business logic for the custom field in the POS extensions.

## Overview of the required steps

First, you must complete these steps to configure Headquarters.

1. On the **Language text** page, add the language text for the custom field.
2. On the **Custom fields** page, add the new custom field.
3. In the screen layout designer, add the new custom field to the **Totals** panel.
4. Run the **Registers (1090)** job.

You must then complete this step in the POS extension project.

- Add the business logic for the custom field.

## Configure Headquarters

1. Sign in to Commerce.
2. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Language text**.
3. On the **POS** tab, select **Add** to add a new POS language text.

The text that appears in the **Totals** panel can be localized. Therefore, you can create multiple texts in different languages for the same text ID. Here is an example.

LANGUAGE ID	TEXT ID	TEXT
en-US	1	Sample
en-UK	1	Demo

4. On the Action Pane, select **Save** to save your changes.
5. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Custom fields**.
6. On the Action Pane, select **New** to add a new custom field, and specify the following information:

- a. In the **Name** field, enter the name of the custom field.
- b. In the **Type** field, select **Totals area**.
- c. In the **Caption text ID** field, specify the text ID that you used in step 3.

Here is an example.

NAME	TYPE	CAPTION TEXT ID
Sample	Totals area	1

7. On the Action Pane, select **Save** to save your changes.
8. Go to **Retail and Commerce > Channel setup > POS setup > POS > Screen layouts**.
9. Select the **F3MGR** screen layout ID, and then, on the Action Pane, select **Designer**.

#### NOTE

You can select any existing layout or create a new layout. The **F3MGR** screen layout ID is available only if you're using demo data.

10. Select the **1440x960 – Full** layout size, and then select **Layout designer**.
11. If you're prompted to confirm that you want to open the application, select **Open**, and then follow the installation instructions.
12. After the designer is installed, you're asked for Azure Active Directory (Azure AD) credentials. Enter the information to start the designer.
13. In the designer, drag the **Totals panel** control from the left pane to the screen layout. If the layout already includes a **Totals** panel, the control appears dimmed in the left pane.
14. In the screen layout, right-click the **Totals panel**, and then select **Customize**.
15. In the **Customization - Totals panel** dialog box, the **SAMPLE** custom field should appear in the **Available fields** list. Move it to either the left column or the right column by using the arrow buttons.
16. To move the custom field up or down, use the **Up** and **Down** buttons.
17. Select **OK** to save your changes and close the **Customization - Totals panel** dialog box.
18. Select the **Close** button (X) in the upper-right corner to close the designer.
19. When you're prompted to save your changes, select **Yes**. If you select **No**, your changes aren't saved.
20. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
21. Select the **Registers (1090)** job, and then select **Run now**.

## Add business logic to the custom field

You can find similar sample code in the Retail software development kit (SDK), at ...  
 \RetailSDK\POS\Extensions\SampleExtensions\ViewExtensions\Cart\TipsCustomField.ts.

1. Start Microsoft Visual Studio 2015 in administrator mode.
2. Open the **ModernPOS** solution from ... \RetailSDK\POS.
3. Under the **POS.Extensions** project, create a folder that is named **CustomFieldExtensions**.



4. Under the **CustomFieldExtensions** folder, create a folder that is named **Cart**.
5. In the **Cart** folder, create a **.ts** (typescript) file that is named **SampleCustomField.ts**.
6. In the **SampleCustomField.ts** file, add the following **import** statement to import the relevant entities and context.

```
import { CartViewTotalsPanelCustomFieldBase } from "PosApi/Extend/Views/CartView";
import { ProxyEntities } from "PosApi/Entities";
```

7. Create a class that is named **SampleCustomField**, and extend it from the **CartViewTotalsPanelCustomFieldBase** class. In this way, the **computeValue** method from the base class will do any custom logic for the custom field.

```
export default class SampleCustomField extends CartViewTotalsPanelCustomFieldBase {
}
```

8. Inside the **SampleCustomField** class, implement the abstract **computeValue** method to set the logic for the custom field.

```
public computeValue(cart: ProxyEntities.Cart): string {
    // Let's show 10% of total amount in the custom field.
    if (isNaN(cart.TotalAmount) || cart.TotalAmount <= 0) {
        return "$0.00";
    }
    return "$" + (cart.TotalAmount * 0.1).toFixed(2).toString();
}
```

The overall class should look like this.

```
import { CartViewTotalsPanelCustomFieldBase } from "PosApi/Extend/Views/CartView";
import { ProxyEntities } from "PosApi/Entities";

export default class SampleCustomField extends CartViewTotalsPanelCustomFieldBase {
    public computeValue(cart: ProxyEntities.Cart): string {
        // Let's show 10% of total amount in the custom field.
        if (isNaN(cart.TotalAmount) || cart.TotalAmount <= 0) {
            return "$0.00";
        }
        return "$" + (cart.TotalAmount * 0.1).toFixed(2).toString();
    }
}
```

9. In the **CustomFieldExtensions** folder, create a **.json** file that is named **manifest.json**.
10. In the **manifest.json** file, paste the below code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Contoso",
  "version": "7.3.0",
  "minimumPosVersion": "7.3.0.0",
  "components": {
    "extend": {
      "views": {
        "CartView": {
          "totalsPanel": {
            "customFields": [
              {
                "fieldName": "Sample",
                "modulePath": "Cart/SampleCustomField"
              }
            ]
          }
        }
      }
    }
  }
}

```

In the manifest, note that the **fieldName** value in the **customFields** section should match the name of the custom field added in the Headquarters, the name you specified for the custom field in step 6 of the "Configure Headquarters" procedure. **modulePath** is the name of the implementation file, the implementation file name is the name of the file that you created in step 5 of "Add business logic to the custom field" procedure.

If you add multiple custom fields, you should add multiple implementation files and update the information under the customFields section.

For example, if you add two custom fields, **Sample1** and **Sample2**, you should have two implementation files that extend from the same base class, **CartViewTotalsPanelCustomFieldBase**.

In this case, the manifest should look like this.

```

"customFields": [
  {
    "fieldName": "Sample1",
    "modulePath": "Cart/SampleCustomField1"
  },
  {
    "fieldName": "Sample2",
    "modulePath": "Cart/SampleCustomField2"
  }
]

```

In this example, **SampleCustomField1** and **SampleCustomField2** are the names of the typescript files where you will do the business logic.

11. Open the **extensions.json** file under the **POS.Extensions** project, and update it with the **CustomFieldExtensions** samples, so that POS will load this extension at runtime.

```
{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "CustomFieldExtensions"
    }
  ]
}
```

12. Open the `tsconfig.json` file, and add the extension folder name `CustomFieldExtensions` and comment out the extension folder in the `exclude` list. POS will use this file to include or exclude the extension. By default, the `exclude` list contains all the excluded extensions. To include an extension as part of the POS, you must add the extension folder name and comment out the extension in the `extension` list, as shown here.

```
"exclude": [
  "SampleExtensions",
  //"SampleExtensions2",
  //"CustomFieldExtensions"
],
```

13. Compile and rebuild the project.
14. Deploy the customized version of Retail Modern POS (MPOS) by selecting the **Local Machine** button. Make sure that the solution platform is x86. Alternatively, you can create a deployable package and install MPOS from it.

#### NOTE

Although MPOS is used in this topic, you can use either MPOS or Cloud POS.

## Validate the customization

1. Sign in to MPOS by using **000160** as the operator ID and **123** as the password.
2. On the welcome screen, select the **Current transaction** button.
3. Add any item to the transaction.

The custom field should appear in the **Totals** panel.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Show custom notifications in the POS

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains how to add custom notifications in the point of sale (POS). This topic applies to Microsoft Dynamics 365 for Finance and Operations, Enterprise edition 7.3 and Dynamics 365 for Retail 7.3, and later versions that have the latest binary fix.

You can extend the POS notification framework for these scenarios:

- Show a custom notification so that a store associate can either perform a required action, based on that notification, or perform custom logic.
- Periodically perform some operation in the background. However, this scenario isn't recommended, because it might cause performance issues.

## How custom notifications work

The notification framework in the POS periodically runs for the configured operation in Retail headquarters. (The frequency of the periodic runs can be configured.) It calls the notification service in the Commerce runtime (CRT) and Real time service in Retail headquarters. When the POS makes the call to the notification service in CRT, CRT runs the business logic and determines whether any notification must be sent to the POS for the operation. The operation ID is passed to the request as a parameter. The CRT notification service checks the operation ID and returns the notification details.

If the notification exists, CRT returns the response to the POS and reports that a notification is available. POS parses this notification and shows it in the user interface (UI). When the POS user selects the notification, the relevant operation in POS is run. Notifications are linked to an operation. Inside the operation handler, you write the business logic that must be performed.

When CRT returns the response for the notification, it returns the number of notifications, messages, and action parameters. The POS framework has built-in logic to parse the response to get the POS operation ID and call it when the user selects the notification.

Inside the operation, you can write custom logic to specify what should happen when the POS user selects the notification. Additional parameters can be sent from CRT to the POS by using the action property. Those parameters will then be available inside the operation request in the POS.

### Example

To notify the POS user to prepare orders for pickup, inside CRT, check whether there is an order for pickup. If there is, update the notification response with the relevant parameters. The POS parses the response and shows the notification. When the POS user selects the notification, the POS calls the operation by using the parameter from the notification. (The extension logic determines whether parameters should be sent.) Inside the operation, the POS reads all the pending orders for pickup and shows the orders in the UI so that the POS user can take further action.

## Required steps to make notifications available for a custom operation

1. In Retail headquarters, go to **Retail and Commerce > Channel Setup > POS Setup > POS > POS operations**, and create an operation.

#### NOTE

For custom operations, use an operation ID that is above 5000.

2. Extend the notification service. When the scheduler for the notification service runs, it calls the extension code for both CRT and the Real time transaction service to get the notification message.
3. Extend the POS. In the POS, write logic inside the POS operation to specify what should happen when the POS user selects the notification. For information about how to create a POS operation, see [Add POS operations to POS layouts by using Button grid designer](#).
4. In Retail headquarters, configure the notification service with the custom operation. Then, when the scheduler runs the notification service, it includes the custom operation. For information about how to configure notifications in the POS, see [Add POS operations to POS layouts by using Button grid designer](#).

When the POS user selects the notification, the framework calls the correct operation, based on the response that is returned from CRT. The notification service runs for all the notifications that are configured in Retail headquarters, and it returns the response that includes the operation ID and notification details. Therefore, the POS will have the context of the operation ID, and the framework will use this context to call the operation implementation.

#### NOTE

The notification scheduler checks both CRT and Retail headquarters for new notifications. Depending on the scenario, you can extend the Real time transaction service in Retail headquarters or only in CRT. In scenarios where real-time processing is required for notifications, extend the Real-time transaction service method in Retail headquarters. If real-time processing isn't required, extend only CRT.

## Extend the notification service in CRT

To extend the notification service in CRT, override `GetNotificationsExtensionServiceRequest`, and return `GetNotificationsExtensionServiceResponse`.

The Retail software development kit (SDK) includes a sample that shows how to extend the notification service (`RetailSDK\SampleExtensions\CommerceRuntime\Extensions.NotificationSample`).

- `GetNotificationsExtensionServiceRequest` – This class contains the operation ID, staff, and channel. Based on the configuration of Retail headquarters, the POS runs the notification scheduler for each configured operation.
- `GetNotificationsExtensionServiceResponse` – In the response, return the notification detail entity (`NotificationDetailCollection`), update `GetNotificationsExtensionServiceResponse`, and return all the notifications. Because the response is a collection, multiple notifications can be returned.

## Properties of the notification detail entity

The notification detail entity has the following properties.

PROPERTY	DATA TYPE	DESCRIPTION
ActionProperty	string	A custom property to send to the POS operation.
DisplayText	string	The display text of the notification.

PROPERTY	DATA TYPE	DESCRIPTION
IsLiveContentOnly	bool	A value that indicates whether the notification is only for live content.
IsNew	bool	A value that indicates whether the notification is new.
IsSuccess	bool	A value that indicates whether the notification was successful.
ItemCount	long	The number of notifications.
LastUpdatedDateTime	DateTimeOffset	The date/time when the item in the action property was last updated.
LastUpdatedDateTimeStr	string	The date/time when the item in the action property was last updated, in string format.

## Detailed steps

1. Open `Runtime.Extensions.NotificationSample.proj` from the Retail SDK (`RetailSDK\SampleExtensions\CommerceRuntime\Extensions.NotificationSample`).
2. Rename the project according to the standard naming convention.
3. Inside the project, there is a class file that is named `NotificationExtensionService.cs`. Open this file.
4. The `GetNotificationsExtensionServiceRequest` class is overridden to add a custom notification. Override `GetNotificationsExtensionServiceRequest`, and return `GetNotificationsExtensionServiceResponse`.
5. Either create a new class and override `GetNotificationsExtensionServiceRequest`, or use the sample template.
6. In the `NotificationExtensionService` class, there is a method that is named `Process`. The code inside that method checks the operation ID and then, based on the operation ID, creates a notification details object and adds any notifications. Check whether the operation ID is custom operation ID, and then write logic to check whether there are any notifications. If there are, create a notification object that contains the details, and return it together with the response. The POS will then parse the response and show the notification. The following code example is based on the template.

### NOTE

Remove the sample implementation inside the process method. Keep only the custom logic.

```

namespace Contoso
{
    namespace Commerce.Runtime.NotificationSample
    {
        using System;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;
        /// <summary>
        /// Service class responsible executing the service requests.
        /// </summary>
        public class NotificationExtensionService :
        SingleRequestHandler<GetNotificationsExtensionServiceRequest,
        GetNotificationsExtensionServiceResponse>
        {
            /// <summary>
            /// The handler for the <c>GetNotificationsExtensionServiceRequest</c> request.
            /// </summary>
            /// <param name="request">The request with the operation.</param>
            /// <returns>The notification details for the operation.</returns>
            protected override GetNotificationsExtensionServiceResponse
            Process(GetNotificationsExtensionServiceRequest request)
            {
                ThrowIf.Null(request, "request");
                NotificationDetailCollection details = new NotificationDetailCollection();
                DateTimeOffset lastNotificationDateTime = DateTimeOffset.Now;
                string myOperationId = "5000";
                // do the actual work here
                if ((request.SubscribedOperation).ToString() == myOperationId)
                {
                    NotificationDetail detail = new NotificationDetail()
                    {
                        // Text which will display for the notification detail in the POS notification
                        center
                        DisplayText = "Custom notification",
                        // Number of notifications found
                        ItemCount = 1,
                        // Timestamp of creation of latest notification item (Used to determine whether
                        notification is new)
                        LastUpdatedDateTime = lastNotificationDateTime,
                        // Boolean value representing whether the attempt to get notifications for the
                        given operation was successful
                        IsSuccess = true,
                        // If you would like POS to navigate to a specific action property for the given
                        operation
                        // when the notification tile is selected, define the action property as well.
                        ActionProperty = "1"
                    };
                    details.Add(detail);
                }
                var serviceResponse = new GetNotificationsExtensionServiceResponse(details);
                return serviceResponse;
            }
        }
    }
}

```

7. After you've completed your changes, build the project, and drop the output library into `\RetailServer\webroot\bin\Ext`.
8. Register the output library in the `CommerceRuntime.Ext.config` file.
9. In the POS, create a new operation that has the same operation ID that is used in the CRT extension. In this example, the operation ID is **5000**. You can use any operation ID that is above 5000.
10. When the POS user selects the notification tile, the POS framework calls the operation handler for the

operation ID that is used. Inside the handler, add the required logic to specify what should happen when the POS user selects the notification. For information about how to create a POS operation request, response, and handler, see [Show order notifications in the point of sale \(POS\)](#).

#### **NOTE**

The action property in the Notification detail entity will be sent to the POS operation request. Use that action property to pass any custom information from the notification service to the POS.

11. Configure the notification scheduler according to the instructions in [Show order notifications in the point of sale \(POS\)](#).

## Validate the customization

1. Open the extended Cloud POS or Modern POS application.

The POS triggers the notification service, based on your notification scheduler configuration.

2. Debug the CRT code by attaching the CRT project to w3wp.exe. The breakpoint should be hit whenever the notification service is called from the POS.
3. When the notification is received in the POS, select it. The notification should call the POS operation handler and run the custom logic that is written inside it.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Sign MPOS appx with a code signing certificate

2/18/2021 • 6 minutes to read • [Edit Online](#)

To install Modern POS (MPOS) you must sign the MPOS app with a code signing certificate from a trusted provider and install the same certificate on all the machines where MPOS is installed under the trusted root folder for the current user.

To sign the MPOS app with a certificate, use one of these options in the **Retail SDK\Build tool\Customization.settings** file:

- Add the Secure file task part of Azure DevOps build steps and upload the certificate to secure the file task. Use the secure file task output path variable as a parameter in the Customization.settings file.

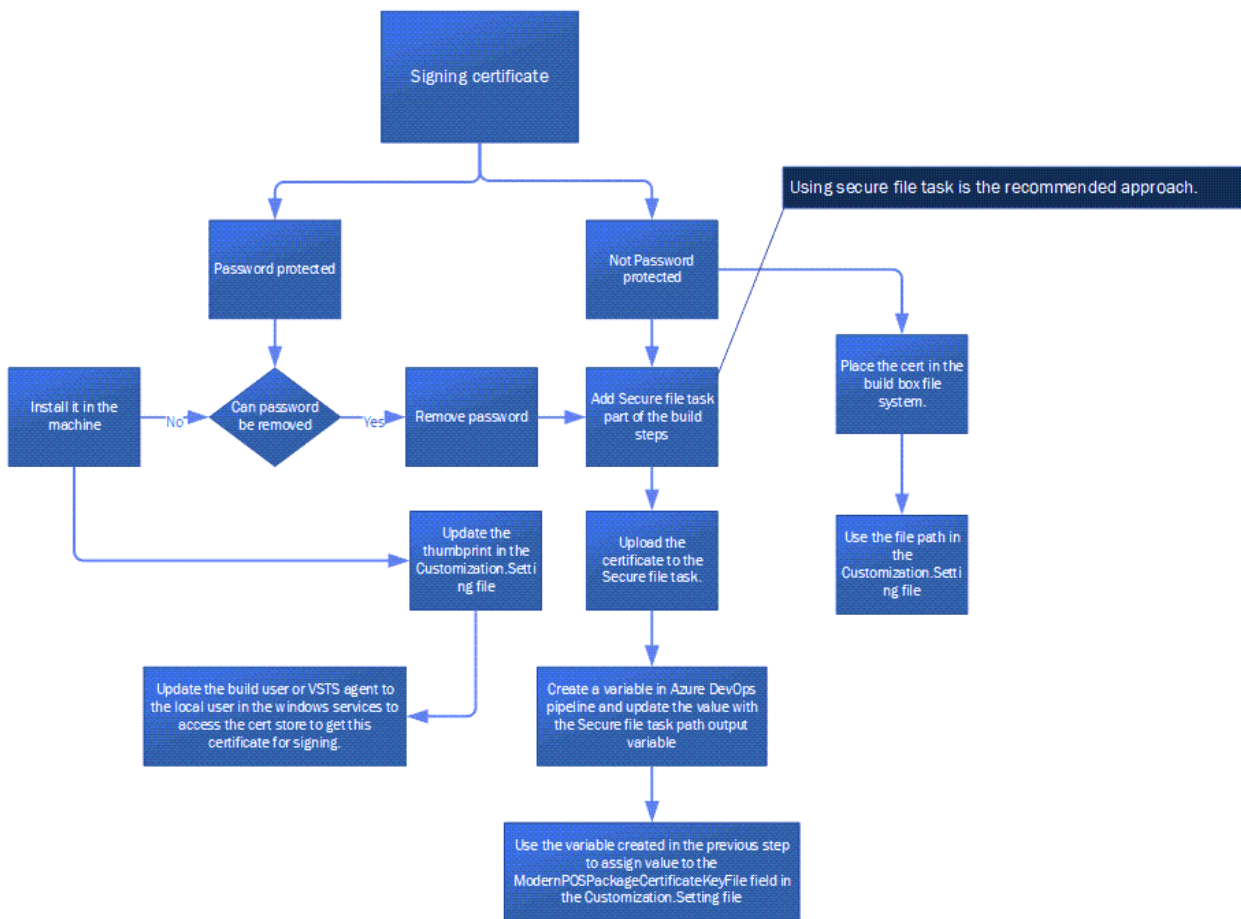
## NOTE

The Secure File task doesn't support a password protected certificate. You must remove the password before uploading this task. Because the certificate is uploaded to the secure file system task in Azure, you can remove the password only for this step. However, you should discuss removing the password with your security experts to determine if this is the correct action for your project. Don't remove the certificate password for other scenarios.

- Use a certificate that is in the file system. To do this, download or generate a certificate and place it in the file system where the build is running. The Microsoft-hosted agent or build user should have access to this path and file.
- Use thumbprint to look up in the certificate in the store and sign in with that certificate.

## Use a Secure File task for Universal Windows Platform app signing

Using a Secure File task is the recommended approach for Universal Windows Platform (UWP) app signing. For more information about package signing, see [Configure package signing](#). This process is shown in the following image.



#### NOTE

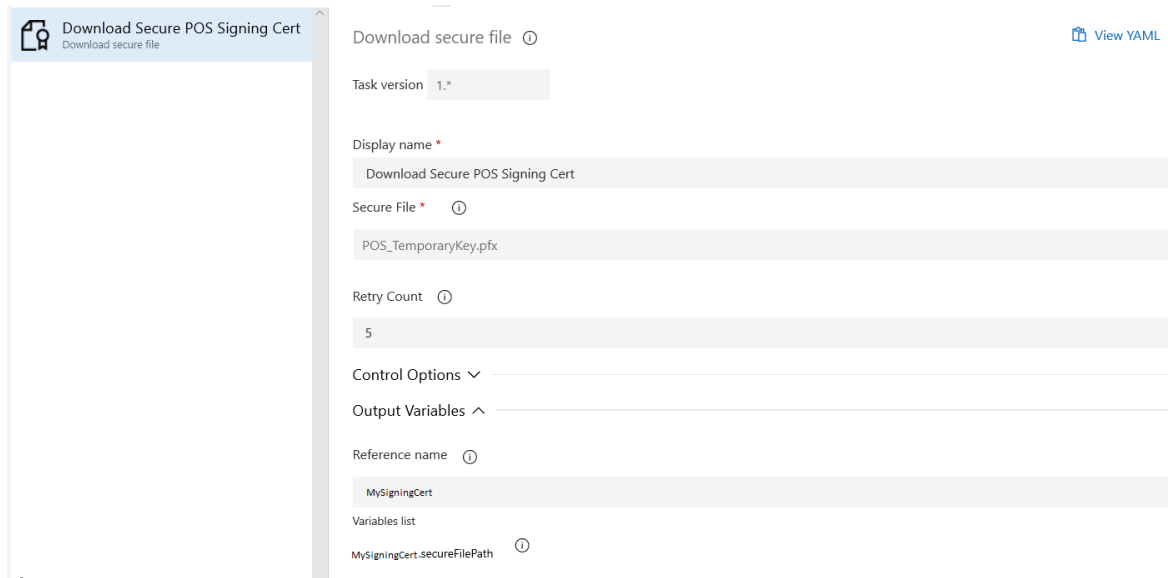
Currently the OOB packaging supports signing only the appx file, the different self-service installers like MPOIS, RSSU, and HWS are not signed by this process. You need to manually sign it using SignTool or other signing tools. The certificate used for signing the appx file must be installed in the machine where Modern POS is installed.

## Steps to configure the certificate for signing

### Certificate in the file system/secure location

Download the [DownloadFile task](#) and add it as the first step in the build process. The advantage of using the Secure File task is that the file is encrypted and placed in the disk during build no matter if the build pipeline succeeds, fails, or is canceled. The file is deleted from the download location after the build process is completed.

1. Download and add the Secure File task as the first step in the Azure DevOps build pipeline. You can download the Secure File task from [DownloadFile](#).
2. Upload the certificate to the Secure File task and set the Reference name under Output Variables, as shown in the following image.



3. Create a new variable in the Azure DevOps pipeline by clicking **New Variable** under the **Variables** tab.
4. Provide a name for the variable in the value field `$(MySigningCert.secureFilePath)`, for example, **CertFile**.
5. Save the variable.
6. Open the `Customization.settings` file from `RetailSDK\BuildTools` and update the `ModernPOSPackageCertificateKeyFile` with the variable name created in the Azure DevOps pipeline (step 3). For example:

```
<ModernPOSPackageCertificateKeyFile Condition="'$(ModernPOSPackageCertificateKeyFile)'
== ''">$(CertFile)</ModernPOSPackageCertificateKeyFile>
```

## Download or generate a certificate to sign the MPOS app

If a downloaded or generated certificate is used to sign the MPOS app, then the update the `ModernPOSPackageCertificateKeyFile` node in the `BuildTools\Customization.settings` file to point to the pfx file location (`$(SdkReferencesPath)\appxsignkey.pfx`). For example:

```
<ModernPOSPackageCertificateKeyFile Condition="'$(ModernPOSPackageCertificateKeyFile)'
== ''">$(SdkReferencesPath)\appxsignkey.pfx</ModernPOSPackageCertificateKeyFile>
```

In this case, the certificate file name is `appxsignkey.pfx`, located in the `Retail SDK\Reference` folder.

## Use thumbprint to sign the MPOS app

If you use thumbprint to sign the MPOS app, then install the certificate locally. Update the thumbprint value in the `ModernPOSPackageCertificateThumbprint` node in the `BuildTools\Customization.settings` file.

This option will work if the build user is a local user. However if you are using the Azure DevOps agents to generate the build, then the agent may not have permission to access the cert store to use the certificate for signing or the build machine will not have the certificate installed. In this case, the workaround is to change the build user to local user and install the certificate in the box. However, this option will not work if you don't have admin access to the box.

## NOTE

If the .pfx file or Secure File task option is used to sign the app, then leave the **ModernPOSPackageCertificateThumbprint** node in **Customization.settings** empty. If the thumbprint option is used, then leave **ModernPOSPackageCertificateKeyFile** empty. If both the values are updated, then the build will fail.

## Certification renewal

### Renew a certificate from trusted CA

Contact your certifying authority (CA) for the certificate renewal process. For a trusted certificate, no action is required on the MPOS side.

### Renew a self-signed certificate

Don't use the sample certificate available in the Retail SDK for production. It can be used only for development purposes. The sample Contoso certificate can't be renewed and the sample certificate included in Retail SDK version 10.0.16 or earlier will expire on December 31, 2020. If this certificate, or a self-signed certificate, has been used to sign a customized Modern POS, there is a strong possibility that Modern POS will not function properly after this date.

### Impact

If the above is true for you, the issue you will be encountering is that the installer will not be able to run after December 31, 2020. Depending on the corporate IT policies used, Modern POS may not be able to function. It is critical that you test this by changing the date temporarily to a future date, to determine the impact to your organization.

### Steps to determine the issue

1. Use Windows settings to change the computer clock to a date and time in the year 2021.
2. Verify that Modern POS can be opened, sign in can occur, and a transaction can be completed.
3. Verify that Modern POS Self-service installer is able to be run, and if so, that installation will complete successfully.
4. Return the Windows clock settings to the correct date and time.

If you can complete all of these steps without issues, then you will be able to operate on the current certificate past December 31, 2020.

### Steps going forward

It is highly recommended that you renew the previously used certificate. We strongly recommend that you obtain a new certificate. To do this, you must perform one of the following actions:

- **Preferred** - Obtain a code signing certificate from a trusted certificate authority.
- **Preferred** - Generate a self-signed code signing certificate to use. This is typically used when within a domain.
- **Available as a temporary Solution** - Use the renewed Contoso code signing certificate. This is typically used for testing purposes, so it's not recommended that it be deployed in production.

Next, generate a new customized Modern POS package that is signed using this certificate obtained from one of the actions above. Depending on the certificate, one of the following steps must be followed:

- If using a new, trusted certificate (or a new, self-signed certificate), you will be required to install a new certificate on every device. After that, you need to take the newly created Modern POS Package (installer), uninstall the existing application, and then reinstall the new Modern POS package. You will need to perform a device activation of Modern POS on every device.

- If using the renewed Contoso certificate, you will be required to install the new certificate on every device and install the Modern POS Package (installer). You are not required to uninstall, however you must reinstall on the device. Note that device activation of Modern POS will not be required. This option is a temporary solution. Only use this option to avoid reactivation and resolve the issue before obtaining a new trusted certificate.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Extend POS views to add custom columns and app bar buttons

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains how you can extend existing point of sale (POS) views. To extend the **Transaction** screen and **Welcome** screen, you can use the screen layout designer. To extend all other POS views, such as the **Customer Add/Edit** screen, you use the Retail software development kit (SDK). This topic focuses on the extension of existing POS views via the Retail SDK.

POS views support the following extension points and patterns:

- **Custom app bar buttons** – Add custom buttons to the app bar on selected pages.
- **Custom column sets** – Replace the grid columns with custom columns on selected pages.
- **Custom controls** – Add new controls to selected pages.

## POS views that currently support extensions

The following table shows the POS views that currently support extensions. It also indicates the types of extension points that each POS view supports.

### NOTE

The upcoming releases and hotfix will add support for more extension points in other views.

POS VIEW (RELEASE)	CUSTOM CONTROLS ARE SUPPORTED	CUSTOM COLUMNS ARE SUPPORTED	CUSTOM APP BAR BUTTONS ARE SUPPORTED
Cart view (Screen layout based)	Yes	Yes	No
CustomerAddEditView	Yes	No	Yes
CustomerDetailsView	Yes	No	Yes
SearchView	No	Yes	Yes
InventoryLookupView	No	Yes	Yes
ShowJournalView	No	Yes	Yes
SimpleProductDetailsView	Yes	No	Yes
AddressAddEditView	Yes	No	Yes
PaymentView	No	No	Yes
PriceCheckView	Yes	No	Yes

POS VIEW (RELEASE)	CUSTOM CONTROLS ARE SUPPORTED	CUSTOM COLUMNS ARE SUPPORTED	CUSTOM APP BAR BUTTONS ARE SUPPORTED
PriceCheckViewPhone	No	No	Yes
SearchOrdersView	No	Yes	No
SearchPickingAndReceivingView	No	Yes	Yes
CustomerOrderHistoryView	No	Yes	No
SearchStockCountView	No	Yes	No
StockCountDetailsView	No	Yes	Yes
ResumeCartView	No	Yes	Yes
InventoryLookupMatrixView	No	No	Yes
SuspendTransactionView	No	Yes	No
ManageShiftView	No	No	Yes
ReportDetailsView	No	No	Yes
SearchReceiptsView	No	No	Yes
TransferOrderDetailsView	No	No	Yes
FulfillmentLineView	No	Yes	Yes
ReturnTransactionView	No	Yes	Yes
PickingAndReceivingDetailsView	No	Yes	Yes
PickingAndReceivingDetailsView (Advanced warehouse)	No	Yes	Yes
SalesInvoiceDetailsView (10.0.11)	No	No	Yes
SalesInvoicesView (10.0.11)	No	Yes	No
InventoryDocumentShippingAndReceivingView (10.0.13)	No	No	Yes
InventoryDocumentListView	No	Yes (10.0.15)	Yes (10.0.13)

#### NOTE

The table shown above is updated based on the latest released version and hotfix. In earlier versions, some of these extension points will not be available.

#### NOTE

In Show journal (lines grid) and Return transaction view custom columns are supported using the row sub fields. These sub fields will be displayed as rows instead of columns, like the info code messages or serial number or discounts values.

Filter extensions are also supported in **Show journal view** and **Search order views** to add custom filters. **Search order views** also supports setting default parameters for search in the user interface (UI) using extension. For example, if you want to add default store search parameter you can do that by using extension and showing that in the UI.

## Add a custom column and an app bar button

1. Start Microsoft Visual Studio 2015 as an administrator.
2. Open the **ModernPOS** solution from ...\**RetailSDK\POS**.
3. In the **POS.Extensions** project, create a folder that is named **SearchExtension**.
4. In the **SearchExtension** folder, create a folder that is named **ViewExtensions**.
5. In the **ViewExtensions** folder, create a folder that is named **Search**.
6. In the **Search** folder, create a Typescript file that is named **CustomCustomerSearchColumns.ts**.
7. In the **CustomCustomerSearchColumns.ts** file, add the following **import** statements to import the relevant entities and context.

```
import { ICustomerSearchColumn } from "PosApi/Extend/Views/SearchView";  
import { ICustomColumnsContext } from "PosApi/Extend/Views/CustomListColumns";  
import { ProxyEntities } from "PosApi/Entities";
```

8. Add the existing column and the custom column to the file.



```

export default (context: ICustomColumnsContext): ICustomerSearchColumn[] => {
  return [
    {
      title: context.resources.getString("string_2"),
      computeValue: (row: ProxyEntities.GlobalCustomer): string => { return row.AccountNumber;
    },
    ratio: 15,
    collapseOrder: 5,
    minWidth: 120
  }, {
    title: context.resources.getString("string_3"),
    computeValue: (row: ProxyEntities.GlobalCustomer): string => { return row.FullName; },
    ratio: 20,
    collapseOrder: 4,
    minWidth: 200
  }, {
    title: context.resources.getString("string_4"),
    computeValue: (row: ProxyEntities.GlobalCustomer): string => { return row.FullAddress; },
    ratio: 25,
    collapseOrder: 1,
    minWidth: 200
  }, {
    title: context.resources.getString("string_5"),
    computeValue: (row: ProxyEntities.GlobalCustomer): string => { return row.Email; },
    ratio: 20,
    collapseOrder: 2,
    minWidth: 200
  }, {
    title: context.resources.getString("string_7"),
    computeValue: (row: ProxyEntities.GlobalCustomer): string => { return row.Phone; },
    ratio: 20,
    collapseOrder: 3,
    minWidth: 120
  }
  ];
};

```

9. You will now add the resource file for localization of the column name. In the **SearchExtension** folder, create a folder that is named **Resources**.
10. In the **Resources** folder, create a folder that is named **Strings**.
11. In the **Strings** folder, create a folder that is named **en-US**.
12. In the **en-us** folder, create a file that is named **resources.resjson**.
13. In the **resources.resjson** file, add the following code.

```

{
  //===== Sample View extensions strings. =====
  "string_0" : "Quick compare products",
  "_string_0.comment" : "Product search page app bar command label.",
  "string_1" : "View customer summary",
  "_string_1.comment" : "Customer search page app bar command label.",
  //===== Column names. =====
  "string_2" : "ACCOUNT NUMBER_CUSTOMIZED",
  "_string_2.comment" : "Customer search column name.",
  "string_3" : "NAME",
  "_string_3.comment" : "Customer search column name.",
  "string_4" : "ADDRESS",
  "_string_4.comment" : "Customer search column name.",
  "string_5" : "CONTACT EMAIL",
  "_string_5.comment" : "Customer search column name.",
  "string_7" : "PHONE NUMBER",
  "_string_7.comment" : "Customer search column name."
}

```

14. In the **SearchExtension** folder, create a folder that is named **DialogSample**.
15. In the **DialogSample** folder, create a TypeScript file that is named **MessageDialog.ts**.
16. In the **MessageDialog.ts** file, add the following **import** statements to import the relevant entities and context.

```

import { ShowMessageDialogClientRequest, ShowMessageDialogClientResponse, IMessageDialogOptions }
from "PosApi/Consume/Dialogs";
import { IExtensionContext } from "PosApi/Framework/ExtensionContext";
import { ClientEntities } from "PosApi/Entities";

```

17. Create a class that is named **MessageDialog**.

```

export default class MessageDialog {}

```

18. In the **MessageDialog** class, add the following **show** method.

```

public static show(context: IExtensionContext, message: string): Promise<void> {
    let promise: Promise<void> = new Promise<void>((resolve: () => void, reject: (reason?: any) =>
void) =>
    {
        let messageDialogOptions: IMessageDialogOptions = {
            title: "Extension Message Dialog",
            message: message,
            showCloseX: true, // this property will return "Close" as result when "X" is clicked to
close dialog.
            button1: {
                id: "Button1Close",
                label: "OK",
                result: "OKResult"
            },
            button2: {
                id: "Button2Cancel",
                label: "Cancel",
                result: "CancelResult"
            }
        };
        let dialogRequest: ShowMessageDialogClientRequest<ShowMessageDialogClientResponse> =
new ShowMessageDialogClientRequest<ShowMessageDialogClientResponse>
(messageDialogOptions);
        context.runtime.executeAsync(dialogRequest).then((
            result: ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>) => {
            if (!result.canceled) {
                context.logger.logInformational("MessageDialog result: " +
result.data.result.dialogResult);
                resolve();
            }
        }).catch((reason: any) => {
            context.logger.logError(JSON.stringify(reason));
            reject(reason);
        });
    });
    return promise;
}

```

19. You will now add a custom app bar button in the search view to open a dialog box that contains details about the selected customer. In the **ViewExtensions** folder, create a Typescript file that is named **ViewCustomerSummaryCommand.ts**.
20. In the **ViewCustomerSummaryCommand.ts** file, add the following **import** statements to import the relevant entities and context.

```

import { ProxyEntities } from "PosApi/Entities";
import { ArrayExtensions, ObjectExtensions } from "PosApi/TypeExtensions";
import { IExtensionCommandContext } from "PosApi/Extend/Views/AppBarCommands";
import * as SearchView from "PosApi/Extend/Views/SearchView";
import MessageDialog from "../../Controls/DialogSample/MessageDialog";

```

21. Create a class that is named **ViewCustomerSummaryCommand**, and extend it from **CustomerSearchExtensionCommandBase**.

```

export default class ViewCustomerSummaryCommand extends SearchView.CustomerSearchExtensionCommandBase
{}

```

22. In the **ViewCustomerSummaryCommand** class, declare a private variable to capture the results when searching for the selected customer.

```
private _customerSearchResults: ProxyEntities.GlobalCustomer[];
```

23. Add the class **constructor** method to initialize and clear the search handler.

```
constructor(context:
IExtensionCommandContext<SearchView.ICustomerSearchToExtensionCommandMessageTypeMap>) {
    super(context);
    this.id = "viewCustomerSummaryCommand";
    this.label = context.resources.getString("string_1");
    this.extraClass = "iconLightningBolt";
    this._customerSearchResults = [];
    this.searchResultsSelectedHandler = (data: SearchView.CustomerSearchSearchResultSelectedData):
void => {
    this._customerSearchResults = data.customers;
    this.canExecute = true;
};
this.searchResultSelectionClearedHandler = (): void => {
    this._customerSearchResults = [];
    this.canExecute = false;
};
}
```

24. Add the **init** method to initialize the **visible** property.

```
protected init(state: SearchView.ICustomerSearchExtensionCommandState): void {
    this.isVisible = true;
}
```

25. Add the **execute** method to handle the app button click handler. The **execute** method reads the data for the selected customer from the handler and shows it in a simple dialog box.

```
protected execute(): void {
    let customer: ProxyEntities.GlobalCustomer =
ArrayExtensions.firstOrUndefined(this._customerSearchResults);
    if (!ObjectExtensions.isNullOrUndefined(customer)) {
        let message: string = "Customer Account: " + (customer.AccountNumber || "") + " | ";
        message += "Name: " + customer.FullName + " | ";
        message += "Phone Number: " + customer.Phone + " | ";
        message += "Email Address: " + customer.Email;
        MessageDialog.show(this.context, message);
    }
}
```

The whole code sample should look like this.

```

import { ProxyEntities } from "PosApi/Entities";
import { ArrayExtensions, ObjectExtensions } from "PosApi/TypeExtensions";
import { IExtensionCommandContext } from "PosApi/Extend/Views/AppBarCommands";
import * as SearchView from "PosApi/Extend/Views/SearchView";
import MessageDialog from "../../DialogSample/MessageDialog";
export default class ViewCustomerSummaryCommand extends SearchView.CustomerSearchExtensionCommandBase
{
    private _customerSearchResults: ProxyEntities.GlobalCustomer[];

    /**
     * Creates a new instance of the ViewCustomerSummaryCommand class.
     * @param
     {IExtensionCommandContext<CustomerDetailsView.ICustomerSearchToExtensionCommandMessageTypeMap>}
    context The command context.
     * @remarks The command context contains APIs through which a command can communicate with POS.
     */
    constructor(context:
    IExtensionCommandContext<SearchView.ICustomerSearchToExtensionCommandMessageTypeMap>) {
        super(context);
        this.id = "viewCustomerSummaryCommand";
        this.label = context.resources.getString("string_1");
        this.extraClass = "iconLightningBolt";
        this._customerSearchResults = [];

        this.searchResultsSelectedHandler = (data:
    SearchView.CustomerSearchSearchResultSelectedData): void => {
            this._customerSearchResults = data.customers;
            this.canExecute = true;
        };

        this.searchResultSelectionClearedHandler = (): void => {
            this._customerSearchResults = [];
            this.canExecute = false;
        };
    }

    /**
     * Initializes the command.
     * @param {CustomerDetailsView.ICustomerDetailsExtensionCommandState} state The state used to
    initialize the command.
     */
    protected init(state: SearchView.ICustomerSearchExtensionCommandState): void {
        this.isVisible = true;
    }

    /**
     * Executes the command.
     */
    protected execute(): void {
        let customer: ProxyEntities.GlobalCustomer =
    ArrayExtensions.firstOrUndefined(this._customerSearchResults);
        if (!ObjectExtensions.isNullOrUndefined(customer)) {
            let message: string = "Customer Account: " + (customer.AccountNumber || "") + " | ";
            message += "Name: " + customer.FullName + " | ";
            message += "Phone Number: " + customer.Phone + " | ";
            message += "Email Address: " + customer.Email;
            MessageDialog.show(this.context, message);
        }
    }
}

```

26. In the **SearchExtension** folder, create a JSON file that is named **manifest.json**.

27. In the **manifest.json** file, add the following code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "resources": {
      "supportedUICultures": [ "en-US" ],
      "fallbackUICulture": "en-US",
      "culturesDirectoryPath": "Resources/Strings ",
      "stringResourcesFileName": "resources.resjson",
      "cultureInfoOverridesFilePath": "Resources/cultureInfoOverrides.json"
    },
    "extend": {
      "views": {
        "SearchView": {
          "customerAppBarCommands": [ { "modulePath":
"ViewExtensions/Search/ViewCustomerSummaryCommand" } ],
          "customerListConfiguration": { "modulePath":
"ViewExtensions/Search/CustomCustomerSearchColumns" }
        }
      }
    }
  }
}

```

28. In the **POS.Extensions** project, open the **extensions.json** file, and update it with **SearchExtension** samples, so that the POS includes this extension at runtime.

```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "SearchExtension"
    }
  ]
}

```

29. In the **tsconfig.json** file, comment out the extension package folders in the exclude list. The POS uses this file to include or exclude the extension. By default, the list contains the whole excluded extensions list. To include an extension as part of the POS, add the name of the extension folder, and comment out the extension in the exclude list, as shown here.

```

"exclude": [
  "SampleExtensions"
  //"SampleExtensions2",
  //"SearchExtension"
],

```

30. Compile and rebuild the project.

## Validate the customization

Follow these steps to validate the customization.

1. Sign in to Modern POS by using **000160** as the operator ID and **123** as the password.

2. Search for customer **2001** by using the search bar on the top.

You should see the custom columns that you added.

3. Select a customer, and then select the new app bar button. A dialog box should appear that contains details about the selected customer.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add custom controls to Modern POS (MPOS) transaction pages

2/18/2021 • 7 minutes to read • [Edit Online](#)

You can add more information to a transaction page by using custom controls. You can add a custom control to a transaction page by using the screen layout designer. In the designer, you can use a drag-and-drop operation to add the custom control, and then set the location, height, and width of the control. You can implement business logic for the custom control in your own extensions by using the POS extension framework. This topic explains how to add a new custom control that shows the details for the selected line item, the item ID, and the description.

## NOTE

This topic applies to Dynamics 365 for Finance and Operations, and to Microsoft Dynamics 365 Retail with platform update 8 and Retail App update 4 hotfix.

## Add a new custom control

1. Sign in to Dynamics 365 Commerce.
2. Select **Retail > Channel setup > POS setup > POS > Screen layouts**.
3. Select the **F3MGR** screen layout ID, and then, on the Action Pane, select **Designer**.
4. Select **1440x960 – Full layout** as the layout size, and then select **Layout designer**.
5. If you're prompted to install the designer tool, select **Open**, and follow the installation instructions.
6. When you're prompted for your Microsoft Azure Active Directory (Azure AD) credentials, enter the information to start the designer.
7. In the designer, drag the custom control from the left pane to the page, and then adjust, resize, or reposition the custom control as you require.
8. On the page, right-click the custom control, and then select **Customize**.
9. In the dialog box, set these properties:
  - **Control Name:** lineDetails
  - **Package Name:** Pos\_Extensibility\_Samples
  - **Publisher Name:** Contoso

## NOTE

These names should match the names in the extension manifest.

10. Close the designer by selecting the **Close** button (X).
11. When you're prompted to save your changes, select **Yes**. If you select **No**, your changes won't be saved.
12. Select **Retail > Retail IT > Distribution schedule**.



13. Select the **Registers (1090)** job, and then select **Run now**.

## Add business logic to the custom control

1. Start Microsoft Visual Studio 2015 as an administrator.
2. Open the **ModernPOS** solution from ...\**RetailSDK\POS**.
3. In the **POS.Extensions** project, create a folder that is named **CustomControlExtensions**.
4. In the **CustomControlExtensions** folder, create a folder that is named **Cart**.
5. In the **Cart** folder, create a Typescript file that is named **CartViewController.ts**.
6. In the **CartViewController.ts** file, add the following **import** statement to import the relevant entities and context.

```
import { ProxyEntities } from "PosApi/Entities";
import { IExtensionCartViewControllerContext } from "PosApi/Extend/Views/CartView";
import * as CartView from "PosApi/Extend/Views/CartView";
```

7. Create a class that is named **CartViewController**, and extend it from **CartExtensionViewControllerBase**. The **CartExtensionViewControllerBase** class contains the cart and tender lines, the cart line selected handler, and the cart line cleared handler. These elements will be used to show the selected line in the custom control.

```
export default class CartViewController extends CartView.CartExtensionViewControllerBase {
}
```

8. In the **CartViewController** class, add two private variables to get the selected cart lines and tender lines.

```
private _selectedCartLines: ProxyEntities.CartLine[];

private _selectedTenderLines: ProxyEntities.TenderLine[];
```

9. Create a class **constructor** method to set the cart and tender selection information.

```
constructor(context: IExtensionCartViewControllerContext) {
  super(context);
  this.cartLineSelectedHandler = (data: CartView.CartLineSelectedData): void => {
    this._selectedCartLines = data.cartLines;
  };

  this.cartLineSelectionClearedHandler = (): void => {
    this._selectedCartLines = undefined;
  };

  this.tenderLineSelectedHandler = (data: CartView.TenderLineSelectedData): void => {
    this._selectedTenderLines = data.tenderLines;
  };

  this.tenderLineSelectionClearedHandler = (): void => {
    this._selectedCartLines = undefined;
  };
}
```

The overall class should look like this.

```

import { ProxyEntities } from "PosApi/Entities";
import { IExtensionCartViewControllerContext } from "PosApi/Extend/Views/CartView";
import * as CartView from "PosApi/Extend/Views/CartView";

export default class CartViewController extends CartView.CartExtensionViewControllerBase {
  private _selectedCartLines: ProxyEntities.CartLine[];
  private _selectedTenderLines: ProxyEntities.TenderLine[];

  /**
   * Creates a new instance of the CartViewController class.
   * @param {IExtensionCartViewControllerContext} context The events Handler context.
   * @remarks The events handler context contains APIs through which a handler can communicate with
   POS.
   */
  constructor(context: IExtensionCartViewControllerContext) {
    super(context);
    this.cartLineSelectedHandler = (data: CartView.CartLineSelectedData): void => {
      this._selectedCartLines = data.cartLines;
    };

    this.cartLineSelectionClearedHandler = (): void => {
      this._selectedCartLines = undefined;
    };

    this.tenderLineSelectedHandler = (data: CartView.TenderLineSelectedData): void => {
      this._selectedTenderLines = data.tenderLines;
    };

    this.tenderLineSelectionClearedHandler = (): void => {
      this._selectedCartLines = undefined;
    };
  }
}

```

10. In the **Cart** folder, create an HTML file that is named **LineDetailsCustomControl.html**.

11. In the **LineDetailsCustomControl.html** file, add two text fields to show the ID and description for the selected line item. Delete the default code, and add the following code.

```

<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
    <!-- Note: The element ID differs from the ID generated by the POS extensibility
framework. This 'template' ID is not used by the POS extensibility framework. -->
    <script id="Microsoft_Pos_Extensibility_Samples_LineDetails" type="text/html">
      <!-- ko ifnot: isCartLineSelected -->
      <h4>No cart lines selected</h4>
      <!-- /ko -->
      <!-- ko if: isCartLineSelected -->
      <h4 data-bind="text: cartLineItemId">Item ID</h4>
      <h4 data-bind="text: cartLineDescription">Item ID</h4>
      <!-- /ko -->;
    </script>
  </body>
</html>
...

```

12. In the **Cart** folder, create a Typescript file that is named **LineDetailsCustomControl.ts**.

13. In the `LineDetailsCustomControl.ts` file, add the logic to bind the line details information.
14. Import the POS entities and type extensions to use the reference type in the constructor and other events.

```
import {
  ObjectExtensions,
  StringExtensions,
  ArrayExtensions
} from "PosApi/TypeExtensions";
import { ProxyEntities } from "PosApi/Entities";
```

15. Create a class, and extend it from `CartViewCustomControlBase`.

```
export default class LineDetailsCustomControl extends CartViewCustomControlBase {}
```

16. Declare the following private variables to set the cart item ID and description.

```
private static readonly TEMPLATE_ID: string = "Microsoft_Pos_Extensibility_Samples_LineDetails";
public readonly cartLineItemId: Computed<string>;
public readonly cartLineDescription: Computed<string>;
public readonly isCartLineSelected: Computed<boolean>;
private readonly _cartLine: Observable<ProxyEntities.CartLine>;
private _state: ICartViewCustomControlState;
```

17. Create the **constructor** method to initialize and get the selected handler.

```
constructor(id: string, context: ICartViewCustomControlContext) {
  super(id, context);
  this._cartLine = ko.observable(null);
  this.cartLineItemId = ko.computed(() => {
    let cartLine: ProxyEntities.CartLine = this._cartLine();
    if (!ObjectExtensions.isNullOrUndefined(cartLine)) {
      return cartLine.ItemId;
    }
    return StringExtensions.EMPTY;
  });

  this.cartLineDescription = ko.computed(() => {
    let cartLine: ProxyEntities.CartLine = this._cartLine();
    if (!ObjectExtensions.isNullOrUndefined(cartLine)) {
      return cartLine.Description;
    }
    return StringExtensions.EMPTY;
  });

  this.isCartLineSelected = ko.computed(() =>
!ObjectExtensions.isNullOrUndefined(this._cartLine()));
  this.cartLineSelectedHandler = (data: CartLineSelectedData) => {
    if (ArrayExtensions.hasElements(data.cartLines)) {
      this._cartLine(data.cartLines[0]);
    }
  };

  this.cartLineSelectionClearedHandler = () => {
    this._cartLine(null);
  };
}
```

18. Add the **onReady** method to bind the control to the specified HTML element.

```

public onReady(element: HTMLElement): void {
    ko.applyBindingsToNode(element, {
        template: {
            name: LineDetailsCustomControl.TEMPLATE_ID,
            data: this
        }
    });
}

```

19. Add the `init` method to set the state.

```

public init(state: ICartViewCustomControlState): void {
    this._state = state;
}

```

The overall class should look like this.

```

import {
    CartViewCustomControlBase,
    ICartViewCustomControlState,
    ICartViewCustomControlContext,
    CartLineSelectedData
} from "PosApi/Extend/Views/CartView";

import {
    ObjectExtensions,
    StringExtensions,
    ArrayExtensions
} from "PosApi/TypeExtensions";

import { ProxyEntities } from "PosApi/Entities";

export default class LineDetailsCustomControl extends CartViewCustomControlBase {
    private static readonly TEMPLATE_ID: string = "Microsoft_Pos_Extensibility_Samples_LineDetails";
    public readonly cartLineItemId: Computed<string>;
    public readonly cartLineDescription: Computed<string>;
    public readonly isCartLineSelected: Computed<boolean>;
    private readonly _cartLine: Observable<ProxyEntities.CartLine>;
    private _state: ICartViewCustomControlState;

    constructor(id: string, context: ICartViewCustomControlContext) {
        super(id, context);
        this._cartLine = ko.observable(null);

        this.cartLineItemId = ko.computed(() => {
            let cartLine: ProxyEntities.CartLine = this._cartLine();
            if (!ObjectExtensions.isNullOrUndefined(cartLine)) {
                return cartLine.ItemId;
            }
            return StringExtensions.EMPTY;
        });

        this.cartLineDescription = ko.computed(() => {
            let cartLine: ProxyEntities.CartLine = this._cartLine();
            if (!ObjectExtensions.isNullOrUndefined(cartLine)) {
                return cartLine.Description;
            }
            return StringExtensions.EMPTY;
        });

        this.isCartLineSelected = ko.computed(() =>
!ObjectExtensions.isNullOrUndefined(this._cartLine()));
        this.cartLineSelectedHandler = (data: CartLineSelectedData) => {
            if (ArrayExtensions.hasElements(data.cartLines)) {

```

```

        this._cartLine(data.cartLines[0]);
    }
};

this.cartLineSelectionClearedHandler = () => {
    this._cartLine(null);
};
}

/**
 * Binds the control to the specified element.
 * @param {HTMLElement} element The element to which the control should be bound.
 */
public onReady(element: HTMLElement): void {
    ko.applyBindingsToNode(element, {
        template: {
            name: LineDetailsCustomControl.TEMPLATE_ID,
            data: this
        }
    });
}

/**
 * Initializes the control.
 * @param {ICartViewCustomControlState} state The initial state of the page used to initialize the control.
 */
public init(state: ICartViewCustomControlState): void {
    this._state = state;
}
}
}

```

20. In the **CustomControlExtensions** folder, create a JSON file that is named **manifest.json**.

21. In the **manifest.json** file, add the following code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Contoso",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "extend": {
      "views": {
        "CartView": {
          "viewController": { "modulePath": "Cart/CartViewController" },
          "controlsConfig": {
            "customControls": [
              {
                "controlName": "lineDetails",
                "htmlPath": "Cart/LineDetailsCustomControl.html",
                "modulePath": "Cart/LineDetailsCustomControl"
              }
            ]
          }
        }
      }
    }
  }
}

```

22. In the **POS.Extensions** project, open the **extensions.json** file, and update it with the following **CustomControlExtensions** samples, so that the POS includes this extension at runtime.

```
{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "CustomControlExtensions"
    }
  ]
}
```

23. In the `tsconfig.json` file, comment out the extension package folders in the exclude list. The POS uses this file to include or exclude the extension. By default, the list contains the whole excluded extensions list. To include an extension as part of the POS, add the name of the extension folder, and comment out the extension in the exclude list, as shown here.

```
"exclude": [
  "SampleExtensions"
  // "SampleExtensions2",
  // "CustomControlExtensions"
],
```

24. Compile and rebuild the project.

## Validate the customization

1. Sign in to Modern POS by using **000160** as the operator ID and **123** as the password.
2. On the **Welcome** screen, select the **Current transaction** button.
3. Add any item to the transaction, and then select the line item that you added.

The custom control should show the ID and description for the selected line item.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add custom columns to a point of sale (POS) transaction grid

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how to add a new custom column to a POS transaction page using the screen layout designer. You can add more information to a transaction page by using the custom column feature. A custom column can be added to the transaction page receipt grid by using the screen layout designer. You can adjust the width and position of the columns by using the designer. There are 10 custom columns in the layout for extensions scenarios. You can use all 10 in one layout. The custom columns are already added to the designer metadata. After adding the column to the layout, you run the distribution job so that the column shows up on the transaction page.

## NOTE

This topic applies to Dynamics 365 for Finance and Operations, and to Microsoft Dynamics 365 Retail with platform update 8 and Retail App update 4 hotfix.

## Add a custom column to the page

1. Sign in to Dynamics 365 Commerce.
2. Navigate to **Retail and Commerce > Channel setup > POS setup > POS > Screen layouts**. Or, search for **Screen layout** in the search bar.
3. Select the **F3MGR** screen layout ID and click the **Designer** button in the action bar.
4. Follow the instructions if prompted to install and enter the Azure Active Directory (AAD) credentials to launch the designer.
5. Select **1440x960 – Full layout** from the layout sizes and click the **Layout designer** button.
6. If prompted, click **Open** and follow the instruction to install the designer tool.
7. After installing, enter your AAD credentials to launch the designer.
8. In the designer, right-click the transaction grid (receipt grid) and select **Customize**.
9. In the **Customization – Receipt** window, select the **lines** in the pivot panel drop-down menu.

## NOTE

Similarly, you can add a custom column to the **Payment and Delivery** tab.

10. In the **Available columns** window, select **Custom column 1**, and then click the **> (arrow)** button to move the column to the **Selected** columns.
11. Click **OK** to save and close the window.
12. Adjust the column width in the transaction grid using the **Screen layout** designer. Make sure the column is visible.
13. Click the **X** button in the designer to close the designer.

14. When prompted to **Save changes**, click **Yes**. If you click **No** the changes will not be saved.
15. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
16. Select the **Registers (1090)** job and click **Run now**.

## Add business logic to a custom column

1. Open Visual Studio 2015 in administrator mode.
2. Open the **ModernPOS** solution from `...\RetailSDK\POS`.
3. Under the **POS.Extensions** project create a new folder named **CustomColumnExtensions**.
4. Under **CustomColumnExtensions**, create a new folder named **Cart**.
5. Under **Cart**, create a new folder named **LinesGrid**.
6. In the **LinesGrid** folder, add a new Typescript file and name it **CustomColumn1Configuration.ts**.
7. Add the following **import** statements to import the relevant entities and context.

```
import {  
  
    ICustomLinesGridColumnContext,  
    CustomLinesGridColumnBase  
  
} from "PosApi/Extend/Views/CartView";  
  
import { CustomGridColumnAlignment } from "PosApi/Extend/Views/CustomGridColumnns";  
import { ProxyEntities } from "PosApi/Entities";
```

8. Create a new class named **LinesCustomGridColumn1** and extend it from **CustomLinesGridColumnBase**.

```
export default class LinesCustomGridColumn1 extends CustomLinesGridColumnBase {}
```

9. Inside the class declare a private variable to capture the selected tender lines.

```
private _selectedTenderLines: ProxyEntities.TenderLine[ ];
```

10. Create a class constructor method to initialize the context.

```
constructor(context: ICustomLinesGridColumnContext) {  
    super(context);  
}
```

11. Add the following methods for the columns title and alignment.

```
public title(): string {  
    return "Line number";  
}  
  
public alignment(): CustomGridColumnAlignment {  
    return CustomGridColumnAlignment.Right;  
}
```



12. Add the column compute value method, which returns the line number.

```
public computeValue(cartLine: ProxyEntities.CartLine): string {
    return cartLine.LineNumber.toString();
}
```

The code for the entire class is:

```
import {
    ICustomLinesGridColumnContext,
    CustomLinesGridColumnBase
} from "PosApi/Extend/Views/CartView";
import { CustomGridColumnAlignment } from "PosApi/Extend/Views/CustomGridColumns";
import { ProxyEntities } from "PosApi/Entities";

export default class LinesCustomGridColumn1 extends CustomLinesGridColumnBase {
    constructor(context: ICustomLinesGridColumnContext) {
        super(context);
    }

    public title(): string {
        return "Line number";
    }

    public computeValue(cartLine: ProxyEntities.CartLine): string {
        return cartLine.LineNumber.toString();
    }

    public alignment(): CustomGridColumnAlignment {
        return CustomGridColumnAlignment.Right;
    }
}
```

13. Create a new .json file under the **CustomColumnExtensions** folder and name it **manifest.json**.

14. In the **manifest.json** file, replace the generated code with the following code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "extend": {
      "views": {
        "CartView": {
          "linesGrid": {
            "customColumn1": { "modulePath": "Cart/LinesGrid/CustomColumn1Configuration"
          }
        }
      }
    }
  }
}

> [!NOTE]
> If you are adding a custom column to payment or delivery grid, you need to update the manifest with
the following code.
"paymentsGrid": {
  "customColumn1": { "modulePath": "Cart/PaymentsGrid/CustomColumn1Configuration" }
},
"deliveryGrid": {
  "customColumn1": { "modulePath": "Cart/DeliveryGrid/CustomColumn1Configuration" }
}

```

- Open the **extensions.json** file under the **POS.Extensions** project and update it with the **CustomColumnExtensions** sample, so that POS during runtime will include this extension.

```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "CustomColumnExtensions"
    }
  ]
}

```

- Open the **tsconfig.json** file and comment out the extension package folders from the exclude list. POS will use this file to include or exclude the extension. By default, the list contains all the excluded extensions. If you want to include any extension part of the POS, then you need add the extension folder name and comment the extension from the extension list as shown.

```
"exclude": [  
  "AuditEventExtensionSample",  
  "B2BSample",  
  "CustomerSearchWithAttributesSample",  
  "FiscalRegisterSample",  
  "PaymentSample",  
  "PromotionsSample",  
  "SalesTransactionSignatureSample",  
  //"SampleExtensions2",  
  "SampleExtensions",  
  "StoreHoursSample",  
  "SuspendTransactionReceiptSample"  
  //"CustomColumnExtensions"  
],
```

17. Compile and rebuild the project.

#### NOTE

You can find the sample for the custom column in the [Retail software development kit \(SDK\) architecture](#).

## Validate the customization

1. Sign in to MPOS using **000160** as the operator ID and **123** as the password.
2. Click the **Current transaction** button on the **Welcome** screen.
3. Add item (**0005**) to the transaction.
4. The custom column should display the line number.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Call point of sale (POS) APIs or operations from POS extensions

2/18/2021 • 6 minutes to read • [Edit Online](#)

The Retail POS APIs help you to build extensions or add new features to POS. For example, if you wanted to add new features that would retrieve product details, change a price, or add an item to a cart, you would call the POS APIs to do that work. The POS APIs simplify the extension pattern and provide continuous support to build the extensions. The extension patterns for Commerce Runtime (RT), POS, and Hardware Station now all use the request/response pattern.

This topic applies to Dynamics 365 for Finance and Operations and Dynamics 365 Retail with Platform update 8 and Retail Application update 4 hotfix.

## Scenarios

The POS APIs are categorized into three different scenarios.

- **Consume** - Consume the public APIs in your extension.
- **Extend** - Extend the public APIs to do some additional logic.
- **Create** - Create new APIs using the POS interface, which can then be used across your extensions.

## Consume POS APIs in extensions

Because the APIs are exposed using a request/response pattern, you can make an external web service call to implement your business logic. For example, if you wanted to change the price of an item, you would call **PriceOverrideOperationRequest**. The APIs are subcategorized by modules such as CRT, peripherals, and store operations.

Many new APIs have been added. You can find a list of all the APIs in the file ...**Retail** SDK\POS\Extensions\Pos.Api.d.ts.

### How to consume an API in an extension

To consume APIs in an extension, follow these steps:

1. Open Visual Studio 2015 in administrator mode.
2. Open the **ModernPOS** solution from ...\**RetailSDK\POS**.
3. Under the **POS.Extensions** project create a new folder named **POSAPIExtension**.
4. Under **POSAPIExtension**, create a new folder named **TriggersHandlers**.
5. In the **TriggersHandlers** folder, add a new Typescript file and name it **PreEndTransactionTrigger.ts**.
6. Add the following **import** statements to import the relevant entities and context.

```

import * as Triggers from "PosApi/Extend/Triggers/TransactionTriggers";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { ObjectExtensions, StringExtensions } from "PosApi/TypeExtensions";
import {
  GetCurrentCartClientRequest, GetCurrentCartClientResponse,
  SaveAttributesOnCartClientRequest, SaveAttributesOnCartClientResponse
} from "PosApi/Consume/Cart";

import {
  GetCustomerClientRequest, GetCustomerClientResponse,
} from "PosApi/Consume/Customer";

import { ShowMessageDialogClientRequest, ShowMessageDialogClientResponse } from
"PosApi/Consume/Dialogs";

```

7. Create a new class called **PreEndTransactionTrigger** and extend it from **PreEndTransactionTrigger**.

```

export default class PreEndTransactionTrigger extends Triggers.PreEndTransactionTrigger { }

```

8. Inside the class declare the following variables for the attributes names and sample values.

```

private static CART_ATTRIBUTE_NAME: string = "ATT SAMPLE";
private static CART_ATTRIBUTE_VALUE_TRUE: string = "True";
private static CART_ATTRIBUTE_VALUE_FALSE: string = "False";
private static DIALOG_RESULT_YES: string = "yes";
private static DIALOG_RESULT_NO: string = "no";
private static DIALOG_YES_BUTTON_ID: string = "CART_PreEndTransactionTrigger_MessageDialog_Yes";
private static DIALOG_NO_BUTTON_ID: string = "CART_PreEndTransactionTrigger_MessageDialog_No";

```

9. Implement the trigger **execute** method and call the existing POS APIs. The **execute** method calls APIs to get the current cart and customer, and then saves the attributes on the cart.

```

public execute(options: Triggers.IPreEndTransactionTriggerOptions):
Promise<ClientEntities.ICancelable> {
  console.log("Executing PreEndTransactionTrigger with options " + JSON.stringify(options) +
  ".");

  let currentCart: ProxyEntities.Cart;
  return this.context.runtime.executeAsync<GetCurrentCartClientResponse>(new
GetCurrentCartClientRequest())
  .then((getCurrentCartClientResponse:
ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>):
  Promise<ClientEntities.ICancelableDataResult<GetCustomerClientResponse>> => {
    currentCart = getCurrentCartClientResponse.data.result;

    // Gets the current customer.

    let result: Promise<ClientEntities.ICancelableDataResult<GetCustomerClientResponse>>;
    if (!ObjectExtensions.isNullOrUndefined(currentCart) &&
!ObjectExtensions.isNullOrUndefined(currentCart.CustomerId)) {
      let getCurrentCustomerClientRequest:
GetCustomerClientRequest<GetCustomerClientResponse> =
        new GetCustomerClientRequest(currentCart.CustomerId);
      result = this.context.runtime.executeAsync<GetCustomerClientResponse>
(getCurrentCustomerClientRequest);
    } else {
      result = Promise.resolve({ canceled: false, data: new
GetCustomerClientResponse(null) });
    }

    return result;
  })
}

```

```

        .then((getCurrentCustomerClientResponse:
ClientEntities.ICancelableDataResult<GetCustomerClientResponse>):
        Promise<ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>> => {
            let currentCustomer: ProxyEntities.Customer =
getCurrentCustomerClientResponse.data.result;
            let result:
Promise<ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>>;

            if (!ObjectExtensions.isNullOrUndefined(currentCart)

                && !ObjectExtensions.isNullOrUndefined(currentCustomer)) {

                let yesButton: ClientEntities.Dialogs.IDialogResultButton = {

                    id: PreEndTransactionTrigger.DIALOG_YES_BUTTON_ID,
                    label: "Yes", // "Yes"

                    result: PreEndTransactionTrigger.DIALOG_RESULT_YES
                };

                let noButton: ClientEntities.Dialogs.IDialogResultButton = {
                    id: PreEndTransactionTrigger.DIALOG_NO_BUTTON_ID,
                    label: "No", // "No"
                    result: PreEndTransactionTrigger.DIALOG_RESULT_NO
                };

                let showMessageDialogClientRequestOptions:
ClientEntities.Dialogs.IMessageDialogOptions = {
                    title: "Save attribute - Sample",
                    subTitle: StringExtensions.EMPTY,
                    message: "Save attribute ?",
                    button1: yesButton,
                    button2: noButton

                };

                let showMessageDialogClientRequest:
ShowMessageDialogClientRequest<ShowMessageDialogClientResponse> =
                    new ShowMessageDialogClientRequest(showMessageDialogClientRequestOptions);
                result = this.context.runtime.executeAsync<ShowMessageDialogClientResponse>
(showMessageDialogClientRequest);
            } else {
                result = Promise.resolve({ canceled: false, data: new
ShowMessageDialogClientResponse(null) });
            }
            return result;
        })

        .then((showMessageDialogClientResponse:
ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>):
        Promise<ClientEntities.ICancelableDataResult<SaveAttributesOnCartClientResponse>> =>
{

            // Save the attribute value depending on the dialog result.
            let messageDialogResult: ClientEntities.Dialogs.IMessageDialogResult =
showMessageDialogClientResponse.data.result;
            let result:
Promise<ClientEntities.ICancelableDataResult<SaveAttributesOnCartClientResponse>>;

            if (!ObjectExtensions.isNullOrUndefined(messageDialogResult)) {
                let attributeValue: ProxyEntities.AttributeTextValue = new
ProxyEntities.AttributeTextValueClass();
                attributeValue.Name = PreEndTransactionTrigger.CART_ATTRIBUTE_NAME;
                attributeValue.TextValue = messageDialogResult.dialogResult ===
PreEndTransactionTrigger.DIALOG_RESULT_YES ?
                    PreEndTransactionTrigger.CART_ATTRIBUTE_VALUE_TRUE :
                    PreEndTransactionTrigger.CART_ATTRIBUTE_VALUE_FALSE;

                let attributeValues: ProxyEntities.AttributeValueBase[] = [attributeValue];

```

```

        let saveAttributesOnCartRequest:
SaveAttributesOnCartClientRequest<SaveAttributesOnCartClientResponse> =
            new SaveAttributesOnCartClientRequest(attributeValues);
            result = this.context.runtime.executeAsync(saveAttributesOnCartRequest);
        } else {
            result = Promise.resolve({ canceled: false, data: new
SaveAttributesOnCartClientResponse(null) });
        }
        return result;
    });
}

```

The overall code should look like the following example.

```

import * as Triggers from "PosApi/Extend/Triggers/TransactionTriggers";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { ObjectExtensions, StringExtensions } from "PosApi/TypeExtensions";
import {

    GetCurrentCartClientRequest, GetCurrentCartClientResponse,
    SaveAttributesOnCartClientRequest, SaveAttributesOnCartClientResponse

} from "PosApi/Consume/Cart";

import {

    GetCustomerClientRequest, GetCustomerClientResponse,
} from "PosApi/Consume/Customer";

import { ShowMessageDialogClientRequest, ShowMessageDialogClientResponse } from
"PosApi/Consume/Dialogs";

/**
 * Example implementation of an PreEndTransactionTrigger trigger that logs to the console.
 */

export default class PreEndTransactionTrigger extends Triggers.PreEndTransactionTrigger {
    private static CART_ATTRIBUTE_NAME: string = "ATT SAMPLE";
    private static CART_ATTRIBUTE_VALUE_TRUE: string = "True";
    private static CART_ATTRIBUTE_VALUE_FALSE: string = "False";
    private static DIALOG_RESULT_YES: string = "yes";
    private static DIALOG_RESULT_NO: string = "no";
    private static DIALOG_YES_BUTTON_ID: string = "CART_PreEndTransactionTrigger_MessageDialog_Yes";
    private static DIALOG_NO_BUTTON_ID: string = " CART_PreEndTransactionTrigger_MessageDialog_No";

    /**
     * Executes the trigger functionality.
     * @param {Triggers.IPreEndTransactionTriggerOptions} options The options provided to the trigger.
     */

    public execute(options: Triggers.IPreEndTransactionTriggerOptions):
Promise<ClientEntities.ICancelable> {
        console.log("Executing PreEndTransactionTrigger with options " + JSON.stringify(options) +
".".");

        let currentCart: ProxyEntities.Cart;
        return this.context.runtime.executeAsync<GetCurrentCartClientResponse>(new
GetCurrentCartClientRequest())
            .then((getCurrentCartClientResponse:
ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>):
Promise<ClientEntities.ICancelableDataResult<GetCustomerClientResponse>> => {
                currentCart = getCurrentCartClientResponse.data.result;

                // Gets the current customer.

                let result: Promise<ClientEntities.ICancelableDataResult<GetCustomerClientResponse>>;
                if (!ObjectExtensions.IsNullOrEmpty(currentCart) &&

```

```

!ObjectExtensions.isNullOrUndefined(currentCart.CustomerId)) {
    let getCurrentCustomerClientRequest:
GetCustomerClientRequest<GetCustomerClientResponse> =
    new GetCustomerClientRequest(currentCart.CustomerId);
    result = this.context.runtime.executeAsync<GetCustomerClientResponse>
(getCurrentCustomerClientRequest);
    } else {
        result = Promise.resolve({ canceled: false, data: new
GetCustomerClientResponse(null) });
    }

    return result;
})
.then((getCurrentCustomerClientResponse:
ClientEntities.ICancelableDataResult<GetCustomerClientResponse>):
Promise<ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>> => {
    let currentCustomer: ProxyEntities.Customer =
getCurrentCustomerClientResponse.data.result;
    let result:
Promise<ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>>;

    if (!ObjectExtensions.isNullOrUndefined(currentCart)

        && !ObjectExtensions.isNullOrUndefined(currentCustomer)) {

        let yesButton: ClientEntities.Dialogs.IDialogResultButton = {

            id: PreEndTransactionTrigger.DIALOG_YES_BUTTON_ID,
            label: "Yes", // "Yes"

            result: PreEndTransactionTrigger.DIALOG_RESULT_YES
        };

        let noButton: ClientEntities.Dialogs.IDialogResultButton = {
            id: PreEndTransactionTrigger.DIALOG_NO_BUTTON_ID,
            label: "No", // "No"
            result: PreEndTransactionTrigger.DIALOG_RESULT_NO
        };

        let showMessageDialogClientRequestOptions:
ClientEntities.Dialogs.IMessageDialogOptions = {
            title: "Save attribute - Sample",
            subTitle: StringExtensions.EMPTY,
            message: "Save attribute ?",
            button1: yesButton,
            button2: noButton

        };

        let showMessageDialogClientRequest:
ShowMessageDialogClientRequest<ShowMessageDialogClientResponse> =
            new ShowMessageDialogClientRequest(showMessageDialogClientRequestOptions);
            result = this.context.runtime.executeAsync<ShowMessageDialogClientResponse>
(showMessageDialogClientRequest);
        } else {
            result = Promise.resolve({ canceled: false, data: new
ShowMessageDialogClientResponse(null) });
        }
        return result;
    })

    .then((showMessageDialogClientResponse:
ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>):
Promise<ClientEntities.ICancelableDataResult<SaveAttributesOnCartClientResponse>> =>
{

        // Save the attribute value depending on the dialog result.
        let messageDialogResult: ClientEntities.Dialogs.IMessageDialogResult =
showMessageDialogClientResponse.data.result;

```



```

        let result:
Promise<ClientEntities.ICancelableDataResult<SaveAttributesOnCartClientResponse>>;

        if (!ObjectExtensions.IsNullOrEmpty(messageDialogResult)) {
            let attributeValue: ProxyEntities.AttributeTextValue = new
ProxyEntities.AttributeTextValueClass();
            attributeValue.Name = PreEndTransactionTrigger.CART_ATTRIBUTE_NAME;
            attributeValue.TextValue = messageDialogResult.dialogResult ===
PreEndTransactionTrigger.DIALOG_RESULT_YES ?
                PreEndTransactionTrigger.CART_ATTRIBUTE_VALUE_TRUE :
PreEndTransactionTrigger.CART_ATTRIBUTE_VALUE_FALSE;

            let attributeValues: ProxyEntities.AttributeValueBase[] = [attributeValue];
            let saveAttributesOnCartRequest:
SaveAttributesOnCartClientRequest<SaveAttributesOnCartClientResponse> =
                new SaveAttributesOnCartClientRequest(attributeValues);
            result = this.context.runtime.executeAsync(saveAttributesOnCartRequest);
        } else {
            result = Promise.resolve({ canceled: false, data: new
SaveAttributesOnCartClientResponse(null) });
        }
        return result;
    });
}
}

```

10. Create a new json file under the POSAPIExtension folder and name it manifest.json.
11. In the manifest.json file, copy and paste the following code. Delete the default generated code before copying this code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_APISample",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "extend": {
      "triggers": [
        {
          "triggerType": "PreEndTransaction",
          "modulePath": "TriggersHandlers/PreEndTransactionTrigger"
        }
      ]
    }
  }
}

```

12. Open the extensions.json file under the POS.Extensions project and update it with the POSAPIExtension samples, so that POS during runtime will include this extension.

```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "POSAPIExtension"
    }
  ]
}

```

#### NOTE

The extension.json file must contain at least two extensions folder names so don't remove the **SampleExtensions** folder name.

13. Open **tsconfig.json** and comment out the extension package folders from the exclude list. POS will use this file to include or exclude the extension. By default, the list contains all the excluded extensions, if you want to include any extensions that are part of the POS, then you need add the extension folder name and comment out the extension from the extension list as shown.

#### NOTE

Comment out both **SampleExtensions2** and **POSAPIExtension**.

```
"exclude": [  
  
    "AuditEventExtensionSample",  
    "B2BSample",  
    "CustomerSearchWithAttributesSample",  
    "FiscalRegisterSample",  
    "PaymentSample",  
    "PromotionsSample",  
    "SalesTransactionSignatureSample",  
    //"SampleExtensions2",  
    "SampleExtensions",  
    "StoreHoursSample",  
    "SuspendTransactionReceiptSample",  
    "CustomControlExtensions"  
    //"POSAPIExtension"  
  
],
```

14. Select the **POS.Extensions** project and click **Show all Files** in Solution Explorer.
15. Right-click and include the **SampleExtensions2** folder in the project.
16. Compile and rebuild the project.

## Test the extension

1. Press F5 and deploy the POS to test your customization.
2. Sign in to POS and add any item to a transaction.
3. Add any customer to a transaction.
4. Click the **Pay** button and commit the transaction. A dialog box should display asking if you want to save the attribute.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# POS triggers

2/18/2021 • 18 minutes to read • [Edit Online](#)

You can use triggers to capture events that occur before or after Retail Modern POS operations. Using triggers supports several business logic scenarios that enable you to do the following:

- Insert custom logic before the operation runs or after it has completed. This includes operation-specific triggers and generic triggers called the PreOperationTrigger and PostOperationTrigger, which run at the beginning and end of all POS operations.
- Continue or cancel an operation. For example, if your validation fails or returns an error, then you can cancel the operation in pre-trigger. Post-triggers are not cancelable.
- Use the post-trigger for scenarios where you want to show custom messages or insert custom fields after the standard logic is performed.

The following table lists the available triggers and denotes whether they can be canceled.

## Application triggers

TRIGGER	TYPE	DESCRIPTION
ApplicationStartTrigger	Non-cancelable	Executed after the POS application is started. You can revert or cancel whatever executed previously.
ApplicationSuspendTrigger	Non-cancelable	Executed after the POS application is suspended.
PreLogOnTriggerTrigger	Cancelable	Executed before the POS log on.
PostLogOnTriggerTrigger	Non-cancelable	Executed after the POS log on.
PostLogOffTrigger	Non-cancelable	Executed after the POS log off.
PreLockTerminalTrigger	Cancelable	Executed before the POS register lock.
PostLockTerminalTrigger	Non-Cancelable	Executed after the POS register lock.
PreUnlockTerminalTrigger	Cancelable	Executed before the POS register is unlocked.
PostDeviceActivationTrigger	Non-Cancelable	Executed after the POS activation.
PreElevateUserTrigger	Cancelable	Executed before the manager override.
PreRegisterAuditEventTrigger	Cancelable	Executed before the audit event.
PostRegisterAuditEventTrigger	Non-Cancelable	Executed after the audit event.
PreOpenUrlTrigger	Cancelable	Executed before the open URL operation.

## Cash management triggers

TRIGGER	TYPE	DESCRIPTION
PreTenderDeclarationTrigger	Cancelable	Executed before the POS tender declaration.
PostTenderDeclarationTrigger	Non-cancelable	Executed after the POS tender declaration.
PreFloatEntryTrigger	Cancelable	Executed before the POS float entry.
PostFloatEntryTrigger	Non-cancelable	Executed after the POS float entry.

## Customer triggers

TRIGGER	TYPE	DESCRIPTION
PreCustomerAddTrigger	Cancelable	Executed before adding a customer to the transaction.
PostCustomerAddTrigger	Non-cancelable	Executed after adding a customer to the transaction.
PreCustomerClearTrigger	Cancelable	Executed before the customer cleared from the cart.
PostCustomerClearTrigger	Non-cancelable	Executed after the customer cleared from the cart.
PreCustomerSetTrigger	Cancelable	Executed before the customer is added to the cart.
PreCustomerSearchTrigger	Cancelable	Executed before customer search is performed.
PostCustomerSearchTrigger	Non-cancelable	Executed after customer search is performed.
PostIssueLoyaltyCardTrigger	Non-cancelable	Executed after the loyalty card is issued.
PreCustomerSaveTrigger	Cancelable	Executed before the customer is created.
PostCustomerSaveTrigger	Non-cancelable	Executed after the customer is created.
PreSaveCustomerAddressTrigger	Cancelable	Executed before the customer address is saved.
PreGetLoyaltyCardBalanceTrigger	Cancelable	Executed before getting the loyalty card balance.

TRIGGER	TYPE	DESCRIPTION
PostGetLoyaltyCardBalanceTrigger	Non-cancelable	Executed after getting the loyalty card balance.
PreDisplayLoyaltyCardBalanceTrigger	Cancelable	Executed before displaying the loyalty card balance.

## Discount triggers

TRIGGER	TYPE	DESCRIPTION
PreLineDiscountAmountTrigger	Cancelable	Executed before line discount amount is added to the cart line.
PostLineDiscountAmountTrigger	Non-cancelable	Executed after line discount amount added to the cart line.
PreLineDiscountPercentTrigger	Cancelable	Executed before line discount percent added to the cart line.
PostLineDiscountPercentTrigger	Non-cancelable	Executed after line discount percent added to the cart line.
PreTotalDiscountAmountTrigger	Cancelable	Executed before total discount percent added to the cart.
PostTotalDiscountAmountTrigger	Non-cancelable	Executed after total amount percent added to the cart.
PreTotalDiscountPercentTrigger	Cancelable	Executed before total discount percent added to the cart.
PostTotalDiscountPercentTrigger	Non-cancelable	Executed after total discount percent added to the cart.
PreAddCouponTrigger	Cancelable	Executed before adding discount coupon to the cart.
PostAddCouponTrigger	Non-cancelable	Executed after adding discount coupon to the cart.

## Operation triggers

TRIGGER	TYPE	DESCRIPTION
PreOperationTrigger	Cancelable	Generic trigger executed before all POS operations. You can use this trigger if there is no specific trigger available for a POS operation.
PreOperationValidationTrigger	Cancelable	Generic trigger executed before the operation validation begins.

TRIGGER	TYPE	DESCRIPTION
OperationFailureTrigger	Non-cancelable	Generic trigger executed after any POS operation failed.
PostOperationTrigger	Non-cancelable	Generic trigger executed after all POS operations. You can use this trigger if there is no specific trigger available for a POS operation.

## Payment triggers

TRIGGER	TYPE	DESCRIPTION
PreAddTenderLineTrigger	Cancelable	Executed before the payment line is added to the cart.
PrePaymentTrigger	Cancelable	Executed after you click the pay button in POS.
PostPaymentTrigger	Non-cancelable	Executed after all the payment processing is done.
PreVoidPaymentTrigger	Cancelable	Executed before the payment line is voided in POS.
PostVoidPaymentTrigger	Non-cancelable	Executed after the payment line is voided in POS.

## Printing Triggers

TRIGGER	TYPE	DESCRIPTION
PrePrintReceiptCopyTrigger	Cancelable	Executed before the receipt copy is printed from the show journal screen or receipt view screen.
PostReceiptPromptTrigger	Non-cancelable	Executed after the receipt prompt - Do you want to print or not print receipt.

## Product triggers

TRIGGER	TYPE	DESCRIPTION
PostGetSerialNumberTrigger	Non-cancelable	Executed after the serial number is added to the cart line.
PreProductSaleTrigger	Cancelable	Executed before the product is added to the cart.
PostProductSaleTrigger	Non-cancelable	Executed after the product is added to the cart.

TRIGGER	TYPE	DESCRIPTION
PreReturnProductTrigger	Cancelable	Executed before the return product is added to the cart.
PostReturnProductTrigger	Non-cancelable	Executed after the return product is added to the cart.
PreSetQuantityTrigger	Cancelable	Executed before the quantity information is updated in the cart line.
PostSetQuantityTrigger	Non-cancelable	Executed after the quantity information is updated in the cart line.
PrePriceOverrideTrigger	Cancelable	Executed before the price is overridden for a cart line.
PostPriceOverrideTrigger	Non-cancelable	Executed after the price is overridden for a cart line.
PreClearQuantityTrigger	Cancelable	Executed before the quantity information is cleared from the cart line.
PostClearQuantityTrigger	Non-cancelable	Executed after the quantity information is cleared from the cart line.
PreVoidProductsTrigger	Cancelable	Executed before the product is voided from the cart.
PostVoidProductsTrigger	Non-cancelable	Executed after the product is voided from the cart.
PostPriceCheckTrigger	Non-cancelable	Executed after the price check for the product is executed.

## Sales order triggers

TRIGGER	TYPE	DESCRIPTION
PreRecallCustomerOrderTrigger	Cancelable	Executed before the customer order is recalled.
PostRecallCustomerOrderTrigger	Non-cancelable	Executed after the customer order is recalled.
PrePickUpCustomerOrderLinesTrigger	Cancelable	Executed before the customer order lines are picked.
PreChangeShippingOriginTrigger	Cancelable	Executed before the shipping origin is changed during a customer order.

TRIGGER	TYPE	DESCRIPTION
PreGetFulfillmentLinesTrigger	Cancelable	Executed before the Order fulfillment lines are loaded in the Order fulfillment view.
PreShipFulfillmentLinesTrigger	Cancelable	Executed before the shipping is done from the Order fulfillment view by selecting the <b>Ship</b> button.
PostShipFulfillmentLinesTrigger	Non-Cancelable	Executed after the shipping is done from the Order fulfillment view by selecting the <b>Ship</b> button.
PreMarkFulfillmentLinesAsPackedTrigger	Cancelable	Executed before the mark as packed option is triggered from the order fulfillment view by selecting the <b>Pack</b> button.
PostMarkFulfillmentLinesAsPackedTrigger	Non-Cancelable	Executed after the mark as packed option is triggered from the order fulfillment view by selecting the <b>Pack</b> button.
PreCreatePackingSlipTrigger	Cancelable	Executed before the create packing slip option is triggered from the order fulfillment view by selecting the <b>Pack</b> button.
PostCreatePackingSlipTrigger	Non-Cancelable	Executed after the create packing slip option is triggered from the order fulfillment view by selecting the <b>Pack</b> button.
PostReturnInvoicedSalesLinesTrigger	Non-Cancelable	Executed after one or more invoices selected for return.
PreResendEmailReceiptTrigger (10.0.13)	Cancelable	Executed before sending the email from the Show journal view.
PreRecallCustomerQuoteTrigger (10.0.18)	Cancelable	Executed before the customer quote is recalled from the recall order view.
PostRecallCustomerQuoteTrigger (10.0.18)	Non-Cancelable	Executed after the customer quote is recalled from the recall order view.

## Shift triggers

TRIGGER	TYPE	DESCRIPTION	RELEASE
PostOpenShiftTrigger	Non-cancelable	This trigger is executed after the new shift is opened.	
PreCloseShiftTrigger	Cancelable	This trigger is executed before the shift is closed.	



TRIGGER	TYPE	DESCRIPTION	RELEASE
PreResumeShiftTrigger	Cancelable	This trigger is executed before the shift is resumed.	10.0.16
PostResumeShiftTrigger	Non-cancelable	This trigger is executed after the shift is resumed.	10.0.16

## Tax override triggers

TRIGGER	TYPE	DESCRIPTION
PreOverrideLineProductTaxTrigger	Cancelable	Executed before the tax amount or code is overridden for a cart line.
PostOverrideLineProductTaxTrigger	Non-cancelable	Executed after the tax amount or code is overridden for a cart line.
PreOverrideTransactionTaxTrigger	Cancelable	Executed before the tax amount or code is overridden for a cart or transaction.
PostOverrideTransactionTaxTrigger	Non-cancelable	Executed before the tax amount or code is overridden for a cart or transaction.

## Transaction triggers

TRIGGER	TYPE	DESCRIPTION
BeginTransactionTrigger	Non-cancelable	Executed before the transaction is initialized.
PreConfirmReturnTransactionTrigger	Cancelable	Executed before the return transaction is confirmed.
PreReturnTransactionTrigger	Cancelable	Executed before the return transaction is processed.
PostReturnTransactionTrigger	Non-cancelable	Executed after the return transaction is processed.
PreEndTransactionTrigger	Cancelable	Executed before the end transaction request is called to commit the changes to DB and close the transaction.
PostEndTransactionTrigger	Non-cancelable	Executed after the end transaction request is called to commit the changes to DB and close the transaction.
PreVoidTransactionTrigger	Cancelable	Executed before the transaction is voided.

TRIGGER	TYPE	DESCRIPTION
PostVoidTransactionTrigger	Non-cancelable	Executed after the transaction is voided.
PreSuspendTransactionTrigger	Cancelable	Executed before the transaction is suspended.
PostSuspendTransactionTrigger	Non-cancelable	Executed after the transaction is suspended.
PreRecallTransactionTrigger	Cancelable	Executed before the transaction or order is recalled.
PostRecallTransactionTrigger	Non-cancelable	Executed after the transaction or order is recalled.
PostCartCheckoutTrigger	Non-cancelable	Executed after the checkout process is completed.
PreRecallTransactionTrigger	Cancelable	Executed before the customer order is recalled.
PostRecallTransactionTrigger	Non-Cancelable	Executed after the customer order is recalled.
PreSelectTransactionPaymentMethodTrigger	Cancelable	When the user selects the <b>Totals</b> button in the <b>Cart view - totals</b> panel, the available payment methods are shown and this trigger will get executed before this dialog is shown. You can use extension code to modify the available payment methods from this trigger.
PreShipSelectedCartLinesTrigger	Cancelable	Executed when the product is selected for shipping.

## Reason code triggers

TRIGGER	TYPE	DESCRIPTION
PostGetReasonCodeLine	Cancelable	This trigger is executed after the reason code line value is entered (before the reason code is added to the cart).

## Transfer Order triggers

TRIGGER	TYPE	DESCRIPTION
PreCreateTransferOrderTrigger	Cancelable	This trigger is executed before the transfer order is created (executed after the order input).

TRIGGER	TYPE	DESCRIPTION
PreUpdateTransferOrderTrigger	Cancelable	This trigger is executed before the transfer order is updated.

## Inventory triggers

TRIGGER	TYPE	DESCRIPTION	RELEASE
PreCreateInventoryDocumentTrigger	Cancelable	This trigger is executed before the inbound/outbound document is created (executed after the order input).	10.0.15
PreUpdateInventoryDocumentTrigger	Cancelable	This trigger is executed before the inbound/outbound document is updated.	10.0.15

## Stock count triggers

TRIGGER	TYPE	DESCRIPTION	RELEASE
PreAdjustStockCountLineQuantityTrigger	Cancelable	This trigger is executed before the stock count for a line is adjusted.	10.0.16
PreSaveStockCountJournalTrigger	Cancelable	This trigger is executed before the stock count journal is saved.	10.0.16

## Business scenario

In this example, a custom receipt is printed when the user suspends a transaction. This example implements the **PostSuspendTransactionTrigger** trigger and prints the custom receipt using the existing print peripheral API.

To implement this scenario, you must complete these steps.

1. **POS extension:** Implement the **PostSuspendTransactionTrigger** trigger to get the receipt data from CRT and send it to the printer for printing.
2. **CRT extension:** Generate the receipt data for printing.

## Implement a trigger

1. Open Visual Studio 2015 in administrator mode.
2. Open the **ModernPOS** solution from `...\RetailSDK\POS`.
3. Under the **POS.Extensions** project, create a new folder named **SuspendReceiptSample**.
4. Under **SuspendReceiptSample**, create new folder named **TriggersHandlers**.
5. In the **TriggersHandlers** folder, add a new Typescript file named **PostSuspendTransactionTrigger.ts**.

6. Add the following **import** statements to import the relevant entities and context.

```
import * as Triggers from "PosApi/Extend/Triggers/TransactionTriggers";
import { ObjectExtensions } from "PosApi/TypeExtensions";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { PrinterPrintRequest, PrinterPrintResponse } from "PosApi/Consume/Peripherals";
import { GetHardwareProfileClientRequest, GetHardwareProfileClientResponse } from
"PosApi/Consume/Device";
import { GetReceiptsClientRequest, GetReceiptsClientResponse } from "PosApi/Consume/SalesOrders";
```

7. Create a new class named **PostSuspendTransactionTrigger** and extend it from **PostSuspendTransactionTrigger**.

```
export default class PostSuspendTransactionTrigger extends Triggers.PostSuspendTransactionTrigger { }
```

8. Implement the trigger's **execute** method to get the receipt profile and print the custom receipt:

```

public execute(options: Triggers.IPostSuspendTransactionTriggerOptions): Promise < void> {
    this.context.logger.logVerbose("Executing PostSuspendTransactionTrigger with options " +
JSON.stringify(options) + ".");
    if(ObjectExtensions.isNullOrUndefined(options) ||
ObjectExtensions.isNullOrUndefined(options.cart)) {

        // This will never happen, but is included to demonstrate how to return a rejected promise
when validation fails.

        let error: ClientEntities.ExtensionError
            = new ClientEntities.ExtensionError("The options provided to the
PostSuspendTransactionTrigger were invalid.");
        return Promise.reject(error);
    } else {
        return this.context.runtime.executeAsync(new GetHardwareProfileClientRequest())
            .then((response: ClientEntities.ICancelableDataResult<GetHardwareProfileClientResponse>
                : Promise<ClientEntities.ICancelableDataResult<GetReceiptsClientResponse>> => {
                    let hardwareProfile: ProxyEntities.HardwareProfile = response.data.result;
                    // Gets the receipts.
                    let salesOrderId: string = options.cart.Id;
                    let receiptRetrievalCriteria: ProxyEntities.ReceiptRetrievalCriteria = {
                        IsCopy: false,
                        IsRemoteTransaction: false,
                        IsPreview: false,
                        QueryBySalesId: true,
                        ReceiptTypeValue: ProxyEntities.ReceiptType.CustomReceipt7,
                        HardwareProfileId: hardwareProfile.ProfileId
                    };
                    let getReceiptsClientRequest: GetReceiptsClientRequest<GetReceiptsClientResponse> =
                        new GetReceiptsClientRequest(salesOrderId, receiptRetrievalCriteria);
                    return this.context.runtime.executeAsync(getReceiptsClientRequest);
                })
            .then((response: ClientEntities.ICancelableDataResult<GetReceiptsClientResponse>
                : Promise<ClientEntities.ICancelableDataResult<PrinterPrintResponse>> => {
                    let receipts: ProxyEntities.Receipt[] = response.data.result;
                    // Prints the receipts.
                    let printerPrintRequest: PrinterPrintRequest<PrinterPrintResponse> = new
PrinterPrintRequest(receipts);
                    return this.context.runtime.executeAsync(printerPrintRequest);
                }).then(): Promise<void> => {
                    // Resolves to a void result when fulfilled.
                    return Promise.resolve();
                }).catch((reason: any): Promise<void> => {
                    // Resolves to a void result when rejected. This matches existing POS printing
behavior.
                    this.context.logger.logError("PostSuspendTransactionTrigger execute error: " +
JSON.stringify(reason));
                    return Promise.resolve();
                });
    }
}

```

The entire example should look like the following.

```

import * as Triggers from "PosApi/Extend/Triggers/TransactionTriggers";
import { ObjectExtensions } from "PosApi/TypeExtensions";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { PrinterPrintRequest, PrinterPrintResponse } from "PosApi/Consume/Peripherals";
import { GetHardwareProfileClientRequest, GetHardwareProfileClientResponse } from
"PosApi/Consume/Device";
import { GetReceiptsClientRequest, GetReceiptsClientResponse } from "PosApi/Consume/SalesOrders";

export default class PostSuspendTransactionTrigger extends Triggers.PostSuspendTransactionTrigger {

/**
 * Executes the trigger functionality.

```

```

    * @param {Triggers.IPostSuspendTransactionTriggerOptions} options The options provided to the
    trigger.
    */
    public execute(options: Triggers.IPostSuspendTransactionTriggerOptions): Promise<void> {
        this.context.logger.logVerbose("Executing PostSuspendTransactionTrigger with options " +
JSON.stringify(options) + ".");
        if (ObjectExtensions.IsNullOrEmpty(options) ||
ObjectExtensions.IsNullOrEmpty(options.cart)) {
            // This will never happen, but is included to demonstrate how to return a rejected
            promise when validation fails.
            let error: ClientEntities.ExtensionError
                = new ClientEntities.ExtensionError("The options provided to the
PostSuspendTransactionTrigger were invalid.");
            return Promise.reject(error);
        } else {
            return this.context.runtime.executeAsync(new GetHardwareProfileClientRequest())
                .then((response:
ClientEntities.ICancelableDataResult<GetHardwareProfileClientResponse>
                : Promise<ClientEntities.ICancelableDataResult<GetReceiptsClientResponse>> => {
                    let hardwareProfile: ProxyEntities.HardwareProfile = response.data.result;
                    // Gets the receipts.
                    let salesOrderId: string = options.cart.Id;
                    let receiptRetrievalCriteria: ProxyEntities.ReceiptRetrievalCriteria = {
                        IsCopy: false,
                        IsRemoteTransaction: false,
                        IsPreview: false,
                        QueryBySalesId: true,
                        ReceiptTypeValue: ProxyEntities.ReceiptType.CustomReceipt7,
                        HardwareProfileId: hardwareProfile.ProfileId
                    };
                    let getReceiptsClientRequest: GetReceiptsClientRequest<GetReceiptsClientResponse>
                    =
                        new GetReceiptsClientRequest(salesOrderId, receiptRetrievalCriteria);
                    return this.context.runtime.executeAsync(getReceiptsClientRequest);
                })
                .then((response: ClientEntities.ICancelableDataResult<GetReceiptsClientResponse>
                : Promise<ClientEntities.ICancelableDataResult<PrinterPrintResponse>> => {
                    let receipts: ProxyEntities.Receipt[] = response.data.result;
                    // Prints the receipts.
                    let printerPrintRequest: PrinterPrintRequest<PrinterPrintResponse> = new
PrinterPrintRequest(receipts);
                    return this.context.runtime.executeAsync(printerPrintRequest);
                }).then(): Promise<void> => {
                    // Resolves to a void result when fulfilled.
                    return Promise.resolve();
                }).catch((reason: any): Promise<void> => {
                    // Resolves to a void result when rejected. This matches existing POS printing
                    behavior.
                    this.context.logger.logError("PostSuspendTransactionTrigger execute error: " +
JSON.stringify(reason));
                    return Promise.resolve();
                });
        }
    }
}
}
}

```

9. Create a new json file under the **SuspendReceiptSample** folder and name it **manifest.json**.
10. Replace the autogenerated code in **manifest.json** with the following code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_SuspendTransactionReceiptSample",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "extend": {
      "triggers": [
        {
          "triggerType": "PostSuspendTransaction",
          "modulePath": "TriggersHandlers/PostSuspendTransactionTrigger"
        }
      ]
    }
  }
}

```

11. Open the **extensions.json** file in the **POS.Extensions** project and update it with the **SuspendReceiptSample** samples, so that during runtime POS will include this extension.

```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "SuspendReceiptSample"
    }
  ]
}

```

12. Open the **tsconfig.json** file and comment out the extension package folders from the exclude list. POS will use this file to include or exclude the extension. By default, the list contains all the excluded extensions list. If you want to include an extension that is part of the POS, then add the extension folder name and comment out the extension from the extension, as shown.

```

"exclude": [
  "AuditEventExtensionSample",
  "B2BSample",
  "CustomerSearchWithAttributesSample",
  "FiscalRegisterSample",
  "PaymentSample",
  "PromotionsSample",
  "SalesTransactionSignatureSample",
  "SampleExtensions",
  //"SampleExtensions2",
  "StoreHoursSample",
  "SuspendTransactionReceiptSample"
  //"POSAPIExtension",
  //"CustomColumnExtensions",
  //"EODSample",
  //"ProdDetailsCustomColumnExtensions",
  //"SerachExtension",
  //"SuspendReceiptSample"
],

```

13. Compile and rebuild the project.

## Override the CRT receipt request to generate the receipt data

This section explains how to override the existing CRT request to print a receipt for suspended transactions.

1. Start Visual Studio 2015.
2. On the **File** menu, select **Open > Project/Solution**. Find the template project (**SampleCRTExtension.csproj**).
3. Rename the template project **Runtime.Extensions.SuspendReceiptSample**.
4. Optional: Change the default namespace.
5. Rename the output assembly **Contoso.Commerce.Runtime.SuspendReceiptSample**.
6. Inside the project, add a new request class file, and name it **GetCustomReceiptsRequestHandler.cs**.
7. Copy the following code, and paste it inside the class. Before you copy it, delete the automatically generated code.

```
namespace Contoso
{
    namespace Commerce.Runtime.ReceiptsSample
    {
        using System.Collections.Generic;
        using System.Collections.ObjectModel;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Workflow;

        /// <summary>
        /// The request handler for GetCustomReceiptsRequestHandler class.
        /// </summary>

        /// <remarks>
        /// This is an example of how to print custom types of receipts. In this example the receipt
        is for a transaction as opposed to
        /// a sales order. The implementation converts the transaction to a sales order so that
        existing receipt fields can be used.
        /// </remarks>

        public class GetCustomReceiptsRequestHandler : SingleRequestHandler<GetCustomReceiptsRequest,
        GetReceiptResponse>
        {
            {
            }
        }
    }
}
```

8. Copy the following code, and paste it inside the class to add a new method to read the transaction from the cart table, because suspended transactions aren't yet created in the commerce transaction table. You must then convert the cart transaction to sales transaction. This conversion is required because the receipt object can understand only sales transactions.

#### **NOTE**

If you're printing a custom receipt for a completed transaction, you don't have to get the cart transaction and convert it to a sales transaction. It has already been converted to a sales transaction, because the transaction is completed.



```

private SalesOrder GetSalesOrderForTransactionWithId(RequestContext requestContext, string
transactionId)
{
    SalesOrder salesOrder = new SalesOrder();
    var getCartRequest = new GetCartRequest(new CartSearchCriteria(transactionId),
QueryResultSettings.SingleRecord);
    var getCartResponse = requestContext.Execute<GetCartResponse>(getCartRequest);
    SalesTransaction salesTransaction = getCartResponse.Transactions.SingleOrDefault(); ;
    if (salesTransaction != null)
    {
        // The sales transaction is converted into a sales order so that existing receipt fields can
be used.
        salesOrder.CopyFrom<SalesTransaction>(salesTransaction);
    }
    else
    {
        throw new DataValidationException(
            DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_ObjectNotFound,
            string.Format("Unable to get the sales transaction. ID: {0}", transactionId));
    }
    return salesOrder;
}

```

9. Copy the following code, and paste it into the class to add a new method to construct the receipt format by using the sales transaction information, based on the custom receipt format that is defined in HQ.

```

private Collection<Receipt> GetCustomReceipts(SalesOrder salesOrder, ReceiptRetrievalCriteria
criteria)
{
    Collection<Receipt> result = new Collection<Receipt>();
    var getReceiptServiceRequest = new GetReceiptServiceRequest(
        salesOrder,
        new Collection<ReceiptType> { criteria.ReceiptType },
        salesOrder.TenderLines,
        criteria.IsCopy,
        criteria.IsPreview,
        criteria.HardwareProfileId);
    ReadOnlyCollection<Receipt> customReceipts = this.Context.Execute<GetReceiptServiceResponse>
(getReceiptServiceRequest).Receipts;
    result.AddRange(customReceipts);
    return result;
}

```

10. Copy the following code, and paste it into the class to add a new method to override the get receipt response. This method processes the request and returns the response.

```

protected override GetReceiptResponse Process(GetCustomReceiptsRequest request)
{
    ThrowIf.Null(request, "request");
    ThrowIf.Null(request.ReceiptRetrievalCriteria, "request.ReceiptRetrievalCriteria");

    // The sales order that we are printing receipts for is retrieved.
    SalesOrder salesOrder = this.GetSalesOrderForTransactionWithId(request.RequestContext,
request.TransactionId);

    // Custom receipts are printed.
    Collection<Receipt> result = new Collection<Receipt>();
    switch (request.ReceiptRetrievalCriteria.ReceiptType)
    {
        // An example of getting custom receipts.
        case ReceiptType.CustomReceipt7:
        {
            IEnumerable<Receipt> customReceipts = this.GetCustomReceipts(salesOrder,
request.ReceiptRetrievalCriteria);
            result.AddRange(customReceipts);
        }
        break;
        default:

            // Add more logic to handle more types of custom receipt types.

            break;
    }
    return new GetReceiptResponse(new ReadOnlyCollection<Receipt>(result));
}

```

The overall code should look like this.

```

namespace Contoso
{
    namespace Commerce.Runtime.ReceiptsSample
    {
        using System.Collections.Generic;
        using System.Collections.ObjectModel;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Workflow;

        /// <summary>
        /// The request handler for GetCustomReceiptsRequestHandler class.
        /// </summary>

        /// <remarks>
        /// This is an example of how to print custom types of receipts. In this example the receipt
is for a transaction as opposed to
        /// a sales order. The implementation converts the transaction to a sales order so that
existing receipt fields can be used.
        /// </remarks>

        public class GetCustomReceiptsRequestHandler : SingleRequestHandler<GetCustomReceiptsRequest,
GetReceiptResponse>
        {
            /// <summary>
            /// Processes the GetCustomReceiptsRequest to return the set of receipts. The request
should not be null.
            /// </summary>

            /// <param name="request">The request parameter.</param>

```

```

/// <returns>The GetReceiptResponse.</returns>

protected override GetReceiptResponse Process(GetCustomReceiptsRequest request)
{
    ThrowIf.Null(request, "request");
    ThrowIf.Null(request.ReceiptRetrievalCriteria, "request.ReceiptRetrievalCriteria");

    // The sales order that we are printing receipts for is retrieved.
    SalesOrder salesOrder =
this.GetSalesOrderForTransactionWithId(request.RequestContext, request.TransactionId);

    // Custom receipts are printed.
    Collection<Receipt> result = new Collection<Receipt>();
    switch (request.ReceiptRetrievalCriteria.ReceiptType)
    {
        // An example of getting custom receipts.
        case ReceiptType.CustomReceipt7:
        {
            IEnumerable<Receipt> customReceipts = this.GetCustomReceipts(salesOrder,
request.ReceiptRetrievalCriteria);
            result.AddRange(customReceipts);
        }
        break;
        default:

            // Add more logic to handle more types of custom receipt types.

            break;
    }
    return new GetReceiptResponse(new ReadOnlyCollection<Receipt>(result));
}

/// <summary>
/// Gets a sales order for the transaction with the given identifier.
/// </summary>

/// <param name="requestContext">The request context.</param>
/// <param name="transactionId">The transaction identifier.</param>

/// <returns>The sales order.</returns>

private SalesOrder GetSalesOrderForTransactionWithId(RequestContext requestContext,
string transactionId)
{
    SalesOrder salesOrder = new SalesOrder();
    var getCartRequest = new GetCartRequest(new CartSearchCriteria(transactionId),
QueryResultSettings.SingleRecord);
    var getCartResponse = requestContext.Execute<GetCartResponse>(getCartRequest);
    SalesTransaction salesTransaction = getCartResponse.Transactions.SingleOrDefault(); ;
    if (salesTransaction != null)
    {
        // The sales transaction is converted into a sales order so that existing receipt
fields can be used.
        salesOrder.CopyFrom<SalesTransaction>(salesTransaction);
    }
    else
    {
        throw new DataValidationException(
DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_ObjectNotFound,
string.Format("Unable to get the sales transaction. ID: {0}", transactionId));
    }
    return salesOrder;
}

/// <summary>
/// An example to get a custom receipt.
/// </summary>

/// <param name="salesOrder">The sales order that we are printing receipts for.</param>

```

```

    /// <param name="criteria">The receipt retrieval criteria.</param>

    /// <returns>A collection of receipts.</returns>

    private Collection<Receipt> GetCustomReceipts(SalesOrder salesOrder,
ReceiptRetrievalCriteria criteria)
    {
        Collection<Receipt> result = new Collection<Receipt>();
        var getReceiptServiceRequest = new GetReceiptServiceRequest(
            salesOrder,
            new Collection<ReceiptType> { criteria.ReceiptType },
            salesOrder.TenderLines,
            criteria.IsCopy,
            criteria.IsPreview,
            criteria.HardwareProfileId);
        ReadOnlyCollection<Receipt> customReceipts =
this.Context.Execute<GetReceiptServiceResponse>(getReceiptServiceRequest).Receipts;
        result.AddRange(customReceipts);
        return result;
    }
}
}
}
}

```

11. Compile and build the project.
12. Go to the output directory, and copy the output assembly.
13. Navigate to the ...\**RetailServer\webroot\bin\ext** folder, and paste the assembly.
14. Also paste the assembly in the ...\**RetailSDK\References** folder.
15. Open the **commercerruntime.ext.config** file, and add the custom assembly information under the <composition> section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.SuspendReceiptSample" />
```

16. Save and close the file.
17. Restart the Commerce Scale Unit to load the new assembly.

## Add the custom receipt layout

1. Open Dynamics 365 Commerce.
2. Go to **Retail and Commerce > Channel setup > POS setup > POS > Receipts formats**.
3. Click **New** in **Receipts formats**.
4. In the **Receipt format filed** field, enter the format name **Suspend**. In the **Receipt type** field, select **CustomReceiptType7**.
5. Click **Designer** to open the receipt designer.
6. Follow the instructions if prompted to install. Enter the Azure Active Directory (AAD) credentials to launch the designer.
7. Drag and drop the required field into the header, lines, or footer. Or, copy the from existing receipt format by using the copy feature and then edit it.
8. Click **Save**.
9. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Receipt profiles**.
10. Select the **Email receipt** profile or the that profile you store. Click the **Add** button in the **General** tab.
11. In the **Receipt type**, select **CustomReceiptType7** and in the **Receipt format** select **Suspend**.
12. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.

13. Select **Registers (1090)** and click **Run now** in the action bar. When prompted, click **Yes** to run the job.

## Configure the XPS printer for quick testing

1. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Hardware profiles**.
2. Select the hardware profile that your device is using. For example, in the demo data all the Houston devices uses **HW002**.
3. Click **Edit** in the action bar.
4. Expand the **Printer** FastTab. In the **Printer** drop-down list, select **Windows driver** and in the device name field enter **Microsoft XPS Document Writer**.
5. Click **Save**.
6. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
7. Select **Registers (1090)** and click **Run now** in the action bar. When prompted, click **Yes** to run the job.

## Validate the extension

1. Press **F5** and deploy the POS to test your customization.
2. After the POS starts, sign in to POS and add an item to a transaction.
3. Suspend the transaction.
4. The custom receipt should print.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add custom controls to non-screen designer-based POS views

2/18/2021 • 6 minutes to read • [Edit Online](#)

You can enhance the information displayed on a Dynamics 365 Commerce POS view by adding custom controls. A custom control allows you to add your own custom information to the existing POS views. Custom controls can be implemented by using the POS extension framework. Currently, you cannot place the custom control in the desired location, at runtime, POS will load it in a fixed position.

This topic applies to Dynamics 365 for Finance and Operations, and Dynamics 365 Retail with Platform update 8, and Retail Application update 4 hotfix.

The following table lists the non-screen layout designer-based views that support custom controls.

POS VIEWS	SUPPORT CUSTOM CONTROL	NUMBER OF CUSTOM CONTROLS
Customer Add/Edit view	Yes	Multiple
Address Add/Edit view	Yes	Multiple
Customer details view	Yes	Multiple
Product details view	Yes	Multiple
Price check view	Yes	Multiple

The following table lists the screen layout designer based-views that support custom controls.

POS VIEWS	SUPPORT CUSTOM CONTROL	NUMBER OF CUSTOM CONTROLS
Cart view	Yes	10

## Create the custom control

The following example demonstrates how to add a custom control to one of the existing POS views using extensions. For example, suppose you want to show the product availability information in the product details view by adding custom data list that has four columns - Location, Inventory, Reserved, and Ordered.

A custom control is an HTML page with the custom information to be displayed. A corresponding Typescript file contains the logic for the control.

1. Open Visual Studio 2015 in administrator mode.
2. Open Modern POS from `\RetailSDK\POS`.
3. Under the `POS.Extensions` project, create a new folder named `ProdDetailsCustomColumnExtensions`.
4. Under `ProdDetailsCustomColumnExtensions`, create a new folder named `ViewExtensions`.
5. Under `ViewExtensions`, create new folder named `SimpleProductDetails`.

6. Add a new HTML file inside the **SimpleProductDetails** folder and name it **ProductAvailabilityPanel.html**.
7. Open **ProductAvailabilityPanel.html** and add the following code. The code adds a POS data list control to show the product availability information and the width of the control.

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>
  <!-- Note: The element ID is different than the ID generated by the POS extensibility framework.
  This 'template' ID is not used by the POS extensibility framework. -->
  <script id="Microsoft_Pos_Extensibility_Samples_ProductAvailabilityPanel" type="text/html">
    <h2 class="marginTop8 marginBottom8" data-bind="text: title"></h2>
    <div class="width400 grow col">
      <div id="Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel_DataList" data-
bind="msPosDataList: dataList"></div>
    </div>
  </script>
</body>
</html>
```

8. In the **SimpleProductDetails** folder, add a new typescript file and name it **ProductAvailabilityPanel.ts**.
9. Add the following **import** statements to import the relevant entities and context.

```
import {

  SimpleProductDetailsCustomControlBase,
  ISimpleProductDetailsCustomControlState,
  ISimpleProductDetailsCustomControlContext

} from "PosApi/Extend/Views/SimpleProductDetailsView";

import { InventoryLookupOperationRequest, InventoryLookupOperationResponse } from
"PosApi/Consume/OrgUnits";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { ArrayExtensions } from "PosApi/TypeExtensions";
import { DataList, SelectionMode } from "PosUISdk/Controls/DataList";
```

10. Create a new class named **ProductAvailabilityPanel** and extend it from **SimpleProductDetailsCustomControlBase**.

```
export default class ProductAvailabilityPanel extends SimpleProductDetailsCustomControlBase { }
```

11. Inside the class, declare the following variables for state and data list information.

```
private static readonly TEMPLATE_ID: string =
"Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel";
public readonly orgUnitAvailabilities: ObservableArray<ProxyEntities.OrgUnitAvailability>;
public readonly dataList: DataList<ProxyEntities.OrgUnitAvailability>;
public readonly title: Observable<string>;
private _state: ISimpleProductDetailsCustomControlState;
```

12. Add a class constructor method to initialize the data list columns.

```

constructor(id: string, context: ISimpleProductDetailsCustomControlContext) {

    super(id, context);
    this.orgUnitAvailabilities = ko.observableArray([]);
    this.title = ko.observable("Product Availability");
    this.dataList = new DataList<ProxyEntities.OrgUnitAvailability>({
        columns: [

            {
                title: "Location",
                ratio: 31,
                collapseOrder: 4,
                minWidth: 100,
                computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                    return value.OrgUnitLocation.OrgUnitName;
                }
            },

            {
                title: "Inventory",
                ratio: 23,
                collapseOrder: 3,
                minWidth: 60,
                computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                    return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
value.ItemAvailabilities[0].AvailableQuantity.toString() : "0";
                }
            },

            {
                title: "Reserved",
                ratio: 23,
                collapseOrder: 1,
                minWidth: 60,
                computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                    return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
value.ItemAvailabilities[0].PhysicalReserved.toString() : "0";
                }
            },

            {
                title: "Ordered",
                ratio: 23,
                collapseOrder: 2,
                minWidth: 60,
                computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                    return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
value.ItemAvailabilities[0].OrderedSum.toString() : "0";
                }
            }
        ],

        itemDataSource: this.orgUnitAvailabilities,
        selectionMode: SelectionMode.None
    });
}

```

13. Add the **OnReady** method to bind the HTML control.



```

public onReady(element: HTMLElement): void {

    ko.applyBindingsToNode(element, {
        template: {
            name: ProductAvailabilityPanel.TEMPLATE_ID,
            data: this
        }
    });
}

```

14. Add the **init** method to get the product availability details so when the page loads, the data is fetched and updated in the data list.

```

public init(state: ISimpleProductDetailsCustomControlState): void {

    this._state = state;
    let correlationId: string = this.context.logger.getNewCorrelationId();
    if(!this._state.isSelectionMode) {
        this.isVisible = true;

        let request: InventoryLookupOperationRequest<InventoryLookupOperationResponse> =
            new InventoryLookupOperationRequest<InventoryLookupOperationResponse>
                (this._state.product.RecordId, correlationId);
        this.context.runtime.executeAsync(request)
            .then((result: ClientEntities.ICancelableDataResult<InventoryLookupOperationResponse>) => {

                if (!result.canceled) {
                    this.orgUnitAvailabilities(result.data.orgUnitAvailability);
                }

            }).catch((reason: any) => {
                this.context.logger.logError(JSON.stringify(reason), correlationId);
            });
    }
}

```

The entire code example is shown below.

```

import {
    SimpleProductDetailsCustomControlBase,
    ISimpleProductDetailsCustomControlState,
    ISimpleProductDetailsCustomControlContext
} from "PosApi/Extend/Views/SimpleProductDetailsView";

import { InventoryLookupOperationRequest, InventoryLookupOperationResponse } from
"PosApi/Consume/OrgUnits";
import { ClientEntities, ProxyEntities } from "PosApi/Entities";
import { ArrayExtensions } from "PosApi/TypeExtensions";
import { DataList, SelectionMode } from "PosUISdk/Controls/DataList";
export default class ProductAvailabilityPanel extends SimpleProductDetailsCustomControlBase {

    private static readonly TEMPLATE_ID: string =
"Microsot_Pos_Extensibility_Samples_ProductAvailabilityPanel";
    public readonly orgUnitAvailabilities: ObservableArray<ProxyEntities.OrgUnitAvailability>;
    public readonly dataList: DataList<ProxyEntities.OrgUnitAvailability>;
    public readonly title: Observable<string>;
    private _state: ISimpleProductDetailsCustomControlState;

    constructor(id: string, context: ISimpleProductDetailsCustomControlContext) {
        super(id, context);
    }
}

```

```

this.orgUnitAvailabilities = ko.observableArray([]);
this.title = ko.observable("Product Availability");
this.dataList = new DataList<ProxyEntities.OrgUnitAvailability>({

    columns: [
        {
            title: "Location",
            ratio: 31,
            collapseOrder: 4,
            minWidth: 100,
            computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                return value.OrgUnitLocation.OrgUnitName;
            }
        },
        {
            title: "Inventory",
            ratio: 23,
            collapseOrder: 3,
            minWidth: 60,
            computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
                    value.ItemAvailabilities[0].AvailableQuantity.toString() : "0";
            }
        },
        {
            title: "Reserved",
            ratio: 23,
            collapseOrder: 1,
            minWidth: 60,
            computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
                    value.ItemAvailabilities[0].PhysicalReserved.toString() : "0";
            }
        },
        {
            title: "Ordered",
            ratio: 23,
            collapseOrder: 2,
            minWidth: 60,
            computeValue: (value: ProxyEntities.OrgUnitAvailability): string => {
                return ArrayExtensions.hasElements(value.ItemAvailabilities) ?
                    value.ItemAvailabilities[0].OrderedSum.toString() : "0";
            }
        }
    ],

    itemDataSource: this.orgUnitAvailabilities,
    selectionMode: SelectionMode.None
});

}

/**
 * Binds the control to the specified element.
 * @param {HTMLElement} element The element to which the control should be bound.
 */

public onReady(element: HTMLElement): void {
    ko.applyBindingsToNode(element, {
        template: {
            name: ProductAvailabilityPanel.TEMPLATE_ID,
            data: this
        }
    }
}

```

```

    });

}

/**
 * Initializes the control.
 * @param {ISimpleProductDetailsCustomControlState} state The initial state of the page used to
initialize the control.
 */

public init(state: ISimpleProductDetailsCustomControlState): void {
    this._state = state;
    let correlationId: string = this.context.logger.getNewCorrelationId();
    if (!this._state.isSelectionMode) {
        this.isVisible = true;
        let request: InventoryLookupOperationRequest<InventoryLookupOperationResponse> =
            new InventoryLookupOperationRequest<InventoryLookupOperationResponse>
                (this._state.product.RecordId, correlationId);
        this.context.runtime.executeAsync(request)
            .then((result:
ClientEntities.ICancelableDataResult<InventoryLookupOperationResponse>) => {
                if (!result.canceled) {
                    this.orgUnitAvailabilities(result.data.orgUnitAvailability);
                }
            })
            .catch((reason: any) => {
                this.context.logger.logError(JSON.stringify(reason), correlationId);
            });
    }
}
}
}

```

15. Create a new .json file and under the **ProdDetailsCustomColumnExtensions** folder and name it **manifest.json**.

16. In the **manifest.json** file, add the following code.

```

{
  "$schema": "../manifestSchema.json",
  "name": "Pos_Extensibility_Samples",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
  "extend": {
    "views": {
      "SimpleProductDetailsView": {
        "controlsConfig": {
          "customControls": [
            {
              "controlName": "productAvailabilityPanel",
              "htmlPath":
"ViewExtensions/SimpleProductDetails/ProductAvailabilityPanel.html",
              "modulePath":
"ViewExtensions/SimpleProductDetails/ProductAvailabilityPanel"
            }
          ]
        }
      }
    }
  }
}

```

17. Open the **extensions.json** file under the **POS.Extensions** project and add the **ProdDetailsCustomColumnExtensions** samples, so during runtime POS will include the extension.

```

{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions2"
    },
    {
      "baseUrl": "ProdDetailsCustomColumnExtensions"
    }
  ]
}

```

18. Open the **tsconfig.json** and comment out the extension package folders from the exclude list. POS uses this file to include or exclude extensions. By default, the list contains the excluded extensions list. If you want to include any extension part of the POS, then you need add the extension folder name and comment out the extension from the extension list as shown.

```

"exclude": [
  "AuditEventExtensionSample",
  "B2BSample",
  "CustomerSearchWithAttributesSample",
  "FiscalRegisterSample",
  "PaymentSample",
  "PromotionsSample",
  "SalesTransactionSignatureSample",
  "SampleExtensions",
  //"SampleExtensions2",
  //"ProdDetailsCustomColumnExtensions"
],

```

19. Compile and rebuild the project.

## Validate the customization

1. Press F5 and deploy the POS to test your customization.
2. After POS launches, login to POS. Search for any product and navigate to the product details view. You should see the custom control that you added.

### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Run the point of sale (POS) samples

2/18/2021 • 2 minutes to read • [Edit Online](#)

There are several samples in the Retail SDK that demonstrate extensions. This topic explains how to run these samples.

## Run the SampleExtensions in POS

1. Open **ModernPos.Sln** or **CloudPos.sln** from the **Retail SDK\POS** folder.
2. Select the **POS.Extensions** project in the solution and click **Show All Files**.
3. Right-click the **SampleExtensions** folder and select **Include in Project**.
4. Right-click the **SampleExtensions2** folder and select **Include in Project**.
5. Open the **extensions.json** file and add the extension folder for **SampleExtensions** and **SampleExtensions2**. This means that during runtime POS will include this extension. The **baseUrl** value must exactly match the relative path and extension folder name.

```
{
  "extensionPackages": [
    {
      "baseUrl": "SampleExtensions"
    },
    {
      "baseUrl": "SampleExtensions2"
    }
  ]
}
```

### NOTE

In the extension.json file you must include at least two extension folders. If you add only one extension folder, then POS will not load the extension.

6. Open the **tsconfig.json** file and comment out the extension package folders from the exclude list. POS will use this file to determine whether to compile the extension. By default, the list contains the sample extensions list. If you want to compile any extension to the POS, then you need add the extension folder name and comment out the extension from the extension as shown below.

```

{
  "extends": "../tsconfigs/tsmodulesconfig",
  "exclude": [
    "AuditEventExtensionSample"
    , "B2BSample"
    ,"CustomerSearchWithAttributesSample"
    ,"FiscalRegisterSample"
    ,"PaymentSample"
    ,"PromotionsSample"
    ,"SalesTransactionSignatureSample"
    // ,"SampleExtensions"
    // ,"SampleExtensions2"
    ,"StoreHoursSample"
    ,"SuspendTransactionReceiptSample"
  ],
  "compilerOptions": {
    // There is an unexpected behavior for TypeScript 2.2.2 in map and source roots generated in
    compiled JS and map files.
    // The following may change in future TypeScript versions.
    // In case there is only one top level extensions folder with .ts files included, the
    following two root
    // directories need to be changed to include the extensions folder.
    // For example, change both roots to "./SampleExtensions" if "SampleExtensions" folder is the
    only top level
    // folder that has .ts files included in the project.
    // That is, either "SampleExtensions" folder is the only top level folder, or all other top
    level folders
    // have .js files only, no .ts files.
    "mapRoot": "./", /* Cannot be specified in base file. Adds full path to ".map" in the js file
    to enable debug in VS. */
    "sourceRoot": "./" /* Cannot be specified in base file. Adds full path to ".ts" in the map
    file to enable debug in VS. */
  }
}

```

If you want to enable other extensions, comment them out from the exclude list. For example, if you want to include **B2BSample**, the code would be as follows.

```

"exclude": [
  "AuditEventExtensionSample"
  // ,"B2BSample"
  ,"CustomerSearchWithAttributesSample"
  ,"FiscalRegisterSample"
  ,"PaymentSample"
  ,"PromotionsSample"
  ,"SalesTransactionSignatureSample"
  // ,"SampleExtensions"
  // ,"SampleExtensions2"
  ,"StoreHoursSample"
  ,"SuspendTransactionReceiptSample"
],

```

#### NOTE

Other extension package folders, even though not included in the Visual Studio project, should be kept in the exclude list if they are not meant to be included.

7. Set **Solution platform** to **x86** and **Deploy option** to **Local Machine** or **Simulator** if you are using Modern POS for validation.
8. Click **Save all** and press the **F5** key to validate the extensions.

**NOTE**

For cloud POS, use a clean solution in Visual Studio, and then rebuild the solution.

9. Go to the product search screen or use the top search bar to search for a product. You should see custom columns in the grid and new app bar buttons.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Device management implementation guidance

2/18/2021 • 13 minutes to read • [Edit Online](#)

This topic is intended for people who implement functionality that is related to device management in a commerce environment. It gives implementation tips and guidance that you should consider as you plan your implementation.

## Overview

A *device* is an instrument on which a point of sale (POS) application can be installed, configured, and used to perform operations that are required in order to run or help run the business that owns that instrument. In other words, a device is a piece of technology that runs a POS application to help run a business. This business doesn't have to be exclusively a commerce operation. For example, a hospital has a gift shop, a warehouse manages inventory, and a law firm generates invoices. What is important is that the application that the device runs makes business operations simpler, more efficient, or just better managed and recorded. Regardless of the scenario that they are used in, devices are critical elements. As the number of devices that are used increases, it becomes more valuable to put processes in place to track and manage those devices. Together, these processes are known as *device management*.

The POS application that this topic discusses is Modern POS (MPOS) in Microsoft Dynamics 365 Commerce. MPOS lets users complete business operations quickly, efficiently, and simply. It also helps the organization manage the many devices that run MPOS across the business. Although this topic discusses many aspects of an MPOS device, the two critical aspects are the business-oriented *register* and the physical concept of a *device*.

The **Registers** page is the virtual tracking mechanism for the business-oriented details of an instance of the MPOS application. These details include the visual profile that is used, the automatic sign-out time, and the store that the register is a part of. All these details are stored in the virtual register. When the register is correctly set up and configured, it's linked to a virtual device.

The **Devices** page is the virtual tracking mechanism for the physical concept of a device. The business-oriented register is linked here to complete the configuration and prepare for installation. The virtual device stores details such as the validation status, when the POS application was activated, and the version that will be installed or the version that is currently installed.

As more registers are generated and then linked to the appropriate devices, both physical and virtual management become critically important. For example, a business has one physical device, a Microsoft Surface that will run MPOS. The Surface must be configured for the business, because it might be on a domain or require additional software that is specific to that business. MPOS must also be installed and activated on the Surface. Over time, the device might require servicing or replacement, or it might even be stolen. If the business doesn't have just one device, but dozens, hundreds, or even thousands, processes must be put in place to appropriately watch, manage, and verify the status of all the devices. These devices include both devices that are currently used and devices that aren't currently used.

Commerce already provides the basic requirements for device management. As you plan your implementation, you must make sure that you take all the appropriate considerations, to help minimize pain and maximize benefit.

## Implementation considerations

This section describes some things that you should consider as you plan to implement features that are related to device management in your retail store and distribution locations.

## Generate the physical topology

Planning is the most critical requirement for successful implementation of an enterprise resource planning (ERP) solution. One of the main deliverables of this planning should be a *physical topology*. The physical topology is a visualization of many details about a company, from the lowest POS device to the highest network connections at the headquarters. At a minimum, the following deliverables should be completed:

- **Store templates** – A store template consists of one or more diagrams that show the layout of a physical location, such as a brick-and-mortar store. In the long term, these diagrams help you easily implement future locations and quickly assess how you can fix or improve any issues that are found. The store template should include the following details:
  - **Location** – The physical layout of a location. Details should include the position of all devices that exist in the location.
  - **Network** – The internal layout of the network infrastructure and details about internet connectivity (for example, bandwidth up and down, thread count, and latency to the headquarters or other important internet locations).
  - **Device** – The details of all devices that exist in a location. These details include the device specifications or some type of associable tag that lets you quickly find the details about a device.
  - **Peripheral** – A list of all peripherals that are attached to a device, a count of peripherals, and physical location of all peripherals. Often, there should also be an associable tag that lets you quickly find details.
- **Naming methodology** – For any implementation, it's important that you maintain naming conventions across all devices. Generate the rules that govern the naming conventions, and follow those rules.

### NOTE

When you generate the naming methodology, we recommend that the name of a register be the same as, or very similar to, the name of the device. The name of the device, in turn, should be the same as, or very similar to, the friendly name of the physical computer that the register works on.

- **Procedures plan** – The more a business grows, the more important it becomes that procedures remain in place to help maintain enough order so that the business can run efficiently. You should consider your plan a living document that you must be maintained and appended often. This plan should explain how to perform almost every action that will be repeated in the stores that make up a company.
- **Servicing plan** – As business continues, servicing becomes more critical. You must decide when you will replace devices, when you will update devices and peripherals (both the applications and the operating systems), and how you will perform these tasks. The answers will vary from business to business, but these tasks should be planned for.
- **Disaster plan** – If something can go wrong, it should be planned for. As planning proceeds, make a note of risks and potential mitigations. Generate a plan that explains how to fix a problematic situation.

After planning is completed and all potential details are known, you will eventually have a list of stores and, for each store, a list of associated registers and devices. The next question that you must answer is how you will deploy all the various devices that are listed.

## Distributed deployment

As the number of devices increases into the hundreds or even thousands, manual configuration, installation, and activation of the POS application quickly becomes impractical. There are several concepts that you can use to alleviate this issue and to manage or help manage devices on a large scale:

- **Devices page** – When you use MPOS, the page that shows the details of a device includes information about the POS client that should be used on the device. Under the **Register package** subheading, there are

four fields. The first three fields (**Package name**, **Package description**, and **Version number**) provide information about the version of the MPOS package that should be installed on the device either now or later. The fourth field shows the version of MPOS that is currently installed. Therefore, you can easily find this information for any device.

- **Channel deployment workspace** – This workspace lets you quickly view a large, filterable set of stores, registers, devices, and more. This workspace also provides options for mass deployment. (For more information, see [Customer orders in Modern POS \(MPOS\)](#)). This workspace gives you quick access to important pages, can help decrease validation times when you do status checks after a deployment, and helps you efficiently manage device statuses.
- **Mass deployment** – Dynamics 365 customers who use MPOS and other client-side components can silently perform various actions to help with installation, configuration, and servicing. By running basic commands at a command prompt, you might be able to deploy and service (that is, update) commerce components. This basic method of skipping the installer's manual user interface can reduce the time that is required for installation and servicing. The log files remain the same and can be viewed for installation details.
- **Scripting** – Based on the mass deployment functionality, scripts can be generated and set up to run automatically to install or update commerce components such as MPOS. These basic scripts can be entered as a scheduled task on a device. The task then runs at a predetermined time and return the logs and results to a predetermined location, where they can be viewed as time permits.
- **Systems management solution** – A systems management solution can help increase the amount of known data about the devices that are used. It can also decrease the time that is required in order to get that data. Examples of systems management solutions include Microsoft System Center and Microsoft Intune. By using a systems management solution together with mass deployment and scripting, you can configure and install components much more quickly. In addition, you can more efficiently validate status after deployment or servicing. Microsoft has published a document specifically about device management via System Center Configuration Manager, [Choose a device management solution for System Center Configuration Manager](#).

For more information that will help you better understand distributed deployment, see [Configure, install, and activate Modern POS \(MPOS\)](#) and [Customer orders in Modern POS \(MPOS\)](#).

## Servicing devices

Although deployment is an important topic that has many nuances, the ongoing nature of business requires that you consider servicing early, and that you plan for it. In that way, you can maximize efficiency and speed of completion every time that servicing is required. Servicing becomes even more important when you understand that, not only do applications have to be serviced, but the operating system and peripherals might also require updates. As we mentioned earlier, this process can be even more difficult when it's time to replace devices. In general, you should consider all the following factors:

- **Devices** – The process of servicing a device involves more than just determining whether servicing has been completed. Instead, many aspects of the device must be reviewed and updated, either individually or as part of a group of devices at the same time:
  - The information on the **Devices** page, or in the **Channel deployment** workspace, in Dynamics 365 headquarters can help you validate the current status and version of MPOS or other self-service components, such as hardware station or Commerce Scale Unit.
  - It's very useful if you know the version of Microsoft Windows. In scenarios where a systems management solution isn't used, you can use the **winver** command in a Command Prompt window to quickly learn the specific version of Windows that is currently installed. By comparing the version number against the [Windows version list](#), you can easily learn what updates a computer is missing at a service pack level.
  - Internal and peripheral drivers must be checked for version updates too. You should include and monitor this information in the device and peripheral lists that you created as part of the store templates during the planning phase, as explained earlier in this topic.

- Always track when a computer goes out of service (warranty or service plan). In this way, you can quickly determine when a computer should be replaced instead of being sent out for servicing (warranty or service plan).
- **Peripherals** – The peripherals that are attached to a device or to the network must be monitored. As part of this process, you must also monitor the network devices themselves. (Routers, switches, firewalls, and all other network devices also have firmware that requires occasional updates to maintain security and compatibility, and to comply with servicing terms.) If appropriate planning has occurred, the process of watching for service dates and correctly updating peripherals should become a simplified task within the larger servicing flow for device management.
- **Replacements** – Eventually, devices and peripherals either fail or reach a point where servicing just isn't enough. In general, we highly recommended that you have a replacement system that is ready before this situation arises. Based on the procedures and servicing plans that you created during the planning phase, this process can be very fast and efficient. For a device that runs MPOS, you can generically prepare the replacement device and install MPOS on it long before you either send the unit on-site or store it on-site, depending on the approach that works best for the business. At the time of replacement, MPOS can "reactivate" a device that is already active. Therefore, you can handle special cases where a system stops responding and must be replaced, because it can't be repaired. However, you must make sure that the replacement is correctly up to date, and that it has recently been serviced.
- **Systems management solution** – A systems management solution can dramatically improve how servicing works and when servicing is done. These solutions include telemetry tools that monitor all the systems that are being used. Because of these telemetry tools and the device information data that is constantly stored and available, servicing should become proactive instead of reactive. In this best-case scenario, appropriate plans exist for procedures and servicing, and the alerts and the known data about a system can let you know about impending servicing or an impending replacement before an issue occurs and becomes a critical task.

### **Worst-case scenarios**

Unfortunately, things sometimes go wrong. It's important that you consider worst-case scenarios and put mitigation plans in place (per the disaster plan that you created during the planning phase) to help resolve issues that arise. In Commerce, the **Devices** page can at least help with device management.

- **Lost/stolen device** – When a device is lost or stolen, it becomes a critical issue, because private data can end up in the hands of a malicious person. The first step is to deactivate any lost or stolen device on the **Devices** page in Headquarters, to immediately prevent sign-on to that device. If you created a disaster plan, follow the rest of the guidelines to complete all tasks that are required in order to tag the device as gone, file all important documents (which might include documents for insurance and the police), and work to replace the device and continue business. If appropriate procedures have been created, a replacement device should already be ready.
- **Infrastructure issues** – Network equipment might go bad, the store layout might have to be altered, or, as a part of a store update, the devices might be changed from desktops to tablets. When issues arise that can affect the management of devices, it can be difficult to react appropriately and fix the issues. In the best-case scenario, you've already discussed these issues and listed them in the procedures, servicing, and disaster plans. At a minimum, before something is done that affects the layout of a business location, you should update the store layout documents to reflect the new layout, so that fewer surprises are met. As we mentioned earlier, for issues that affect MPOS specifically, the ability to reactivate should help mitigate downtime and should help with change management. If you're concerned about outages of network equipment, or if outages occur, the MPOS client can support offline databases. This functionality can help mitigate the impact that outages have on business.
- **Disaster** – When a disaster occurs, everything is affected. If the disaster is environmental, the employees should always be the highest priority. However, eventually, you must handle the management of the internal systems, and those systems must be repaired. Unfortunately, no simple advice or specific considerations can

be listed. Instead, you should plan early in a more general way. What sorts of disasters could affect the business? Utilities such as water and power can have devastating effects on equipment. How can you mitigate the effects on employees, locations, and equipment in various disasters? Keep in mind concepts such as the data that is stored in the location, and when and how it's backed up and secured.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# POS Cart view events and handlers

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how extensions can consume the point of sale (POS) view events and handlers for custom scenarios. For example, the Cart view in POS exposes multiple events and handlers so that your extensions can perform custom business flows, based on events. The extensions can subscribe to an event and receive a notification when that event occurs.

## Cart view event handlers

The base class that is used to consume the Cart view handlers is **CartExtensionViewControllerBase**.

```
export default class CartViewController extends CartView.CartExtensionViewControllerBase {  
}
```

BASE CLASS	EVENT/HANDLER	DESCRIPTION
CartExtensionViewControllerBase	cartLineSelectedHandler: (data: CartLineSelectedData) => void;	The handler for the message that appears when a cart line is selected.
	cartLineSelectionClearedHandler: () => void;	The handler for the message that appears when the selection of a cart line is cleared.
	tenderLineSelectedHandler: (data: TenderLineSelectedData) => void;	The handler for the message that appears when a tender line is selected.
	protected tenderLineSelectionClearedHandler: () => void;	The handler for the message that appears when the selection of a tender line is cleared.
	protected cartChangedHandler: (data: CartChangedData) => void;	The handler for the message that appears when the cart is changed.
	protected processingAddItemOrCustomerChangedHandler: (processing: boolean) => void;	The handler that adds the message about the item or customer processing state.
	protected setSelectedCartLines(setSelectedCartLinesData: SetSelectedCartLinesData): void;	The handler that selects the cart lines that are shown on the page.

The following example shows how to consume Cart view events.

```

import { ProxyEntities } from "PosApi/Entities";
import { IExtensionCartViewControllerContext } from "PosApi/Extend/Views/CartView";
import * as CartView from "PosApi/Extend/Views/CartView";
import { ArrayExtensions, StringExtensions } from "PosApi/TypeExtensions";

export default class CartViewController extends CartView.CartExtensionViewControllerBase {

    public static selectedCartLineId: string = StringExtensions.EMPTY;

    private _selectedCartLines: ProxyEntities.CartLine[];
    private _selectedTenderLines: ProxyEntities.TenderLine[];
    private _isProcessingAddItemOrCustomer: boolean;

    /**
     * Creates a new instance of the CartViewController class.
     * @param {IExtensionCartViewControllerContext} context The events Handler context.
     * @remarks The events handler context contains APIs through which a handler can communicate with POS.
     */
    constructor(context: IExtensionCartViewControllerContext) {
        super(context);

        this.cartLineSelectedHandler = (data: CartView.CartLineSelectedData): void => {
            this._selectedCartLines = data.cartLines;

            if (ArrayExtensions.hasElements(this._selectedCartLines)) {
                CartViewController.selectedCartLineId = this._selectedCartLines[0].LineId;
            }
        };

        this.cartLineSelectionClearedHandler = (): void => {
            this._selectedCartLines = undefined;
            CartViewController.selectedCartLineId = null;
        };

        this.tenderLineSelectedHandler = (data: CartView.TenderLineSelectedData): void => {
            this._selectedTenderLines = data.tenderLines;
        };

        this.tenderLineSelectionClearedHandler = (): void => {
            this._selectedCartLines = undefined;
        };

        this.processingAddItemOrCustomerChangedHandler = (processing: boolean): void => {
            this._isProcessingAddItemOrCustomer = processing;
        };
    }
}

```

## Base classes for consuming Cart view events in Cart view UI extensions

There are several base classes for consuming events in the user interface (UI):

- Custom controls
- Custom fields in the **Totals** pane
- Custom columns in the **Lines** grid
- Custom columns in the **Payment** grid
- Custom columns in the **Delivery** grid

### Custom controls

BASE CLASS	EVENT	DESCRIPTION
CartViewCustomControlBase	cartLineSelectedHandler: (data: CartLineSelectedData) => void;	The handler for the message that appears when a cart line is selected.
	cartLineSelectionClearedHandler: () => void;	The handler for the message that appears when the selection of a cart line is cleared.
	tenderLineSelectedHandler: (data: TenderLineSelectedData) => void;	The handler for the message that appears when a tender line is selected.
	protected tenderLineSelectionClearedHandler: () => void;	The handler for the message that appears when the selection of a tender line is cleared.
	protected cartChangedHandler: (data: CartChangedData) => void;	The handler for the message that appears when the cart is changed.
	protected processingAddItemOrCustomerChangedHandler: (processing: boolean) => void;	The handler that adds the message about the item or customer processing state.
	protected setSelectedCartLines(setSelectedCartLinesData: SetSelectedCartLinesData): void;	The handler that selects the cart lines that are shown on the page.

### Custom fields in the Totals pane

BASE CLASS	EVENT	DESCRIPTION
CartViewTotalsPanelCustomFieldBase	public computeValue(cart: ProxyEntities.Cart): string { }	Compute the value for the custom field.

### Custom columns in the Lines grid

BASE CLASS	EVENT	DESCRIPTION
CustomLinesGridColumnBase	public title(): string { }	Set the title for the custom column.
	public computeValue(cartLine: ProxyEntities.CartLine): string { }	Compute the value for the custom column.
	public alignment(): CustomGridColumnAlignment { } <b>Supported values:</b> enum CustomGridColumnAlignment { Left = 0, Right = 1 }	Set left or right alignment for the custom column.

### Custom columns in the Payment grid

BASE CLASS	EVENT	DESCRIPTION
CustomPaymentsGridColumnBase	public title(): string { }	Set the title for the custom column.



BASE CLASS	EVENT	DESCRIPTION
	<pre>public computeValue(cartLine: ProxyEntities.CartLine): string { }</pre>	Compute the value for the custom column.
	<pre>public alignment(): CustomGridColumnAlignment { } <b>Supported values:</b> enum CustomGridColumnAlignment { Left = 0, Right = 1 }</pre>	Set left or right alignment for the custom column.

### Custom columns in the Delivery grid

BASE CLASS	EVENT	DESCRIPTION
CustomDeliveryGridColumnBase	<pre>public title(): string { }</pre>	Set the title for the custom column.
	<pre>public computeValue(cartLine: ProxyEntities.CartLine): string { }</pre>	Compute the value for the custom column.
	<pre>public alignment(): CustomGridColumnAlignment { } <b>Supported values:</b> enum CustomGridColumnAlignment { Left = 0, Right = 1 }</pre>	Set left or right alignment for the custom column.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Add custom buttons to the POS header bar

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how to add a new custom button to the header bar in the point of sale (POS). The POS header bar extension is supported by Microsoft Dynamics 365 Commerce version 10.0.14 and later.

The custom header button must contain an HTML file that includes Cascading Style Sheets (CSS) code that describes the control user interface (UI), styles, and themes. It must also contain a TypeScript view model file that specifies the logic. The class inside the file view model file must extend the **CustomPackingItem** class, so that it inherits the header button properties and events. The **cartChangedHandler** event is exposed on the header button to provide notification if something changes in the cart. Your extension code can do custom logic that is based on cart events, or it can do custom business logic.

## CustomPackingItem class

### Properties

PROPERTY	DESCRIPTION
CustomPackingItemPosition	The item's position relative to the POS header items.
ICustomPackingItemContext	The context of the custom header packing item.

### Events and methods

EVENT OR METHOD	DESCRIPTION
cartChangedHandler	The handler for the <b>CartUpdated</b> event.
constructor(id: string, context: ICustomPackingItemContext)	This method creates a new instance of the <b>CustomHeaderPackingItem</b> class.
id()	This method gets the identifier of the custom header packing item.
get visible(): boolean	This method gets the visible value.
set visible(isVisible: boolean)	This method sets the visible value.
abstract onReady(packedElement: HTMLElement, unpackedElement: HTMLElement): void	This method is called when the control element is ready.
dispose(): void	This method disposes of the control and releases its resources.
protected abstract init(state: ICustomPackingItemState): void	This method initializes the control.

You must add nodes for the header button extension in the **manifest.json** file, as shown in the following example.

```

"header": {
  "customPackingItems": [
    {
      "name": " name of the control",
      "description": "Description.",
      "modulePath": " view model file name with path",
      "htmlPath": " html file name with path"
    }
  ]
}

```

## Add a custom button to the POS header bar

Follow these steps to add a custom button to the header bar and show the amount due by reading it from the cart.

1. Open Visual Studio 2017.
2. Open the **ModernPOS/CloudPOS** solution from **\RetailSDK\POS**.
3. In the **POS.Extensions** project, create a folder that is named **HeaderExtensionSample**.
4. In the **HeaderExtensionSample** folder, create an HTML file that is named **CartAmountDuePackingItem.html**.
5. Copy the following code, and paste it into the file.

```

<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Cart Amount Due Item Templates</title>
</head>
<body>
  <script id="Microsoft_Pos_Extensibility_Samples_UnpackedCartAmountDueItem" type="text/html">
    <button class="pad0 center row"
      data-bind="click: onItemClickHandler">
      <div class="iconShop buttonIcon height20"></div>
      <div class="h4 padRight8" data-bind="text: amountDueLabel"></div>
    </button>
  </script>
  <script id="Microsoft_Pos_Extensibility_Samples_PackedCartAmountDueItem" type="text/html">
    <button class="row pad0"
      data-bind="click: onItemClickHandler">
      <div class="iconShop buttonIcon height20 margin12">
      </div>
      <div class="grow row marginLeft8 padTop12">
        <div class="h4 padRight4">Amount due: </div>
        <div class="h4 padRight8" data-bind="text: amountDueLabel"></div>
      </div>
    </button>
  </script>
</body>
</html>

```

6. In the **HeaderExtensionSample** folder, create a TypeScript file that is named **CartAmountDuePackingItem.ts**.
7. Copy the following code, and paste it into the file.

```

import {
  CustomPackingItem, ICustomPackingItemContext, CustomPackingItemPosition, ICustomPackingItemState,

```

```

CartChangedData
} from "PosApi/Extend/Header";
import { CurrencyFormatter } from "PosApi/Consume/Formatters";
import { ClientEntities } from "PosApi/Entities";
import { StringExtensions } from "PosApi/TypeExtensions";

/**
 * (Sample) Custom packing item that shows the cart's amount due and navigates to the cart on click.
 */
export default class CartAmountDuePackingItem extends CustomPackingItem {
    /**
     * The position of the custom packing item relative to the out-of-the-box items.
     */
    public readonly position: CustomPackingItemPosition = CustomPackingItemPosition.After;

    /**
     * Label displayed in the custom packing item with the current amount due.
     */
    public amountDueLabel: Observable<string>;

    private _currentAmountDue: Observable<number>;
    private _amountDueSubscription: IDisposable;

    /**
     * Initializes a new instance of the CartAmountDuePackingItem class.
     * @param {string} id The item identifier.
     * @param {ICustomPackingItemContext} context The custom packing item context.
     */
    constructor(id: string, context: ICustomPackingItemContext) {
        super(id, context);

        this.amountDueLabel = ko.observable(StringExtensions.EMPTY);

        this._currentAmountDue = ko.observable(0);
        this._amountDueSubscription = this._currentAmountDue.subscribe((newValue: number) => {
            if (newValue > 0) {
                this.amountDueLabel(CurrencyFormatter.toCurrency(newValue));
                this.visible = true;
            } else {
                this.visible = false;
            }
        });

        this.cartChangedHandler = this._cartChangedHandler.bind(this);
    }

    /**
     * Called when the control element is ready.
     * @param {HTMLElement} packedElement The DOM element of the packed element.
     * @param {HTMLElement} unpackedElement The DOM element of the unpacked element.
     */
    public onReady(packedElement: HTMLElement, unpackedElement: HTMLElement): void {
        this.context.logger.logInformational("Executing onReady!");
        ko.applyBindingsToNode(unpackedElement, {
            template: {
                name: "Microsoft_Pos_Extensibility_Samples_UnpackedCartAmountDueItem",
                data: this
            }
        });

        ko.applyBindingsToNode(packedElement, {
            template: {
                name: "Microsoft_Pos_Extensibility_Samples_PackedCartAmountDueItem",
                data: this
            }
        });
    }
}

/**

```

```

    * Initializes the control.
    * @param {ICustomPackingItemState} state The custom control state.
    */
    public init(state: ICustomPackingItemState): void {
        return;
    }

    /**
     * Disposes the control releasing its resources.
     */
    public dispose(): void {
        this._amountDueSubscription.dispose();
        super.dispose();
    }

    /**
     * Method used to handle the onClick of the custom packing item.
     */
    public onItemClickClickedHandler(): void {
        const correlationId: string = this.context.logger.getNewCorrelationId();
        let cartViewOptions: ClientEntities.CartViewNavigationParameters = new
        ClientEntities.CartViewNavigationParameters(correlationId);

        this.context.logger.logInformational("Cart amount due packing item clicked.", correlationId);

        this.context.navigator.navigateToPOSVIEW("CartView", cartViewOptions);
    }

    /**
     * Handler for the cart changed event.
     * @param {CartChangedData} data The data sent with the event.
     */
    private _cartChangedHandler(data: CartChangedData): void {
        this.context.logger.logInformational("CartChanged received.");
        this._currentAmountDue(data.cart.AmountDue);
    }
}

```

8. In the **HeaderExtensionSample** folder, create a JavaScript Object Notation (JSON) file that is named **manifest.json**.

9. Copy the following code, and paste it into the file.

```

{
  "$schema": "../schemas/manifestSchema.json",
  "name": "HeaderExtensionSample ",
  "publisher": "Microsoft",
  "version": "10.0.14",
  "minimumPosVersion": "9.24.0.0",
  "components": {
    "extend": {
      "header": {
        "customPackingItems": [
          {
            "name": " CartAmountDuePackingItem",
            "description": "An item showing amount due.",
            "modulePath": " CartAmountDuePackingItem",
            "htmlPath": " CartAmountDuePackingItem.html"
          }
        ]
      }
    }
  }
}

```

10. In the `POS.Extensions` project, open the `extensions.json` file.
11. Update the details of the `HeaderExtensionSample` package, so that the POS can include this extension package during the initial load.

```
{
  "extensionPackages": [
    {
      "baseUri": "HeaderExtensionSample"
    }
  ]
}
```

12. Build the project.

### Validate the customization

Follow these steps to validate the customization.

1. Sign in to the POS by entering the operator ID and password.
2. Look at the header bar. The custom button that you added should be visible.

You can view the status of the extension package on the [POS settings page](#).

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Hardware Station extensibility

2/18/2021 • 3 minutes to read • [Edit Online](#)

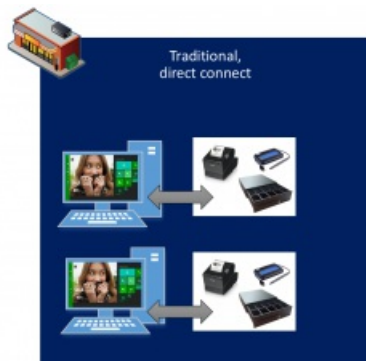
## NOTE

This topic is applicable for Dynamics 365 for Finance and Operations version 7.1 and earlier. This implementation is not supported for versions 7.2 and higher. For those versions, follow the extension model without overlayering.

This topic explains how to extend Hardware Station to add support for new devices and new device types for existing devices.

## Hardware Station overview

Hardware Station is used by Modern POS and Cloud POS to connect to hardware peripherals, such as printers, cash drawers, scanners, and payment terminals.



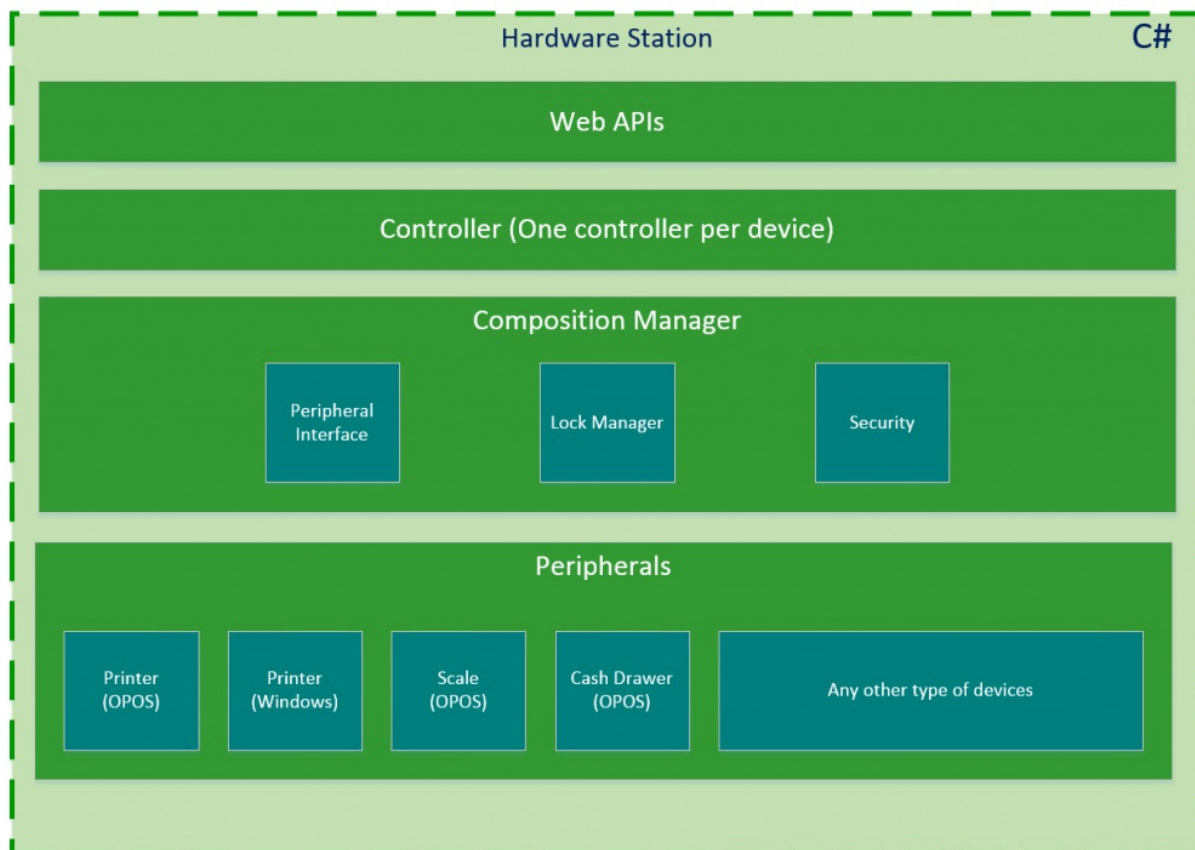
## Hardware Station setup

Before you start, use the information in [Configure and install Retail hardware station](#) to install Hardware Station, and to get a feel of what hardware is and how it's installed.

## Hardware Station architecture

Hardware Station exposes Web API for Hardware Station application programming interfaces (APIs). Hardware Station can be extended either by implementing a new controller for a new device (for example, a cash dispenser) or by overriding an existing controller for an existing device type (for example, a new Audio Jack magnetic stripe reader (MSR) implementation).

# Hardware Station Architecture



## Hardware Station extensibility scenarios

Extensibility in Hardware Station is achieved by using [Managed Extensibility Framework \(MEF\)](#), which is supported by .NET. **Extensibility guideline:** Always write your extension in your own extension assembly. That way, you're writing a true extension, and upgrades will be much easier. There are two basic scenarios for extension:

- **Adding a new device** – The out-of-box Hardware Station doesn't already support the device (for example, a cash dispenser). Therefore, you must add support for the new device in Hardware Station.
- **Adding a new device type for an existing device** – The out-of-box Hardware Station implementation already supports the device (for example, an MSR), but you must add support for a specific device type (for example, an Audio Jack MSR implementation).

### Scenario 1: Adding a new device

For this scenario, we will add support for a cash dispenser device in Hardware Station. In our example, we will create a fake cash dispenser that dispenses cash in the Notepad file. However, this example will help you understand the end-to-end extensibility of Hardware Station.

- The Retail software development kit (SDK) has a cash dispenser sample. See `RetailSdk\SampleExtensions\HardwareStation`.
- In this case, we must add a new Web API controller and helper properties/methods.
- The new `CashDispenser` controller must extend `ApiController` and `IHardwareStationController`.
- The `Export` attribute string here specifies the device that this controller is used for:  
`[Export("CASHDISPENSER", typeof(IHardwareStationController))]`



```

namespace Contoso
{
    namespace Commerce.HardwareStation.CashDispenserSample
    {
        using System;
        using System.Composition;
        using System.Web.Http;
        using Microsoft.Dynamics.Commerce.HardwareStation;
        using Microsoft.Dynamics.Retail.Diagnostics;
        /// <summary>
        /// Cash dispenser web API controller class.
        /// </summary>
        [Export("CASHDISPENSER", typeof(IHardwareStationController))]
        public class CashDispenserController : ApiController, IHardwareStationController
        {
            // Add your controller code here
        }
    }
}

```

## Scenario 2: Adding a new device type for an existing device

For this scenario, we will add support for a new device type for an existing device (an Audio Jack MSR implementation).

- The **Export** attribute string specifies the device that this controller is used for: [Export("MSR", typeof(IHardwareStationController))]
- Because there will be multiple controllers for MSRs, Hardware Station uses the configuration file to determine which implementation to use at run time. For more information, see the "Hardware Station extensibility configuration" section later in this article.

```

namespace Contoso
{
    namespace Commerce.HardwareStation.RamblerService
    {
        using System;
        using System.Composition;
        using System.Threading.Tasks;
        using System.Web.Http;
        using System.Web.Http.Controllers;
        using Microsoft.Dynamics.Commerce.HardwareStation;
        using Microsoft.Dynamics.Commerce.HardwareStation.DataEntity;
        using Microsoft.Dynamics.Commerce.HardwareStation.Models;
        using Microsoft.Dynamics.Retail.Diagnostics;
        /// <summary>
        /// MSR device web API controller class.
        /// </summary>
        [Export("MSR", typeof(IHardwareStationController))]
        [Authorize]
        public class AudioJackMSRController : ApiController, IHardwareStationController
        {
            // Add controller implementation here
        }
    }
}

```

## Hardware Station extensibility configuration

### Configuration for IIS-hosted Hardware Station

Before Hardware Station can consume your extension, the **composition** section in the Hardware Station Web.config file must be updated so that it includes an entry for your extension. The order of the composition targets in the configuration file determines precedence.

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="diagnosticsSection" type="Microsoft.Dynamics.Retail.Diagnostics.Core.Desktop.DiagnosticsConfigSection, Mi
    <section name="hardwareStation" type="Microsoft.Dynamics.Commerce.HardwareStation.Configuration.HardwareStationSection,
  </configSections>
  <hardwareStation>
    <secureStore cipherProvider="DPAPI" />
    <!-- Pairing key expiration period in days -->
    <pairingKeyExpirationPeriod="100" />
    <composition>
      <add source="assembly" value="Contoso.Commerce.HardwareStation.CashDispenserSample" />
      <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Security" />
      <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.Desktop" />
      <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.Network" />
      <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.Opos" />
      <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.SampleMx925Library" />
    </composition>
  </hardwareStation>

```

## Configuration for local IPC-based Hardware Station

Before local Hardware Station can consume your extension, the **composition** section in the Modern POS DLLHost.exe.config file (C:\Program Files (x86)\Microsoft Dynamics AX\70\Retail Modern POS\ClientBroker) must be updated so that it includes an entry for your extension. The order of the composition targets in the configuration file determines precedence.

```

<!-- Set to "true" to enable event tunneling. -->
<add key="IsNativeLoggingEnabled" value="true" />
<!-- Default download and upload interval for offline data sync are 1 minute -->
<add key="DefaultOfflineDownloadInterval" value="1" />
<add key="DefaultOfflineUploadInterval" value="1" />
<!-- AAD settings -->
<add key="AADLoginUrl" value="https://login.windows.net/common" />
<add key="AADClientId" value="d6b5a0bd-bf3f-4a8c-b370-619fb3d0e1cc" />
<add key="AADRetailServerResourceId" value="https://commerce.dynamics.com" />
<add key="LocatorServiceUrl" value="" />
<add key="CommerceAuthenticationAudience" value="Modern POS" />
</appSettings>
<commerceRuntime configSource="commerceRuntime.RealtimeServiceNotSupported.config" />
<commerceRuntime.HardwareStation configSource="commerceRuntime.HardwareStation.config" />
<hardwareStation>
  <secureStore cipherProvider="DPAPI" />
  <composition>
    <add source="assembly" value="Contoso.Commerce.HardwareStation.CashDispenserSample" />
    <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Security" />
    <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.Desktop" />
    <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.Opos" />
    <add source="assembly" value="Microsoft.Dynamics.Commerce.HardwareStation.Peripherals.SampleMx925Lib
  </composition>
</hardwareStation>
<diagnosticsSection>
  <application name="Modern POS" id="BF3114C0-8302-46EF-872A-8A479A04EDC5" />
  <emergencySink assembly="Microsoft.Dynamics.Retail.Diagnostics.Sinks" class="Microsoft.Dynamics.Retail
  <properties>
    <property name="LocatorServiceName" value="Microsoft.Dynamics.Modern POS" />

```

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Integrate the POS with a new hardware device

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how to integrate the point of sale (POS) with a new hardware device.

To call Hardware station from the POS, you must use a request and a response:

- **HardwareStationDeviceActionRequest** – The request that is sent from the POS to Hardware station.
- **HardwareStationDeviceActionResponse** – The response that the POS receives from Hardware station.

The class that you extend depends on the version of the Retail software development kit (SDK) that you're using.

- For Retail SDK version 10.0.11 or later, you extend the **IController** interface.
- For Retail SDK versions that are earlier than version 10.0.11, you extend the **HardwareStationController** and **IHardwareStationController** classes.

## HardwareStationDeviceActionRequest

The following code example shows the definition of **HardwareStationDeviceActionRequest**.

```
class HardwareStationDeviceActionRequest<TResponse extends HardwareStationDeviceActionResponse> extends Request<TResponse> {
    readonly device: string;
    readonly action: string;
    readonly actionData: any;
    constructor(device: string, action: string, actionData: any, correlationId?: string);
}
```

The following table describes the parameters.

PARAMETER	DATA TYPE	DESCRIPTION
device	String	The device name that is passed to the Hardware station request should match the <b>Export</b> attribute that is added in the Hardware station Device extension controller class.
action	String	The method that should be called in the Hardware station extension. The method name should be passed as a string value. The core POS Hardware station layer will then call the corresponding method from your Hardware station extension code. The method should exactly match the method name in your Hardware station extension. The Hardware station extension should be passed as a parameter.
actionData	any	A custom parameter for the extension to pass.

### Sample code

The following code example creates a **HardwareStationDeviceActionRequest** object.

```
let hardwareStationDeviceActionRequest:
HardwareStationDeviceActionRequest<HardwareStationDeviceActionResponse> =
    new HardwareStationDeviceActionRequest("Export attribute in Hardware station controller class",
        "extension method name in Hardware station", "Custom parameters/you can also pass custom object");
return this.extensionContextRuntime.executeAsync(hardwareStationDeviceActionRequest);
```

## HardwareStationDeviceActionResponse

The following code example shows the definition of **HardwareStationDeviceActionResponse**.

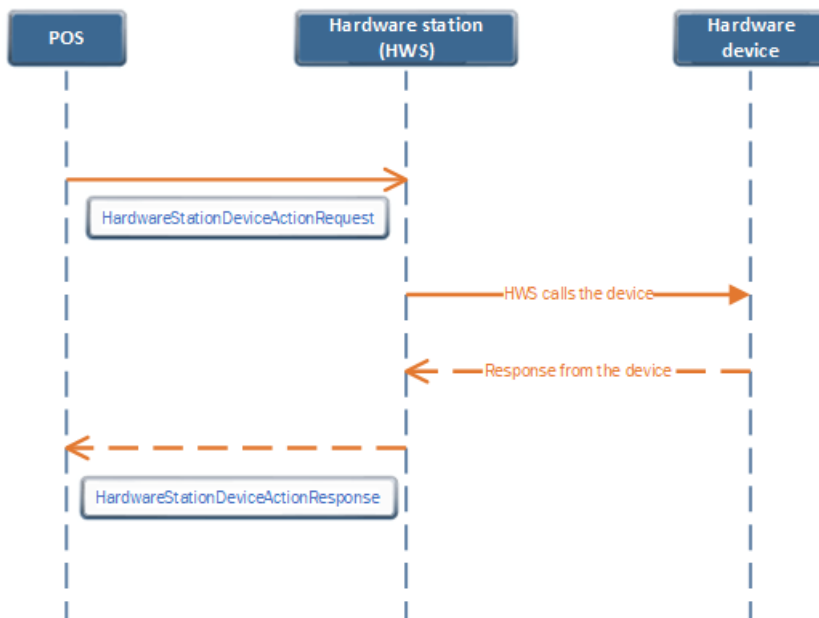
```
class HardwareStationDeviceActionResponse extends Response {
    readonly response: any;
    constructor(response: any);
}
```

The following table describes the parameters.

PARAMETER	DATA TYPE	DESCRIPTION
response	any	The response that is sent from the Hardware station extension code to the POS.

## End-to-end flow

The follow diagram shows the flow between the POS, Hardware station, and the hardware device.



## Hardware station extension

To call your new hardware device, you must implement the Hardware station code. You call your hardware device from that code.

To implement the Hardware station extension for Retail SDK version 10.0.11 or later, follow these steps.

1. Create a new C# class library project by using the Microsoft .NET Framework version 4.6.1. Alternatively, use one of the samples in the Retail SDK as a template. (You can find the samples at

...\RetailSDK\SampleExtensions\HardwareStation\.) We recommend that you use a sample as a template.

2. In the extension project, use the NuGet package manager to add the **Microsoft.Dynamics.Commerce.Hosting.Contracts** package. You can find the NuGet packages in the **RetailSDK\pkgs** folder.
3. Add a new controller class that extends the **IController** interface.
4. Add the **RoutePrefix** attribute to the controller class to expose the controller class to clients.

```
[RoutePrefix("ISVEXTENSIONDEVICE")]
```

5. To implement your custom logic to call the hardware device, in the controller class, add a method that has the **HttpPost** attribute. This method will be passed as the second parameter (action parameter) to the POS **HardwareStationDeviceActionRequest**. From the extension method, the extension can call other requests, such as printing and cash drawer requests. Just include the relevant NuGet packages from the Retail SDK.

```
[HttpPost]
public async Task<bool> IsReady(IEndpointContext context)
{
}
```

6. Build the project.

To implement the Hardware station extension for Retail SDK versions that are earlier than version 10.0.11, follow these steps.

1. Create a new C# class library project.
2. Add a new controller class that extends **HardwareStationController** and **IHardwareStationController**.
3. Add the **Export** attribute to the controller class. The **Export** attribute must be in all uppercase letters, and you must pass the value as a parameter from the POS extension. The device parameter that is passed from the POS **HardwareStationDeviceActionRequest** must match this value.
4. To implement your custom logic to call the hardware device, add your method in the controller class. This method will be passed as the second parameter (action parameter) to the POS **HardwareStationDeviceActionRequest**.
5. Build the project.

To deploy the Hardware station extension in Modern POS and test it by using the local Hardware station, follow these steps.

1. Copy the output library to the **C:\Program Files (x86)\Microsoft Dynamics 365\70\Retail Modern POS\ClientBroker\ext** folder.
2. Open the **HardwareStation.Extension.config** file.
3. In the **composition** section, add the extension library details.

```
<add source="assembly" value="your extension library name" />
```

4. Save the file.
5. Close Modern POS if it's running.
6. Open Task Manager, and end the **dllhost.exe** task.

7. Open Modern POS, and configure it to use the local Hardware station.

8. Validate your scenario.

To test by using Cloud POS, deploy the dynamics-link library (DLL) of the Hardware station extension to the shared Hardware station **ext** folder. Then update the **HardwareStation.Extension.config** file with the custom library in the shared Hardware station folder.

## Retail SDK samples

The Retail SDK includes some samples that you can use for reference.

- **POS:** \RetailSDK\POS\Extensions\FiscalRegisterSample
- **Hardware station:** \RetailSDK\SampleExtensions\HardwareStation\Extension.FiscalRegisterSample

## Sample code for Retail SDK version 10.0.11 or later

```

namespace Contoso
{
    namespace Commerce.HardwareStation.ISVExtensionDevice
    {
        using Microsoft.Dynamics.Commerce.Runtime.Hosting.Contracts;
        using System;
        using System.Threading.Tasks;

        /// <summary>
        /// Sample hardware station extension
        /// </summary>

        [RoutePrefix("ISVEXTENSIONDEVICE")]
        public class ISVExtensionDeviceController : IController
        {
            /// <summary>
            /// Sample.
            /// </summary>

            /// <param name="request">Custom request.</param>
            /// <returns>Result of Custom response.</returns>

            [HttpPost]
            public async Task<CustomResponse> Sample(CustomRequest request, IEndpointContext context)
            {
                CustomResponse response;
                try
                {
                    response = new CustomResponse();
                }
                catch (Exception ex)
                {
                    throw ex;
                }
                return await Task.FromResult(response);
            }
        }
        public class CustomResponse
        {
            public string sampleProp { get; set; }
            public CustomResponse()
            {
                this.sampleProp = "sampleValue";
            }
        }
    }
}

```

Sample code for Retail SDK versions before version 10.0.11

```

namespace Contoso
{
    namespace Commerce.HardwareStation.ISVExtensionDevice
    {
        using System;
        using System.Composition;
        using System.Web.Http;
        using Microsoft.Dynamics.Commerce.HardwareStation;

        /// <summary>
        /// Fiscal register peripheral web API controller class.
        /// </summary>

        [Export("ISVEXTENSIONDEVICE", typeof(IHardwareStationController))]
        [Authorize]
        public class ISVExtensionDeviceController : HardwareStationController, IHardwareStationController
        {
            /// <summary>
            /// Sample.
            /// </summary>
            /// <param name="request">Custom request.</param>
            /// <returns>Result of Custom response.</returns>

            [HttpPost]
            public CustomResponse Sample(CustomRequest request)
            {
                ThrowIf.Null(request, "request");
                try
                {
                    return null;
                }
                catch (Exception ex)
                {
                    throw ex;
                }
            }
        }
    }
}

```

## Sample POS code to call the Hardware station extension

From your POS extension, call the Hardware station by using the following pattern.

```

let hardwareStationDeviceActionRequest:
HardwareStationDeviceActionRequest<HardwareStationDeviceActionResponse> =
    new HardwareStationDeviceActionRequest("ISVEXTENSIONDEVICE",
        "Sample", "Custom parameters or custom object");
return this.extensionContextRuntime.executeAsync(hardwareStationDeviceActionRequest);

```

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Commerce Scale Unit customer and consumer APIs

2/18/2021 • 29 minutes to read • [Edit Online](#)

This topic provides an overview of the APIs that are available across various roles, and that can be used by various clients. The focus is on customer-facing application clients and eCommerce clients.

## Overview

- Commerce Scale Unit business data and operations are available to any connected device through the OData Web API, across both employee (point of sale) scenarios and customer (online store) scenarios.
- The embedded commerce runtime (CRT) enables a unified omni-channel platform.
- The application programming interfaces (APIs) are stateless and can process requests from many channels.
- The APIs have a linear scale-out model ("brick" scale-out).
- You use a composition pattern for plug-and-play customizations.
- The APIs are built on the .NET stack by using C#.

## Roles

Every request to Commerce Scale Unit (via commerce proxy) operates under these main roles:

- CommerceRole.Employee
- CommerceRole.Anonymous
- CommerceRole.Customer
- CommerceRole.Application

The Anonymous and Customer roles apply to eCommerce (customer/consumer) scenarios. The Anonymous role is used for requests that represent an eCommerce customer who hasn't signed in. The Customer role is used for requests that represent an eCommerce customer who has been authenticated and has signed in. A role filter is applied to every API that is exposed in Commerce Scale Unit. For eCommerce scenarios, you can use only APIs that have either CommerceRole.Anonymous or CommerceRole.Customer associated with them.

### NOTE

By default, Anonymous access is not enabled. To enable Anonymous access for your environment, contact [Support](#).

## Customer controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetOrderShipmentsHistory	string accountNumber, QueryResultSettings queryResultSettings	PageResult<OrderShipments>	Employee, Customer, Application	Gets order shipments history for the customers
CreateEntity	Customer	Customer	Employee, Anonymous, Application	Creates customer.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
UpdateEntity	string key, Customer update	Customer	Employee, Customer, Application	Updates customer.
GetOrderHistory	string accountNumber, QueryResultSettings queryResultSettings	PageResult<SalesOrder>	Employee, Customer, Application	Returns the collection of sales orders.
Search	CustomerSearchCriteria customerSearchCriteria, QueryResultSettings queryResultSettings	PageResult<GlobalCustomer>	Employee, Application	Searches for the customers
GetPurchaseHistory	string accountNumber, QueryResultSettings queryResultSettings	PageResult<PurchaseHistory>	Employee, Customer, Application	Gets purchase history for the customers
GetByAccountNumbers	IEnumerable<string> accountNumbers, int searchLocationValue, QueryResultSettings queryResultSettings	IEnumerable<Customer>	Employee, Customer, Application	Gets customers list from the list of customer account numbers
GetCustomerSearchFields	queryResultSettings	IEnumerable<CustomerSearchField>	Employee, Customer, Application	Gets the customer search fields for the store set in headquarters
SearchByFields	Customer entity	PageResult<GlobalCustomer>	Employee, Customer, Application	Searches for customers by specified fields
PostNonTransactionalActivityLoyaltyPoints	string loyaltyCardNumber, long channelId, long affiliationId, string activityTypeId	void	Employee, Customer, Application	Post non-transactional activity loyalty points

## Sales order controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetReceipts	string id, ReceiptRetrievalCriteria receiptRetrievalCriteria, QueryResultSettings queryResultSettings	PageResult<Receipt>	Employee	Gets a set of receipts based on the formTypes for Printing.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetGiftReceipts	string id, IEnumerable<decimal> salesLineNumbers, ReceiptRetrievalCriteria receiptRetrievalCriteria, QueryResultSettings queryResultSettings	PageResult<Receipt>	Employee	Gets the gift receipts
GetByReceiptId	string receiptId, string orderStoreNumber, string orderTerminalId, QueryResultSettings queryResultSettings	PageResult<SalesOrder>	Employee	Gets sales orders by the receipt identifier
SearchSalesTransactionsBy- ReceiptId	string receiptId, QueryResultSettings queryResultSettings	PageResult<SalesOrder>	Employee	Search sales transaction by the receipt identifier
Search	SalesOrderSearchCriteria salesOrderSearchCriteria, QueryResultSettings queryResultSettings	PageResult<SalesOrder>	Employee, Customer	Searches for any orders matching the given search criteria
SearchOrders	OrderSearchCriteria orderSearchCriteria, QueryResultSettings queryResultSettings	PageResult<SalesOrder>	Employee, Customer	Searches for orders matching the given search criteria.
GetInvoicesBySalesId	string salesId, QueryResultSettings queryResultSettings	PageResult<SalesInvoice>	Employee	Gets the sales invoice associated with the passed sales identifier
GetOrderInvoices	string customerAccount, QueryResultSettings queryResultSettings	PageResult<OrderInvoice>	Employee	Gets the open order invoices associated with the customer associated with the given customer identifier
GetInvoices	InvoiceSearchCriteria invoiceSearchCriteria, QueryResultSettings queryResultSettings	PageResult<OrderInvoice>	Employee	Gets the open invoices associated with the search criteria
GetInvoicedSalesLinesBy- SalesIds	IEnumerable<string> salesIds, QueryResultSettings queryResultSettings	PageResult<SalesLine>	Employee	Gets the list of invoiced sales lines by sales order identifiers

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
CreatePickingList [Obsolete("Use CreatePickingListForItems instead.")]	string salesId	void	Employee	Creates a picking list for a sales order
CreatePickingListForItems	string salesId, IEnumerable<PickAndPackSalesLineParameter> pickAndPackSalesLineParameters	string	Employee	Creates a picking list for selected lines on sales order.
GetPickingLists	string salesId, QueryResultSettings queryResultSettings	PageResult<PickingList>	Employee	Gets the picking lists for an order from Headquarters.
CreatePackingSlip		void	Employee	Creates a packing slip
GetSalesOrderDetails By- TransactionId	string transactionId, int searchLocationValue	SalesOrder	Employee, Customer	Gets the sales order details by transaction id.
GetSalesOrderDetails By- SalesId	string salesId	SalesOrder	Employee, Customer	Gets the sales order details by sales id.
GetSalesOrderDetails By- QuotationId	string quotationId	SalesOrder	Employee, Customer	Gets the sales order details by Quotation id.

## Cart controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
Checkout	string id, string receiptEmail, TokenizedPaymentCard tokenizedPaymentCard, string receiptNumberSequence, IEnumerable<CartTenderLine> cartTenderLines, long? cartVersion	SalesOrder	Employee, Customer, Anonymous, Application	Checkout the cart.
AddCartLines	string id, System.Collections.Generic.IEnumerable<CartLine> cartLines, long? cartVersion	Cart	Employee, Customer, Anonymous, Application	Adds the cart lines into Cart.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
VoidCartLines	string id, System.Collections.Generic.IEnumerable<CartLine> cartLines	Cart	Employee	voids the cart lines in the cart.
UpdateCartLines	string id, System.Collections.Generic.IEnumerable<CartLine> cartLines	Cart	Employee, Customer, Anonymous, Application	Updates the cart lines in the cart.
RefillGiftCard	string id, string giftCardId, decimal amount, string currencyCode, string lineDescription)	Cart	Employee	Adds balance to gift card
IssueGiftCard	string id, string giftCardId, decimal amount, string currencyCode, string lineDescription, string tenderTypeId	Cart	Employee	Issues gift card.
CashOutGiftCard	string id, string giftCardId, decimal amount, string currencyCode, string lineDescription	Cart	Employee	Cash out a gift card.
AddTenderLine	string id, CartTenderLine cartTenderLine, long? cartVersion	Cart	Employee	Adds the cart tender line.
AddPreprocessed-TenderLine	string id, TenderLine preprocessedTenderLine, long? cartVersion	Cart	Employee	Adds the pre-processed tender line.
ValidateTender-LineForAdd	string id, TenderLine tenderLine	void	Employee	Validates the tender line.
UpdateTenderLine-Signature	string id, string tenderLineId, string signatureData	Cart	Employee	Updates the cart tender line signature.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
VoidTenderLine	string id, string tenderLineId, System.Collections.Generic.IEnumerable<ReasonCodeLine> reasonCodeLines, bool? isPreprocessed = false, bool? forceVoid = false	Cart	Employee.	voids the cart tender line.
SuspendWithJournal	string id, string journalCartId, string receiptNumberSequence	Cart	Employee	Suspends a cart and makes journal entry.
Resume	string id	Cart	Employee	Resumes a suspended cart.
ResumeFromReceiptId	string receiptId	Cart	Employee	Resumes a suspended cart based on receipt id.
RecallOrder	string transactionId, string salesId	Cart	Employee	Recalls a customer order.
AddInvoicedSalesLinesToCart	string transactionId, IEnumerable<long> invoicedLineIds	Cart	Employee.	Add invoiced sales lines to cart.
RecallQuote	string transactionId, string quoteId	Cart	Employee	Recalls a quote.
RecallSalesInvoice	string transactionId, string invoiceId	Cart	Employee	Gets the cart that represents the invoice associated with the passed invoice identifier.
AddOrderInvoice	string id, string invoiceId, string lineDescription	Cart	Employee	Adds the invoice associated with the passed invoice identifier to the cart.
AddInvoices	string key, IEnumerable<string> invoiceIds	Cart	Employee	Add invoices to cart.
RecalculateOrder	string id	Cart	Employee	Recalculates a customer order.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
UpdateCommission-SalesGroup	string transactionId, string cartLineId, string commissionSalesGroup, bool isUserInitiated	Cart	Employee	Updates a commission sales group on a line or transaction.
CartDeliveryPreferences	string id	CartDeliveryPreferences	Customer, Anonymous, Application	Gets the applicable delivery preference types based on the items in the cart.
GetLineDeliveryOptions	string id, IEnumerable<LineShippingAddresses> lineShippingAddresses, QueryResultSettings queryResultSettings	PageResult<SalesLineDeliveryOption>	Employee, Customer, Anonymous, Application	Gets the delivery line option of the cart.
GetLineDeliveryOptionsBy-ChannelId	string id, IEnumerable<LineShippingAddresses> lineShippingAddresses, long channelId, QueryResultSettings queryResultSettings	PageResult<SalesLineDeliveryOption>	Employee, Customer, Anonymous, Application	Gets the delivery line option of the cart by the channel identifier.
GetPaymentsHistory	string id, QueryResultSettings queryResultSettings	PageResult<TenderLine>	Employee	Gets the payments history given the cart identifier.
GetDeliveryOptions	string id, Address shippingAddress, QueryResultSettings queryResultSettings	PageResult<DeliveryOption>	Employee, Customer, Anonymous, Application	Gets the delivery options for the cart.
UpdateLineDelivery-Specifications	string id, System.Collections.Generic.IEnumerable<LineDeliverySpecification> lineDeliverySpecifications	Cart	Customer, Anonymous, Application	Updates the delivery specifications per cart line.
AddCharge	string cartId, int moduleTypeValue, string chargeCode, decimal calculatedAmount	Cart	Employee, Application	Add a charge to the cart.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
OverrideCharge	string cartId, string chargeLineId, decimal amount, IEnumerable<ReasonCodeLine> reasonCodeLines	Cart	Employee, Application	Override the amount of a charge in the cart.
AddCartLineCharge	string cartId, string cartLineId, int moduleTypeValue, string chargeCode, decimal calculatedAmount	Cart	Employee, Application	Add a charge to the cart line.
OverrideCartLineCharge	string cartId, string cartLineId, string chargeLineId, decimal amount, IEnumerable<ReasonCodeLine> reasonCodeLines	Cart	Employee, Application	Override the amount of a cart line charge.
UpdateDelivery-Specification	string id, DeliverySpecification deliverySpecification	Cart	Customer, Anonymous, Application	Updates the delivery specification for cart header.
OverrideCartLinePrice	string id, string cartLineId, decimal price	Cart	Employee	Processes the barcode workflow by sending the cart identifier and barcode scanned information.
GetPromotions	string id	CartPromotions	Customer, Anonymous, Application	Gets the promotions for cart.
AddDiscountCode	string id, string discountCode	Cart	Employee, Customer, Anonymous, Application	Add discount code to cart.
RemoveDiscountCodes	string id, IEnumerable<string> discountCodes	Cart	Customer, Anonymous, Application	Remove discount code from cart.
RemoveCartLines	string id, System.Collections.Generic.IEnumerable<string> cartLineIds	Cart	Customer, Anonymous, Application	Deletes the cart lines.
Search	CartSearchCriteria cartSearchCriteria, QueryResultSettings queryResultSettings	PageResult<Cart>	Customer	Gets the carts by customer.



API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCardPayment-AcceptPoint	string id, CardPaymentAcceptSettings cardPaymentAcceptSettings	CardPaymentAcceptPoint	Employee, Customer, Anonymous, Application	Gets the accepting point of card payment, e.g. a web page.
RetrieveCardPayment - AcceptResult	string resultAccessCode	CardPayment-AcceptResult	Employee, Customer, Anonymous, Application	Retrieves the accepting result of card payment, e.g. payment authorization, card token.
AddCoupons	string id, IEnumerable<string> couponCodes, bool? isLegacyDiscountCode = false	Cart	Employee, Customer, Anonymous, Application	Add coupons to the cart.
RemoveCoupons	string id, IEnumerable<string> couponCodes	Cart	Employee, Customer, Anonymous, Application	Remove coupon codes from the cart.
GetChargeCodes	QueryResultSettings settings	PageResult<ChargeCode>	Employee, Application	Gets all the charge codes.
GetMaxLoyaltyPointsTo- RedeemFor- TransactionBalance	string cartId, string loyaltyCardNumber, string redeemCurrency	LoyaltyPoint-RedemptionEstimate	Employee, Customer	Get a LoyaltyPoint-RedemptionEstimate object, which contains the max currency amount a loyalty card can apply to a transaction's balance, and the reward point quantities used to generate that max currency amount.
GetDeclinedOrVoided - CardReceipts	string cartId, TenderLine preprocessedTenderLine, ReceiptRetrievalCriteria criteria, QueryResultSettings queryResultSettings	PageResult<Receipt>	Employee	Gets a collection of gift receipts for a declined or terminated card tender line.
ResetAllCharges	string id	Cart	Employee, Application	Recalculates charges (including deleting all manually added and overridden charges) for the cart.

## Address controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetAddressPurposes	QueryResultSettings queryResultSettings	PageResult<AddressPurpose>	Employee, Customer, Anonymous, Application	Gets the address purposes

## Barcode controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetBarcodeById	string barcodeId	Barcode	Employee	Gets barcode by identifier.

## Cash declaration controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCashDeclarations	QueryResultSettings queryResultSettings	PageResult<CashDeclaration>	Employee	Gets barcode by identifier.

## Cities controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCities	string countryRegionId, string stateProvinceId, string countyId, QueryResultSettings queryResultSettings	PageResult<CityInfo>	Employee	Get all the cities filtered by Country/Region, State Province and County.

## Counties controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCounties	string countryRegionId, string stateProvinceId, QueryResultSettings queryResultSettings	PageResult<CountyInfo>	Employee	Get all the counties filtered by country/region and state province.

## Country region controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCountryRegionsFor- Shipping	QueryResultSettings queryResultSettings PageResult<CountryRegionInfo>		Employee, Customer, Anonymous, Application	Gets the translated countries/regions with delivery modes configured for the current channel.
GetCountryRegionsBy- LanguageId	string languageId, QueryResultSettings queryResultSettings	PageResult<CountryRegionInfo>	Employee, Customer, Anonymous, Application	Get all the countries/regions filter by Language Id.
GetCountryRegions	QueryResultSettings queryResultSettings	PageResult<CountryRegionInfo>	Employee	Get all the countries/regions.

## Credit memo controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCreditMemoById	string creditMemoid	CreditMemo	Employee	Get credit memo by identifier.

## Delivery options controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetDeliveryOptions	string id, Address shippingAddress, QueryResultSettings queryResultSettings	PageResult<DeliveryOption>	Employee, Customer, Anonymous, Application	Get the delivery options for the channel.

## Customer group controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCustomerGroups	QueryResultSettings queryResultSettings	PageResult<CustomerGroup>	Employee, Customer, Anonymous, Application	Gets collection of customer group.

## Currency controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCurrenciesAmount	string currencyCode, decimal amount, QueryResultSettings queryResultSettings	PageResult<CurrencyAmount>	Employee	Gets the currencies amount.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
CalculateTotalCurrencyAmount	System.Collections.Generic.IEnumerable<CurrencyRequest> currenciesAmount	CurrencyAmount	Employee	Calculates the total currency amount.

## Customer balance controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCustomerBalance	string accountNumber, string invoiceAccountNumber	CustomerBalances	Employee	Gets the customer balance.

## Device configuration controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetDeviceConfiguration		DeviceConfiguration	Employee	Gets a single device configuration.

## District controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetDistricts	string countryRegionId, string stateProvinceId, string countyId, string cityName, QueryResultSettings queryResultSettings	PageResult<DistrictInfo>	Employee	Get all the district info filtered by country/region, state province, county, city.

## State province controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetStateProvinces	string countryRegionId, QueryResultSettings queryResultSettings	PageResult<StateProvinceInfo>	Employee, Customer, Anonymous, Application	Get all the states or provinces filtered by country/region.

## Discount controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetDiscountCodes	QueryResultSettings queryResultSettings	PageResult<Discount Code>	Employee	Gets collection of discount codes.
GetDiscountCodesBy OfferId	string offerId, QueryResultSettings queryResultSettings	PageResult<Discount Code>	Employee	Gets collection of discount codes filtered by offer identifier.
GetDiscountCodesBy Keyword	string keyword, DateTimeOffset activeDate, QueryResultSettings queryResultSettings	PageResult<Discount Code>	Employee	Searches the discount codes.
GetDiscountCode	string discountCode	DiscountCode	Employee	Gets the discount code.

## Zipcodes controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetZipCodes	string countryRegionId, string stateProvinceId, string countyId, string cityName, string district, QueryResultSettings queryResultSettings	PageResult<ZipCodeInfo>	Employee	Get all the zip codes filtered by country/region, state province, county, city and district.
GetAddressFromZipCode	string countryRegionId, string zipPostalCode, QueryResultSettings queryResultSettings	PageResult<ZipCodeInfo>	Employee, Customer, Anonymous, Application	Get details of addresses associated with zip code filtered by country/region.

## Suspended cart controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetAllSuspendedCarts	QueryResultSettings queryResultSettings	PageResult<SuspendedCart>	Employee	Gets all suspended carts.

## Tender types controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
-----	-----------	--------------	--------------------------	-------------

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetTenderTypes	QueryResultSettings queryResultSettings	PageResult<TenderType>	Employee, Customer, Anonymous, Application	Gets tender types.
RoundAmountByTenderType	decimal amount, string tenderType	decimal	Employee	Round amount by tender type.

## Publishing controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
SetOnlineChannelPublishStatus	int publishingStatus, string publishingStatusMessage	void	Application	Updates Online Channel publishing status.

## Language controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetLanguages	QueryResultSettings queryResultSettings	PageResult<SupportedLanguage>	Employee	Gets collection of supported languages.

## Localized string controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetLocalizedStrings	string languageId, int? textId, QueryResultSettings queryResultSettings	PageResult<LocalizedString>	Employee	Gets all localized strings filtered by language identifier, text identifier.

## Notification controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetNotifications	IEnumerable<int> subscribedOperations, QueryResultSettings queryResultSettings	ICollection<NotificationItem>	Employee	Gets the notifications.
AcknowledgeNotifications	DateTimeOffset lastPullDateTime	void	Employee	Acknowledge notifications.

## Number sequence controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetLatestNumberSequence	QueryResultSettings queryResultSettings	PageResult<Localized String>	Employee	Gets the next number sequence for the current terminal.

## ReasonCodes controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetReasonCodes	QueryResultSettings queryResultSettings	PageResult<ReasonCode>	Employee	Gets the reason codes.
GetReturnOrderReasonCodes	QueryResultSettings queryResultSettings	PageResult<ReasonCode>	Employee	Gets return order reason codes.
GetReasonCodesById	string reasonCodeGroupId, QueryResultSettings queryResultSettings	PageResult<ReasonCode>	Employee	Gets the reason codes by group or single identifier.

## Receipt controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetReceiptMasks	int? receiptTransactionType, QueryResultSettings queryResultSettings	PageResult<ReceiptMask>	Employee	Gets the receipts masks.
ValidatePrintReceiptCopyAllowed	SalesOrder salesOrderToPrint	void	Employee	Performs validation, whether receipt copy printing operation is allowed or not.

## Report datasets controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
SearchReportDataSet	int? receiptTransactionType, QueryResultSettings queryResultSettings	ReportDataSet	Employee	Searches all the report dataset filtered by Report Identifier, Parameters and Locale.
GetReportDataSetById	SalesOrder salesOrderToPrint	ReportDataSet	Employee	Gets report data set by id.
GetSrsReportDataSet		ReportDataSet	Employee	Gets SSRS report data set.

## Search controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetSearchSuggestions	SearchSuggestionCriteria suggestionCriteria, QueryResultSettings settings	PageResult<SearchSuggestion>	Employee, Customer, Anonymous, Application	Gets Search suggestions.
GetSearchConfiguration		SearchConfiguration	Employee, Customer, Anonymous, Application	Gets Channel Search Configuration from Azure Search

## Tax controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetTaxOverrides	SearchSuggestionCriteria suggestionCriteria, QueryResultSettings settings	PageResult<TaxOverride>	Employee	Searches for any tax overrides matching the given search criteria.
GetSalesTaxGroups		PageResult<SalesTaxGroup>	Employee	Gets the sales tax groups.

## Tender drop and declare operation controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
CreateDropAndDeclareTransaction	DropAndDeclareTransaction dropAndDeclareTransaction	DropAndDeclareTransaction	Employee	Performs saving tender drop and declare store operations.

## Unit of measure controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetUnitsOfMeasure	QueryResultSettings queryResultSettings	PageResult<UnitOfMeasure>	Employee	Get all the units of measure supported by the store.

## Income expense accounts controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
-----	-----------	--------------	--------------------------	-------------



API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetIncomeExpenseAccounts	int incomeExpenseAccountType, QueryResultSettings queryResultSettings	PageResult<IncomeExpenseAccount>	Employee	Gets the income or expense accounts.

## Products controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
Search	ProductSearchCriteria productSearchCriteria, QueryResultSettings queryResultSettings	IEnumerable<Product>	Employee, Customer, Anonymous, Application	Searches for the product using OData query.
GetById	long recordId, long channelId	SimpleProduct	Employee, Customer, Anonymous, Application	Gets a SimpleProduct by its record identifier.
Get		PageResult<Product>	Employee, Customer, Storefront	Searches for the product.
GetByIds	long channelId, IEnumerable<long> productIds, QueryResultSettings queryResultSettings	PageResult<SimpleProduct>	Employee, Customer, Anonymous, Application	Gets a collection of products based on channel identifier and record identifier.
GetRecommendedProducts	IEnumerable<long> productIds, string customerAccountNumber, QueryResultSettings queryResultSettings	PageResult<ProductSearchResult>	Employee, Customer, Anonymous, Application	Retrieves a collection of SimpleProduct recommendations given a collection of product identifiers.
Compare	long channelId, long catalogId, IEnumerable<long> productIds, QueryResultSettings queryResultSettings	PageResult<ProductComparisonLine>	Employee, Customer, Anonymous, Application	Compares products.
SearchByCategory	long channelId, long catalogId, long categoryId, QueryResultSettings queryResultSettings	PageResult<ProductSearchResult>	Employee, Customer, Anonymous, Application	Searches for products that belong to a category directly or via its child categories.
SearchByText	long channelId, long catalogId, string searchText, QueryResultSettings queryResultSettings	PageResult<ProductSearchResult>	Employee, Customer, Anonymous, Application	Searches for products that are associated to the given search text.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetSearchSuggestions	long channelId, long catalogId, string searchText, string hitPrefix, string hitSuffix, QueryResultSettings queryResultSettings	PageResult-<SearchSuggestion>	Employee, Customer, Anonymous, Application	Gets recommended search phrases based on a (partial) search text.
GetRefinersByCategory	long catalogId, long categoryId, QueryResultSettings queryResultSettings	PageResult-<ProductRefiner>	Employee, Customer, Anonymous, Application	Gets the product refiner(s) available for the given category product(s).
GetRefinersByText	long catalogId, string searchText, QueryResultSettings queryResultSettings	PageResult-<ProductRefiner>	Employee, Customer, Anonymous, Application	Gets the product refiner(s) available for product(s) resulting from searching the given text.
GetProductSearchRefiners	ProductSearchCriteria searchCriteria, QueryResultSettings queryResultSettings	PageResult-<ProductRefiner>	Employee, Customer, Anonymous, Application	Gets the product refiner(s) available for product(s) resulting from the combination of refiners and search text/ category ids being used.
GetRefinerValuesByCategory	long catalogId, long categoryId, long refinerId, int refinerSourceValue, QueryResultSettings queryResultSettings	PageResult-<ProductRefinerValue>	Employee, Customer, Anonymous, Application	Gets the product refiner value(s) available for the given category product(s).
GetRefinerValuesByText	long catalogId, string searchText, long refinerId, int refinerSourceValue, QueryResultSettings queryResultSettings	PageResult-<ProductRefinerValue>	Employee, Customer, Anonymous, Application	Gets the product refiner value(s) available for product(s) resulting from searching the given text.
RefineSearchByCategory	long channelId, long catalogId, long categoryId, IEnumerable-<ProductRefinerValue> refinementCriteria, QueryResultSettings queryResultSettings	PageResult-<ProductSearchResult>	Employee, Customer, Anonymous, Application	Refines searches performed on products that belong to a category directly or via its child categories.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
RefineSearchByText	long channelId, long catalogId, string searchText, IEnumerable<ProductRefinerValue> refinementCriteria, QueryResultSettings queryResultSettings	PageResult<ProductSearchResult>	Employee, Customer, Anonymous, Application	Refines searches performed on products that are associated to the given search text.
GetDimensionValues	long recordId, long channelId, int dimension, IEnumerable<ProductDimension> matchingDimensionValues, QueryResultSettings queryResultSettings	PageResult<ProductDimensionValue>	Employee, Customer, Anonymous, Application	Gets the dimension values for a product based on the specified requirements.
GetVariantsBy-DimensionValues	long recordId, long channelId, IEnumerable<ProductDimension> matchingDimensionValues, QueryResultSettings queryResultSettings	PageResult<SimpleProduct>	Employee, Customer, Anonymous, Application	Gets the variations of a product based on the specified requirements.
GetVariantsBy-ComponentsInSlots	long recordId, long channelId, IEnumerable<ComponentInSlotRelation> matchingSlotTo-ComponentRelation, QueryResultSettings queryResultSettings	PageResult<SimpleProduct>	Employee, Customer, Anonymous, Application	Gets the variations of a product based on the component in slot combination specified.
GetDefaultComponents	long recordId, long channelId, QueryResultSettings queryResultSettings	PageResult<ProductComponent>	Employee, Customer, Anonymous, Application	Gets the default individual parts that constitute the specified product.
GetComponentByProduct-SlotRelation	long channelId, ComponentInSlotRelation componentRelation	ProductComponent	Employee, Customer, Anonymous, Application	Gets a specific product component based on the provided ComponentInSlotRelation.
GetSlotComponents	long recordId, long channelId, long slotId, QueryResultSettings queryResultSettings	PageResult<ProductComponent>	Employee, Customer, Anonymous, Application	Gets the default individual parts that can fit into a slot of a product to complete its constitution.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetFiltered-SlotComponents	long recordId, long channelId, long slotId, IEnumerable<ComponentInSlotRelation> selectedComponents, QueryResultSettings queryResultSettings	PageResult-<ProductComponent >	Employee, Customer, Anonymous, Application	Gets the Product components that may be selected given a set of previously selected components.
GetAttributeValues	long recordId, long channelId, long catalogId, QueryResultSettings queryResultSettings	PageResult<Attribute Value>	Employee, Customer, Anonymous, Application	Gets the attribute values of the specified product.
GetRelationTypes	long recordId, long channelId, long catalogId, QueryResultSettings queryResultSettings	PageResult-<ProductRelationType>	Employee, Customer, Anonymous, Application	Gets the types of relationships the specified product has with other products.
GetRelatedProducts	long recordId, long channelId, long catalogId, long relationTypeId, QueryResultSettings queryResultSettings	PageResult-<ProductSearchResult>	Employee, Customer, Anonymous, Application	Searches for products that are associated to the specified product by the specified relationship.
GetRefiners	ProductSearchCriteria productSearchCriteria, QueryResultSettings queryResultSettings	PageResult<ProductRefiner>	Employee, Customer, Anonymous, Application	Searches for product refiners using OData query.
Changes	ChangedProductsSearchCriteria productSearchCriteria, QueryResultSettings queryResultSettings	IEnumerable<Product>	Employee, Storefront	Searches and retrieves changed products given the specified query criteria.
ReadChangedProducts	ChangedProductsSearchCriteria productSearchCriteria, QueryResultSettings queryResultSettings	PageResult<Product >	Application	Searches and retrieves changed products given the specified query criteria.
GetDeletedListings	long catalogId, long skip, long top	DeletedListingsResult	Application	Searches and retrieves changed products given the specified query criteria.
GetDeletedCatalogs	QueryResultSettings queryResultSettings	IEnumerable<long>	Application	Gets deleted catalogs.
GetDeletedLanguages	QueryResultSettings queryResultSettings	IEnumerable<string>	Application	Gets deleted languages.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
DeleteListingsBy-Catalogs	IEnumerable<long> catalogIds	void	Application	Deletes listings by catalogs.
DeleteListingsBy-Languages	IEnumerable<string> languages	void	Application	Deletes listings by languages.
BeginRead-ChangedProducts	ChangedProductsSearchCriteria changedProductSearchCriteria	ReadChanged-ProductsSession	Application	Begins session to read changed products.
EndReadChangedProducts	ReadChangedProductsSession session	void	Application	Ends session to read changed products.
UpdateListing-PublishingStatus	IEnumerable<ListingPublishStatus> publishingStatuses	void	Application	Searches and retrieves changed products given the specified query criteria.
GetProductAvailabilities	IEnumerable<long> itemIds, long channelId, QueryResultSettings queryResultSettings	PageResult<ProductAvailableQuantity>	Employee, Customer, Anonymous, Application	Get available inventory for given list of items for given channel and customer.
GetPrices	string itemId, string inventoryDimensionId, string barcode, string customerAccountNumber, string unitOfMeasureSymbol, decimal quantity, QueryResultSettings queryResultSettings	PageResult<ProductPrice>	Employee	Gets the price of an item in context of the current customer.
GetPrice	long recordId, string customerAccountNumber, string unitOfMeasureSymbol	ProductPrice	Employee, Customer, Anonymous, Application	Gets the price of a product in context of the current customer.
CalculateProductPrice	long recordId, string customerAccountNumber, string unitOfMeasureSymbol, string loyaltyCardId, IEnumerable<AffiliationLoyaltyTier> affiliationLoyaltyTiers	ProductPrice	Employee, Customer, Anonymous, Application	Gets the price.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetActivePrices	ProjectionDomain projectDomain, IEnumerable<long> productIds, DateTimeOffset activeDate, string customerId, IEnumerable<Affiliati onLoyaltyTier> affiliationLoyaltyTiers, bool? includeSimpleDiscoun tsIn- ContextualPrice, QueryResultSettings queryResultSettings	PageResult<ProductP rice>	Employee, Customer, Anonymous, Application	Gets the price.
GetMediaLocations	long recordId, long channelId, long catalogId, QueryResultSettings queryResultSettings	PageResult<MediaLo cation>	Employee, Customer, Anonymous, Application	Gets the media locations for the specified product.
GetMediaBlobs	long recordId, long channelId, long catalogId, QueryResultSettings queryResultSettings	PageResult<MediaBlo b>	Employee, Customer, Anonymous, Application	Gets the media blobs for the specified product.
GetUnitsOfMeasure	long recordId, QueryResultSettings queryResultSettings	PageResult<UnitOfM easure>	Employee, Customer, Anonymous, Application	Gets the unit(s) of measure for the specified product.
GetChannel- ProductAttributes	QueryResultSettings queryResultSettings	PageResult<Attribute Product>	Employee, Customer, Anonymous, Application	Gets the channel product attributes.
GetProductRatings	IEnumerable<long> productIds, QueryResultSettings settings	PageResult<ProductR ating>	Employee, Customer, Anonymous, Application	Gets a collection of product ratings based on product identifiers.
GetEstimatedAvailabil ity	InventoryAvailabilityS earchCriteria searchCriteria	ProductWarehouseIn ventoryInformation	Employee, Customer, Anonymous, Application	Get estimated product availability based on search criteria.
GetEstimatedProduct WarehouseAvailabilit y	InventoryAvailabilityS earchCriteria searchCriteria	IEnumerable<Produc tWarehouse>	Employee, Customer, Anonymous, Application	Get estimated product availability for specific product warehouse pairs.

## Sales orders fulfillment controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
ShipFulfillmentLines	IEnumerable<ShipFulfillmentLine> fulfillmentLines	void	Employee	Ship the fulfillment lines. Invoices in AX.
AcceptFulfillmentLines	IEnumerable<FulfillmentLineParameter> fulfillmentLines	void	Employee	Updates the status of the fulfillment lines to Accepted.
PickFulfillmentLines	IEnumerable<FulfillmentLineParameter> fulfillmentLines, IEnumerable<FulfillmentLineParameter> fulfillmentLines	void	Employee	Updates the status of the fulfillment lines to Picking.
MarkAsPicked	IEnumerable<FulfillmentLineParameter> fulfillmentLines	void	Employee	Updates the status of the fulfillment lines to Picked.
PackFulfillmentLines	IEnumerable<FulfillmentLineParameter> fulfillmentLines	void	Employee	Updates the status of the fulfillment lines to Packed or Partially Packed.
MarkFulfillmentLinesAsPacked	IEnumerable<FulfillmentLineParameter> fulfillmentLines	String	Employee	Updates the status of the fulfillment lines to Packed or Partially Packed.
GetFulfillmentLines	FulfillmentLineSearchCriteria criteria, QueryResultSettings settings	IEnumerable<FulfillmentLine>	Employee	Gets the fulfillment lines.
GetFulfillmentPackingSlips		IEnumerable<Receipt>	Employee	Gets the packing slips.
GetFulfillmentPackingSlipsById	string salesId, string packingSlipId, string hardwareProfileId	IEnumerable<Receipt>	Employee	Gets the packing slips by packing slip Id and sales Id.
GetFulfillmentPickingLists	IEnumerable<FulfillmentLineParameter> pickingListFulfillmentLines, string hardwareProfileId, QueryResultSettings queryResultSettings	IEnumerable<Receipt>	Employee	Gets the picking lists.
RejectFulfillmentLines	IEnumerable<RejectFulfillmentLine> fulfillmentLines	void	Employee	Updates the status of the fulfillment lines to Rejected.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetPackingSlipsData	string salesId	IEnumerable<Packing SlipData>	Employee	Gets the list of packing slip data given a sales identifier.

## Hardware profiles controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetHardwareProfileById	string hardwareProfileId	HardwareProfile	Employee	Gets hardware profile by id.
GetHardwareStationProfiles	QueryResultSettings queryResultSettings	PageResult<HardwareStationProfile>	Employee	Gets collection of hardware station profiles.

## Income expense account controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetIncomeExpenseAccounts	int incomeExpenseAccountType, QueryResultSettings queryResultSettings	PageResult<IncomeExpenseAccount>	Employee	Gets the income or expense accounts.

## Kits controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
DisassembleKitTransactions	KitTransaction	KitTransaction	Employee	Performs kit (disassembly) transaction operations.

## Gift card controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetGiftCardInquiry	string giftCardId	GiftCard	Employee, Customer, Anonymous, Application	Get gift card with additional information by identifier.

## Image controller



API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetImageBlob	long imageId	MediaBlob	Employee, Customer, Anonymous, Application	Gets image blob by image identifier.

## Store safe controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetStoreSafes	QueryResultSettings settings	PageResult<StoreSafe>	Employee,	Get store safe list.

## Warehouse controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetWarehouseById	string inventLocation	PageResult<StoreSafe>	Employee, Customer	Gets a Warehouse by its record identifier.
SearchWarehouses	string searchText, QueryResultSettings queryResultSettings	PageResult<Warehouse>	Application, Employee, Customer	Retrieves a list of warehouses that matches the given search text.
GetLocations	string inventLocation, QueryResultSettings queryResultSettings	PageResult<WarehouseLocation>	Application, Employee, Customer	Gets the warehouse locations of the specified Warehouse.
SearchLocations	string inventLocation, string searchText, QueryResultSettings queryResultSettings	PageResult<WarehouseLocation>	Application, Employee, Customer	Retrieves a list of warehouse locations for the given warehouse that matches the given search text.

## Recommendation controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
Get		PageResult<Recommendation>	Application, Employee, Customer, Anonymous	Gets the list of recommendations.
GetElements	string listId, RecommendationCriteria criteria, QueryResultSettings queryResultSettings	PagedResult<RecommendedElement>	Application, Employee, Customer, Anonymous	Gets the collection of recommended elements given the (optional) contextual information as criteria.

# Transfer order controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
Get	QueryResultSettings queryResultSettings	PageResult<Transfer Order>	Employee	Gets open transfer orders for the store.
Commit	string orderId	Void	Employee	Commits a transfer order.
GetTransferOrderJournals	string orderId, QueryResultSettings queryResultSettings	PageResult<Transfer OrderJournal>	Employee	Gets the transfer order journals of the specified transfer order.
GetTransferOrderLines	string orderId, QueryResultSettings queryResultSettings	PageResult<Transfer OrderLine>	Employee	Gets the transfer order lines.
CreateTransferOrderLines	string orderId, IEnumerable<TransferOrderLine> transferOrderLines, QueryResultSettings queryResultSettings	PageResult<Transfer OrderLine>	Employee	Creates the transfer order lines.
UpdateTransferOrderLines	string orderId, IEnumerable<TransferOrderLine> transferOrderLines, QueryResultSettings queryResultSettings	PageResult<Transfer OrderLine>	Employee	Updates the transfer order lines.
DeleteTransferOrderLines	string orderId, IEnumerable<TransferOrderLine> transferOrderLines, QueryResultSettings queryResultSettings	PageResult<Transfer OrderLine>	Employee	Deletes the transfer order lines.
GetTransferOrderComments	string orderId, QueryResultSettings queryResultSettings	PageResult<Comment>	Employee	Gets the comments of the specified transfer order.
AddTransferOrderComment	string orderId, string commentedBy, string comment	Comment	Employee	Gets the comments of the specified transfer order.
GetTransferPackingSlip	string orderId, string voucherId, ReceiptRetrievalCriteria criteria, QueryResultSettings queryResultSettings	PageResult<Receipt>	Employee	Gets the packing slip for the specified transfer order journal.
PatchEntity	TransferOrder entity	TransferOrder	Employee	Saves a transfer order to the local database.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetEntityByKey	string orderId	TransferOrder	Employee	Gets a transfer order by order identifier.
DeleteEntity	TransferOrder entity	void	Employee	Deletes the specified transfer order.
CreateEntity	TransferOrder entity	TransferOrder	Employee	Creates transfer order.

## Purchase order controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
Get	QueryResultSettings queryResultSettings	PageResult<Purchase Order>	Employee	Gets open purchase orders for the store.
Commit	string orderId	Void	Employee	Commits a purchase order.
PatchEntity	PurchaseOrder entity	PurchaseOrder	Employee	Saves a purchase order to the local database.
GetEntityByKey	string orderId	PurchaseOrder	Employee	Get a purchase order by order identifier.

## Org units controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
Get	QueryResultSettings queryResultSettings	PageResult<OrgUnit >	Application,Employee , Customer, Anonymous	Gets all organization as IQueryable.
GetOrgUnitLocations ByArea	SearchArea searchArea, QueryResultSettings queryResultSettings	PageResult<OrgUnitL ocation>	Application,Employee , Customer, Anonymous	Finds stores in a defined area.
SearchOrgUnitLocati ons	OrgUnitLocationSear chCriteria orgUnitLocationSear chCriteria, QueryResultSettings queryResultSettings	PageResult<OrgUnitL ocation>	Application,Employee , Customer, Anonymous	Search stores with specified filter criteria within current fulfillment group.
GetAvailableInventor y	string itemId, string variantId, string barcode, QueryResultSettings queryResultSettings	PageResult<OrgUnit Availability>	Application,Employee , Customer, Anonymous	Get available inventory across all stores for an item identifier or barcode.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetProductAvailability	long productId, QueryResultSettings queryResultSettings	PageResult<OrgUnit Availability>	Application,Employee , Customer, Anonymous	Get available inventory across all stores for a product.
SearchProductAvailability	long productId, OrgUnitAvailabilitySe archCriteria orgUnitAvailabilitySe archCriteria, QueryResultSettings queryResultSettings	PageResult<OrgUnit Availability>	Application,Employee , Customer, Anonymous	Search available inventory across all stores for a product.
GetAvailableInventoryNearby	IEnumerable<ItemUn it> itemIds, SearchArea searchArea, QueryResultSettings queryResultSettings	PageResult<OrgUnit Availability>	Application,Employee , Customer, Anonymous	Get available inventory of nearby stores for given list of items in defined search area.
GetTillLayout	int? height, int? width	TillLayout	Application,Employee , Customer, Anonymous	Gets a single till layout.
GetOrgUnitConfigura tion		ChannelConfiguratio n	Application,Employee , Customer, Anonymous	Gets the configuration for the current organization unit.
Search	SearchStoreCriteria storeSearchCriteria, QueryResultSettings queryResultSettings	PageResult<OrgUnit >	Application,Employee , Customer, Anonymous	Searches for organization unit by the given search query.
GetTerminalInfo	string orgUnitNumber, int deviceType, QueryResultSettings queryResultSettings	PageResult<Terminal Info>	Employee,	Retrieves terminal and device association information data of the store.
GetProductAvailability ByDimensions	IEnumerable<string> inventLocationIds, long productId, IEnumerable<Produc tDimensionCombinat ion> productDimensionCo mbinations, QueryResultSettings queryResultSettings	PageResult<OrgUnit Availability>	Application,Employee , Customer, Anonymous	Gets orgUnit availabilities at each specified inventory location based on the given product dimensions.
GetStoreHours	string storeNumber	StoreHours	Application,Employee , Customer, Anonymous	Retrieve the store hours for a given store number.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetEntityByKey	string orgUnitNumber	OrgUnit	Application,Employee , Customer, Anonymous	Gets organization entity by key.

## Catalogs controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCatalogs	long channelId, bool activeOnly, QueryResultSettings queryResultSettings	PageResult<ProductCatalog>	Application,Employee , Customer, Anonymous	Gets catalogs by OData query.

## Categories controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCategories	long channelId, QueryResultSettings queryResultSettings	PageResult<Category>	Application,Employee , Customer, Anonymous	Gets categories by OData query.
GetChildren	long channelId, long categoryId, QueryResultSettings queryResultSettings	PageResult<Category>	Application, Employee, Anonymous	Gets subcategories by given Channel Id and Category Id.
GetAttributes	long categoryId, QueryResultSettings queryResultSettings	PageResult<AttributeCategory>	Application	Gets categories' attributes by OData query.
Get	QueryResultSettings queryResultSettings	PageResult<Category>	Application, Employee, Anonymous	Gets full list of categories as IQueryable.

## AppInfo controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
UpdateApplicationVersion	string appVersion	void	Employee	Updates the POS device's current application version.

## Attribute controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetAttributeDefinitions	AttributeDefinitionCriteria attributeDefinitionCriteria, QueryResultSettings queryResultSettings	PageResult<AttributeDefinition>	Employee	Gets the attribute definitions by an attribute group identifier.

## Attribute group controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetAttributeGroupDefinitions	AttributeGroupDefinitionCriteria attributeGroupDefinitionCriteria, QueryResultSettings queryResultSettings	PageResult<AttributeGroupDefinition>	Employee	Gets the attribute group definitions by collection of attribute group identifiers.

## Audit event controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
RegisterAuditEvent	AuditEvent auditEvent	void	Employee	Performs the audit event saving operation.
RegisterAndGetAuditEvent	AuditEvent auditEvent	AuditEvent	Anonymous, Customer, Device, Employee, Application	Performs the audit event saving operation.

## Shifts controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetShift	long shiftId, string terminalId	Shift	Employee	Gets the shift by shift id and terminal id.
GetByStatus	int statusValue, QueryResultSettings queryResultSettings	PageResult<Shift>	Employee	Gets the shifts by status.
GetByStatusFilterByUserRole	int statusValue, bool filterByUserRole, QueryResultSettings queryResultSettings	PageResult<Shift>	Employee	Gets the shifts by status.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetByRetrievalCriteria	ShiftRetrievalCriteria shiftRetrievalCriteria, QueryResultSettings queryResultSettings	PageResult<Shift>	Employee	Gets the shifts by retrieval criteria.
UpsertAndValidateShifts	long? shiftId, string terminalId, IEnumerable<Shift> shifts	bool	Employee	Inserts or update given shifts and validate them.
DeleteShifts		bool	Employee	Delete shifts is not supported in the online context.
Open		Shift	Employee	Opens a new shift.
Close	long shiftId, string terminalId, string transactionId, bool forceClose	Shift	Employee	Closes the shift for the given terminal.
BlindClose	long shiftId, string terminalId, string transactionId, bool forceClose	Shift	Employee	Blind closes a shift.
ForceDelete	long shiftId, string terminalId, string transactionId	void	Employee	Forcefully deletes a shift. Used to delete invalid shifts.
Resume	long shiftId, string terminalId, string cashDrawer	Shift	Employee	Resumes a shift.
Use	long shiftId, string terminalId	Shift	Employee	Uses an existing shift.
Suspend	long shiftId, string terminalId, string transactionId	Shift	Employee	Suspends a shift.
PostShift	Shift shift	HttpResponseMessage	Employee	Handles POST requests that create new shift
PatchShift	long shiftId, string terminalId, Delta<Shift> delta	Shift	Employee	Handles Patch request that update existing shift.
GetXReport	long shiftId, string terminalId, string transactionId, string hardwareProfileId	Receipt	Employee	Gets receipt for X report.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetZReport	string transactionId, string hardwareProfileId	Receipt	Employee	Gets receipt for Z report.
ValidateCashDrawerLimit	string shiftTerminalId, long shiftId	void	Employee	Gets all suspended carts for given shift.
GetSuspendedCartsByShift	string shiftTerminalId, long shiftId, QueryResultSettings queryResultSettings	PageResult<SuspendedCart>	Employee	voids the suspended transactions for given shift.
VoidSuspendedCarts	long shiftId, string shiftTerminalId	void	Employee	voids the suspended transactions for given shift.

## Async service controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetDownloadInterval	string dataStoreName	string	Device	Gets download interval.
GetUploadInterval	GetUploadInterval	string	Device	Gets upload interval.
GetTerminalDataStoreName	string terminalId	string	Device	Gets data store name.
GetDownloadLink	string dataStoreName, long downloadSessionId	string	Device	Gets download link.
GetDownloadSessions	string dataStoreName, QueryResultSettings queryResultSettings	PageResult<DownloadSession>	Device	Gets the download sessions.
GetInitialDownloadSessions	string dataStoreName, QueryResultSettings queryResultSettings	PageResult<DownloadSession>	Device	Gets initial download sessions.
GetUploadJobDefinitions	string dataStoreName, QueryResultSettings queryResultSettings	IEnumerable<string>	Device	Gets the download sessions.
UpdateDownloadSession	DownloadSession downloadSession	bool	Device	Update download session status.
PostOfflineTransactions	IEnumerable<string> offlineTransactionForMPOS	bool	Device	Posts offline transactions.



## Card type controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCardTypes	QueryResultSettings queryResultSettings	PageResult <CardType Info>	Application,Employee , Customer, Anonymous	Returns the list of card types.
GetSupportedPaymentCardTypes	QueryResultSettings queryResultSettings	PageResult <string>	Application,Customer ,Anonymous	Returns the list of payment cards supported by the payment connector.

## Commission sales group controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetCommissionSalesGroups	QueryResultSettings queryResultSettings	PageResult <Commiss ionSalesGroup>	Employee	Gets collection of commission sales groups for the channel.
SearchCommissionSalesGroups	string searchText, QueryResultSettings queryResultSettings	PageResult <Commiss ionSalesGroup>	Employee	Searches for the commission sales groups for the channel for a given search text.

## Environment configuration controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetEnvironmentConfiguration		EnvironmentConfigur ation	Anonymous, Employee, Application	Gets a single environment configuration.
GetExtensionProfile		ExtensionProfile	Anonymous, Employee, Application	Gets the extension profile which can be used to download extension package and communicate with micro-services.

## Extension package definition controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetExtensionPackageDefinitions	QueryResultSettings queryResultSettings	IEnumerable<Extensi onPackageDefinition >	Device, Employee, Application	Gets the configured extension package definitions.

## Extensible enumeration package definition controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetExtensibleEnumerations	QueryResultSettings queryResultSettings	IEnumerable<ExtensibleEnumerationContainer>	Device, Employee, Application, Anonymous, Customer, Storefront	Gets all extensible enumeration classes.

## Loyalty card controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
IssueLoyaltyCard	LoyaltyCard loyaltyCard	LoyaltyCard	Employee, Customer	Issues a new loyalty card.
GetLoyaltyCard	string cardNumber	LoyaltyCard	Employee, Customer	Gets a loyalty card.
GetCustomerLoyaltyCards	string accountNumber, QueryResultSettings queryResultSettings	PageResult<LoyaltyCard>	Employee, Customer	Gets the customer loyalty cards.
GetLoyaltyCardTransactions	string cardNumber, string rewardPointId, QueryResultSettings queryResultSettings	PageResult<LoyaltyCardTransaction>	Employee, Customer	Gets the loyalty card transactions.
GetLoyaltyRewardPointActivityTimeline	string cardNumber, string rewardPointId, QueryResultSettings queryResultSettings	PageResult<LoyaltyRewardPointActivity>	Employee, Customer	Gets the timeline activity for the reward point of a loyalty card.
GetLoyaltyRewardPointActivityTimelineForExpiredPoints	string cardNumber, string rewardPointId, QueryResultSettings queryResultSettings	PageResult<LoyaltyRewardPointActivity>	Employee, Customer	Gets the expired points timeline activity for the reward point of a loyalty card.
GetLoyaltyRewardPointsExpiringSoon	string cardNumber, string rewardPointId, int daysToExpiry, QueryResultSettings queryResultSettings	PageResult<LoyaltyRewardPointActivity>	Employee, Customer	Gets the loyalty card reward points that are going to expire soon.

## Non sales transaction tender operations controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
-----	-----------	--------------	--------------------------	-------------

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetNonSalesTransactions	string shiftId, string shiftTerminalId, int nonSalesTenderTypeValue, QueryResultSettings queryResultSettings	PageResult<NonSalesTransaction>	Employee	Gets the aggregated amount for non sale tender operation.
CreateNonSalesTransaction	NonSalesTransaction nonSalesTransaction	NonSalesTransaction	Employee	Performs saving drawer type of operations like declare start amount / tender removal / float entry.
GetAffiliations	QueryResultSettings queryResultSettings	PageResult<Affiliation>	Employee	Gets affiliations.

## Operations controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetOperationPermissionById	int operationId	OperationPermission	Employee	Gets Operation permission by using operation identifier.
GetOperationPermissions	QueryResultSettings queryResultSettings	PageResult<OperationPermission>	Employee	Returns a collection of operation permissions.
SearchJournalTransactions	TransactionSearchCriteria searchCriteria, QueryResultSettings queryResultSettings	PageResult<Transaction>	Employee	Returns a collection of transactions matching the specified search criteria.
GetInventoryAvailableToPromise	long productId, string itemId, string inventoryLocationId, QueryResultSettings queryResultSettings	PageResult<InventoryAvailableToPromise>	Employee	Get available inventory across all stores for a product.
VoidSuspendedTransactions	IEnumerable<string> suspendedCartIds	void	Employee	Void the suspended transactions specified by given cart ids.

## Shift reconciliation lines controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
-----	-----------	--------------	--------------------------	-------------

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetShiftReconciliationLines	ShiftReconciliationLineRetrievalCriteria shiftReconciliationLineRetrievalCriteria, QueryResultSettings queryResultSettings	PageResult<ShiftReconciliationLine>	Employee	Gets download interval.
ReconcileLines	IEnumerable<ShiftReconciliationLine> lines, string description	void	Employee	Reconciles the lines.
UndoReconciliation	IEnumerable<ShiftReconciliationLine> lines	void	Employee	Unreconciles all the lines that are a part of any of the groups in the lines passed in.

## Stock count journal controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
Get	QueryResultSettings queryResultSettings	PageResult<StockCountJournal>	Employee	Gets StockCountJournal entities as IQueryable.
Sync	QueryResultSettings queryResultSettings	PageResult<StockCountJournal>	Employee	Syncs the Stock Count journal from AX to RetailServer DB and gets the current list of SC journal from DB.
SyncTransactions	string journalId, QueryResultSettings queryResultSettings	PageResult<StockCountJournalTransaction>	Employee	Syncs the Stock Count journal from AX to RetailServer and gets the current list of journal transactions.
RemoveJournal	string journalId	void	Employee	Deletes the stock count journals from local.
RemoveTransaction	string journalId, string itemId, string inventSizeld, string inventColorId, string inventStyleId, string configId	void	Employee	Deletes the stock count journal transaction from local.

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
RemoveStockCountLineByLineId	string journalId, long stockCountLineId	void	Employee	Deletes the stock count journal transaction from local by stock count line identifier.
RemoveStockCountLineByProductRecId	string journalId, long productRecId	void	Employee	Deletes the stock count journal transaction from local by product identifier.
Commit	string journalId	void	Employee	Commits the list of Stock journal transactions to AX.
GetEntityByKey	string journalId	StockCountJournal	Employee	Creates journal entity.
UpdateEntity	StockCountJournal entity	StockCountJournal	Employee	Updates journal entity.
PatchEntity	StockCountJournal entity	StockCountJournal	Employee	Partially updates journal entity.

## Scan result controller

API	PARAMETER	RETURN VALUE	SUPPORTED COMMERCE ROLES	DESCRIPTION
GetEntityByKey	string scannedText	ScanResult	Employee	Gets the ScanResult entity by key.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a Retail Server extension API (Retail SDK version 10.0.11 and later)

2/18/2021 • 8 minutes to read • [Edit Online](#)

This topic explains how to create a new Retail Server application programming interface (API), and how to expose it so that point of sale (POS) or other clients can consume it. Modification of the existing Retail Server APIs isn't supported.

This topic applies to Retail software development kit (SDK) version 10.0.11 and later.

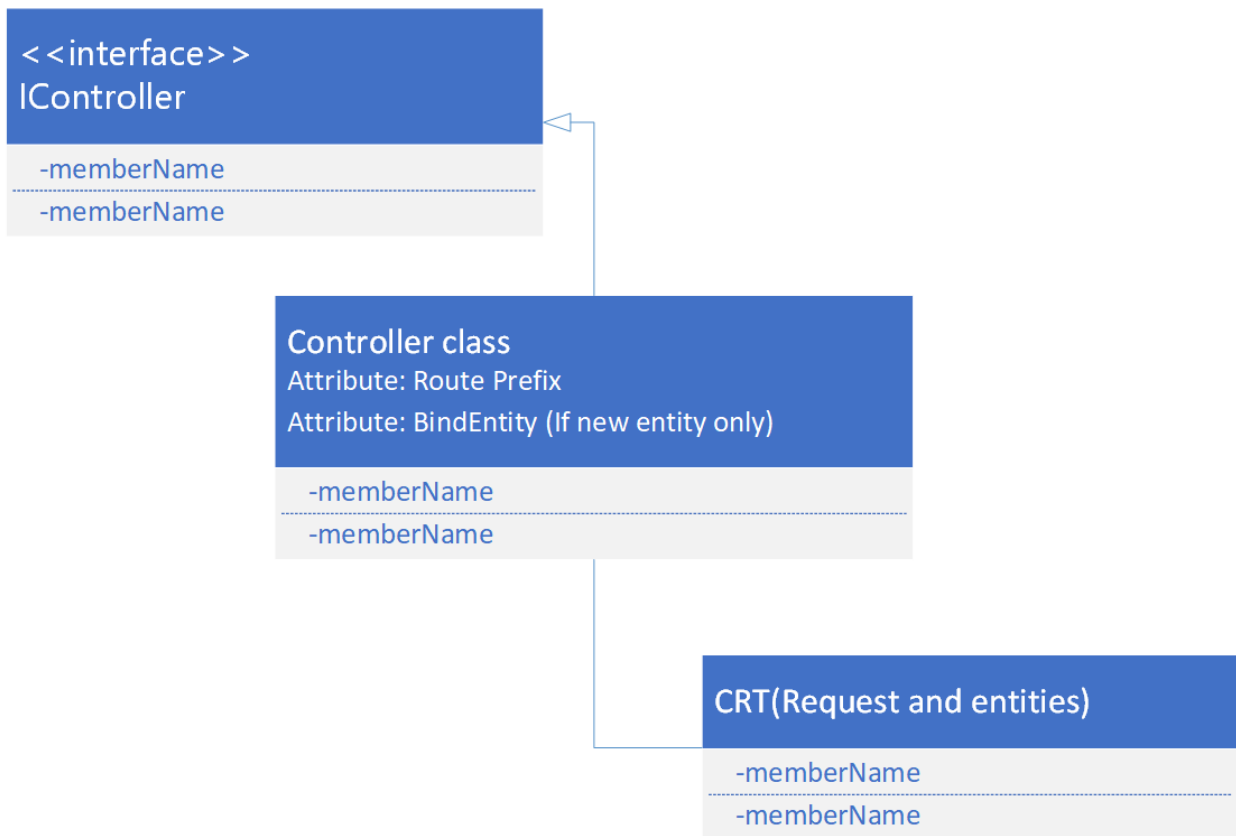
The Retail SDK includes only a few samples of end-to-end Retail Server extensions that include the Commerce runtime (CRT). You can use these samples as templates to start your extensions. You can find the sample extensions in the `RetailSDK\SampleExtensions\RetailServer` folder.

## End-to-end sample repository in the Retail SDK

SAMPLE EXTENSION (RETAILSDK\SAMPLEEXTENSIONS\RETAILSERVER)	CRT SAMPLE (RETAILSDK\SAMPLEEXTENSIONS\COMMERCERUNTIME)	POS SAMPLE (RETAILSDK\POS\EXTENSIONS)
Extensions.StoreHoursSample	Extensions.StoreHoursSample	StoreHoursSample
Extensions.SalesTransactionSignatureSample	Extensions.SalesTransactionSignatureSample	SalesTransactionSignatureSample
Extensions.PrintPackingSlipSample	Extensions.PrintPackingSlipSample	
Extensions.CrossLoyaltySample	Extensions.CrossLoyaltySample	

## Extension class diagram

The following illustration shows the class structure of the extension.



**NOTE**

Retail server does not support loading both IController and CommerceController extensions. If you include both type of extensions, then Retail server load will fail. Extensions should have either IController or CommerceController. If you are migrating to the IController extension, migrate all the Retail server extensions to IController.

## Create a new Retail Server API

1. Create the CRT extension. You must create the CRT extension before you create the Retail Server extension. A Retail Server API should have no logic except logic that calls the CRT with the parameters.
2. Create a new C# class library project that uses the Microsoft .NET Framework version netstandard 2.0 as the target framework. Alternatively, use one of the Retail Server samples in the Retail SDK as a template.
3. In the Retail Server extension project, add a reference to your CRT extension library or project. This reference lets you call the CRT request, response, and entities.
4. In the Retail Server extension project, add the **Microsoft.Dynamics.Commerce.Runtime.Hosting.Contracts** package using the NuGet package manager. The NuGet packages can be found in the **RetailSDK\pkgs** folder.
5. Create a new public controller class and extend it from **IController**. This controller class will contain the method that the Retail Server API must expose, the controller class must be public.
6. Inside the controller class, add methods to call the CRT request. Don't extend the new controller class from existing controller classes such as **CustomerController**, **SalesOrdersController**, or **ProductController**. Extension classes must extend only the **IController** class.
7. Add the **RoutePrefix** attribute on the controller class (Controller class name).

```
[RoutePrefix("SimpleExtension")]
```

8. Add the **BindEntity** attribute on the controller class. This attribute is required if you're creating a new controller and exposing an entity.

```
[BindEntity(typeof(SimpleEntity))]
```

**NOTE**

Steps 7 and 8 are required if the extension class is bound to an entity. These steps are not required for an unbounded controller class returning simple types, not any entity.

The following sample code creates a simple Retail Server API to return an entity, a string, and a bool value. The CRT request and response used in the sample is not included in this sample. For an example of the CRT request and response, see [Commerce runtime \(CRT\) extensibility and triggers](#).

**Sample code for a controller class bounded to a custom entity****NOTE**

Extension code should not bound the existing OOB entity, such as Customer or Product.



```

// New extended controller.
[RoutePrefix("SimpleExtension")]
[BindEntity(typeof(SimpleEntity))]
public class SimpleExtensionController : IController
{
    /// <summary>
    /// The action to get the string value.
    /// </summary>
    /// <param name="context">The context parameters.</param>
    /// <param name="stringValue">The string value parameters.</param>
    /// <returns>The string value.</returns>
    [HttpPost]
    [Authorization(CommerceRoles.Customer, CommerceRoles.Employee)]
    public async Task<string> GetStringValue(IEndpointContext context, string stringValue)
    {
        GetStringValueResponse resp = await context.ExecuteAsync<GetStringValueResponse>
            (new GetStringValueRequest(stringValue)).ConfigureAwait(false);
        return resp.StringValue;
    }

    /// <summary>
    /// The action to get the bool value.
    /// </summary>
    /// <param name="context">The context parameters.</param>
    /// <param name="boolValue">The string value parameters.</param>
    /// <returns>The bool value.</returns>
    [HttpPost]
    [Authorization(CommerceRoles.Customer, CommerceRoles.Employee)]
    public async Task<bool> GetBoolValue(IEndpointContext context, string boolValue)
    {
        GetBoolValueResponse resp = await context.ExecuteAsync<GetBoolValueResponse>
            (new GetBoolValueRequest(boolValue)).ConfigureAwait(false);
        return resp.BoolValue;
    }

    /// <summary>
    /// The action to get the simple entity.
    /// </summary>
    /// <param name="context">The context parameters.</param>
    /// <param name="name">The name parameters.</param>
    /// <returns>The simple entity.</returns>
    [HttpPost]
    [Authorization(CommerceRoles.Customer, CommerceRoles.Employee)]
    public async Task<SimpleEntity> GetSimpleEntity(IEndpointContext context, string name)
    {
        GetSimpleEntityResponse resp = await context.ExecuteAsync<GetSimpleEntityResponse>
            (new GetSimpleEntityRequest(name)).ConfigureAwait(false);
        return resp.SimpleEntityObj;
    }
}

```

### Sample code for a controller class not bounded to a custom entity

```

namespace Contoso.UnboundController.Sample
{
    using System.Threading.Tasks;
    using Microsoft.Dynamics.Commerce.Runtime.DataModel;
    using Microsoft.Dynamics.Commerce.Runtime.Hosting.Contracts;

    /// <summary>
    /// An extension unbounded controller sample.
    /// </summary>
    public class UnboundController : IController
    {
        /// <summary>
        /// A simple GET endpoint to demonstrate GET endpoints on an unbound controller.
        /// </summary>
        /// <returns>A simple true value to indicate the endpoint was reached.</returns>
        [HttpGet]
        [Authorization(CommerceRoles.Anonymous, CommerceRoles.Application, CommerceRoles.Customer,
CommerceRoles.Device, CommerceRoles.Employee, CommerceRoles.Storefront)]
        public Task<bool> SampleGet()
        {
            return Task.FromResult(true);
        }

        /// <summary>
        /// A simple POST endpoint to demonstrate POST endpoints on an unbound controller.
        /// </summary>
        /// <returns>A simple true value to indicate the endpoint was reached.</returns>
        [HttpPost]
        [Authorization(CommerceRoles.Customer, CommerceRoles.Device, CommerceRoles.Employee)]
        public Task<bool> SamplePost()
        {
            return Task.FromResult(true);
        }
    }
}

```

The Retail Server APIs support different authorization roles. Access to the controller method is permitted based on the authorization roles that are specified in the controller method **Authorizations** attribute. The following example shows the supported authorization roles. Extension code should not use the **CommerceAuthorization** attribute instead of the **Authorizations** attribute. The **CommerceAuthorization** attribute is only supported in SDK versions earlier than 10.0.11.

```

// Represents the type of logon type.
[DataContract]
public static class CommerceRoles
{
    // Anonymous Role.
    [DataMember]
    public const string Anonymous = "Anonymous";

    // SharePoint Role used by Connector.
    [DataMember]
    public const string Storefront = "Storefront";

    // Employee Role.
    [DataMember]
    public const string Employee = "Employee";

    // Customer Role.
    [DataMember]
    public const string Customer = "Customer";

    // Represents the Device level of authentication.
    [DataMember]
    public const string Device = "Device";

    // Represents Application level of authentication.
    [DataMember]
    public const string Application = "Application";

    // The list of all possible Microsoft.Dynamics.Commerce.Runtime.DataModel.CommerceRoles values.
    public static readonly string[] All;
}

```

## Support paging in Retail Server APIs

Starting in release 10.0.18, if the API requires paging you can add the **QueryResultSettings** parameter to the API and pass the value from the client. **QueryResultSettings** contains **PagingInfo** and other parameters for records to fetch or skip.

The extension can pass **QueryResultSettings** to the CRT request, which the CRT request can use when there is a database query.

The full sample code is available in the Retail SDK:

RetailSDK\SampleExtensions\CommerceRuntime\Extensions.StoreHoursSample\StoreHoursDataService.cs  
 RetailSDK\SampleExtensions\RetailServer\Extensions.StoreHoursSample\StoreHoursController.cs"

```

[HttpPost]
[Authorization(CommerceRoles.Anonymous, CommerceRoles.Customer, CommerceRoles.Device,
CommerceRoles.Employee)]
public async Task<PagedResult<SampleDataModel.StoreDayHours>> GetStoreDaysByStore(IEndpointContext
context, string StoreNumber, QueryResultSettings queryResultSettings)
{
    var request = new GetStoreHoursDataRequest(StoreNumber) { QueryResultSettings =
queryResultSettings };
    var hoursResponse = await context.ExecuteAsync<GetStoreHoursDataResponse>
(request).ConfigureAwait(false);
    return hoursResponse.DayHours;
}

```

```

private async Task<Response> GetStoreDayHoursAsync(GetStoreHoursDataRequest request)
{
    ThrowIf.Null(request, "request");

    using (DatabaseContext databaseContext = new DatabaseContext(request.RequestContext))
    {
        var query = new SqlPagedQuery(request.QueryResultSettings)
        {
            DatabaseSchema = "ext",
            Select = new ColumnSet("DAY", "OPENTIME", "CLOSINGTIME", "RECID"),
            From = "CONTOSORETAILSTOREHOURSVIEW",
            Where = "STORENUMBER = @storeNumber",
        };

        query.Parameters["@storeNumber"] = request.StoreNumber;
        return new GetStoreHoursDataResponse(await
        databaseContext.ReadEntityAsync<DataModel.StoreDayHours>(query).ConfigureAwait(false));
    }
}

```

### Register the extension

1. Build the extension project, and copy the binary to the `\RetailServer\webroot\bin\Ext` folder.
2. Update the Commerce Scale Unit `web.config` file in the `\RetailServer\webroot` folder by adding the new extension library name in the `extensionComposition` section.

```

<extensionComposition>
  <!-- Use fully qualified assembly names for ALL if you need to support loading from the Global
  Assembly Cache.
  If you host in an application with a bin folder, this is not required. -->
  <add source="assembly" value="SimpleExtensionSample" >
</extensionComposition>

```

3. In Internet Information Services (IIS), restart the Commerce Scale Unit to load the new extension.

### Validate the extension

1. To verify that the extension loaded successfully, you can browse the Retail Server metadata. Confirm that your entities and methods appear in the list. To browse the metadata, open a URL in the following format in a web browser:

```
https://RS-URL/Commerce/$metadata
```

2. To call the Retail Server extension in your client, you must generate the client Typescript proxy. You can then use the proxy to call your new Retail Server APIs from the client.

You don't have to add or include any `EdmModelExtender` files in the extension with the Retail Server extensions APIs. The files are required only if you're using Retail SDK version 10.0.10 or earlier.

### Debugging RS extension

To debug the RS extension project in Visual Studio. Go to **Debug > Attach to Process**. Select `w3wp.exe` (the IIS process for Retail Server). If there are multiple `w3wp.exe` processes, use the correct process based on the process ID. The retail server process ID can be found using **IIS > Worker processes** or by using the command prompt and the `tasklist` command.

## Generate the Typescript proxy for POS

The POS uses the Typescript proxy to access the Retail Server APIs and CRT entities. The proxy class acts as

manger class or wrapper to access the Retail server APIs without the proxy extension manually finding the Retail server API and entities metadata.

### Steps to generate the proxy files

1. In Visual Studio 2017, open the sample proxy template project from `\RetailSDK\Code\SampleExtensions\TypeScriptProxy\TypeScriptProxy.Extensions.StoreHoursSample\Proxies.TypeScriptProxy.Extensions.StoreHoursSample.csproj` in Visual Studio 2017. Rename the project if a new name is required.
2. Add the Retail Server extension project to this proxy template project as a project reference project. Remove the existing `StoreHoursSample` project reference.
3. Right-click the `Proxies.TypeScriptProxy.Extensions.StoreHoursSample.csproj` project, and then select `Edit Proxies.TypeScriptProxy.Extensions.StoreHoursSample.csproj`.
4. Under the `<RetailServerExtensionAssemblies>` node, specify your extension Retail Server assembly name. The following example shows how to add the assembly name.

```
<ItemGroup>
  <RetailServerExtensionAssemblies
    Include="..\..\RetailServer\Extensions.Sample\bin\$(Configuration)\net461\$(AssemblyNamePrefix).RetailServer.Extension.Sample.dll" />
</ItemGroup>
```

5. Under the `<Copy>` node, update the `DestinationFolder` path of your POS extension folder, so that generated proxy files are automatically copied to the POS extension folder automatically. The generated proxy files will also be copied to `\RetailSDK\Code\SampleExtensions\TypeScriptProxy\TypeScriptProxy.Extensions.StoreHoursSample\DataService`. The following example shows how to update the path.

```
<Copy SourceFiles="@{(GeneratedDataServiceContracts)"
  DestinationFolder="$(SdkRootPath)\POS\Extensions\Sample\DataService" SkipUnchangedFiles="true" />
```

6. After the changes are completed, build the proxy project to generate the TypeScript proxy files. When the build is completed, the proxy files will be available in the `\RetailSDK\Code\SampleExtensions\TypeScriptProxy\TypeScriptProxy.Extensions.StoreHoursSample\DataService` folder and the folder that is specified in the `Copy` command. The path and folder path can vary, depending on the folder structure.

## Retail server extension in offline

A Retail Server extension built using the `Microsoft.Dynamics.Commerce.Runtime.Hosting.Contracts` API can be used in an offline implementation. You don't need to generate a separate C# proxy library. Copy the Retail Server extension library in the `\Microsoft Dynamics 365\70\Retail Modern POS\ClientBroker\ext` folder and update the `RetailProxy.MPOSOOffline.ext` config file to include the this library. This extension must only generate the Typescript proxy. SDK samples can be found in the `\RetailSDK\SampleExtensions\TypeScriptProxy` folder.

The following example shows how to update the `add` element in the `RetailProxy.MPOSOOffline.ext` configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<retailProxyExtensions>
  <composition>
    <add source="assembly" value="Contoso.RetailServer.StoreHoursSample" />
  </composition>
</retailProxyExtensions>
```

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create a new Retail Server extension API (Retail SDK version 10.0.10 and earlier)

2/18/2021 • 5 minutes to read • [Edit Online](#)

This document explains how to create a new Commerce Scale Unit application programming interface (API), and how to expose it so that POS or other clients can consume it. Modification of the existing Commerce Scale Unit APIs isn't supported.

This topic applies to Retail SDK version 10.0.10 and earlier.

The Retail software development kit (SDK) includes only a few samples of end-to-end Commerce Scale Unit extensions that include the Commerce Runtime (CRT). You can use these samples as templates to start your extensions. You can find the sample extensions in the RetailSDK\SampleExtensions\RetailServer folder.

## NOTE

The CommerceController extension model will be soon deprecated. You will need to migrate all Retail server extensions to IController. Having both CommerceController and IController is not supported. For more information about IController, see [Create a new Retail Server extension API \(Retail SDK version 10.0.11 and later\)](#).

## End-to-end sample repository in the Retail SDK

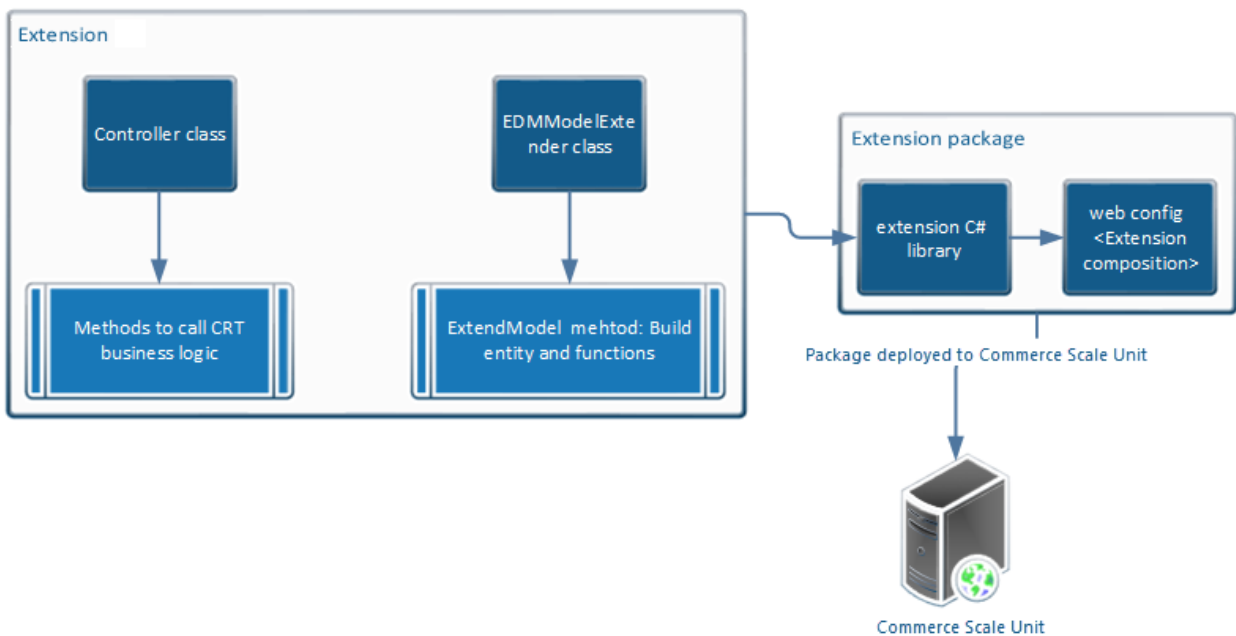
SAMPLE EXTENSION (RETAILSDK\SAMPLEEXTENSIONS\RETAILSERVER)	CRT SAMPLE (RETAILSDK\SAMPLEEXTENSIONS\COMMERCERUNTIME)	POS SAMPLE (RETAILSDK\POS\EXTENSIONS)
Extensions.StoreHoursSample	Extensions.StoreHoursSample	StoreHoursSample
Extensions.SalesTransactionSignatureSample	Extensions.SalesTransactionSignatureSample	SalesTransactionSignatureSample
Extensions.PrintPackingSlipSample	Extensions.PrintPackingSlipSample	
Extensions.CrossLoyaltySample	Extensions.CrossLoyaltySample	

## Create a new extension

Follow the steps in this section to create a new Commerce Scale Unit extension.

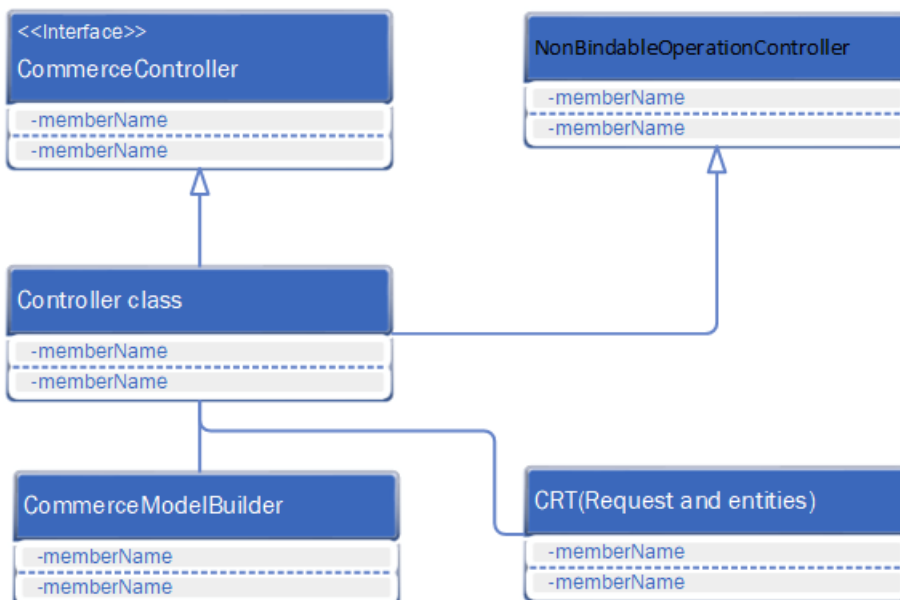
### End-to-end flow

The following illustration shows the flow of the extension.



### Extension class diagram

The following illustration shows the class structure of the extension.



#### NOTE

Retail server does not support loading both IController and CommerceController extensions. If you include both type of extensions, then Retail server load will fail. Extensions should have either IController or CommerceController. If you are migrating to the IController extension, migrate all the Retail server extensions to IController.

### Steps

1. Before you create the Commerce Scale Unit extension, create the CRT extension. Commerce Scale Unit APIs should have no logic except logic that calls the CRT with the parameters.
2. Create a new C# class library project that uses the Microsoft .NET Framework version netstandard 2.0 as the target framework.
3. In the Commerce Scale Unit extension project, add a reference to your CRT extension library or project. This reference lets you call the CRT request and response. It also lets you use the entities from the Commerce Scale Unit extension project.



4. In the Commerce Scale Unit extension project, create a new controller class that extends **NonBindableOperationController** or **CommerceController**. The base class depends on your scenario. This controller class will contain the method that must be exposed by the Commerce Scale Unit API. Inside the controller class, add methods to call the CRT request.

```

/// <summary>;
/// The controller to retrieve a new entity.
/// <summary>
[ComVisible(false)]
public class SampleController : CommerceController<SampleEntity, long>;
{
    ///<summary>;
    /// Gets the controller name used to load extended controller.
    /// <summary>
    public override string ControllerName
    {
        get { return "SampleEntity"; }
    }
    /// <summary>;
    /// Gets the sample entity.
    /// <summary>;
    /// <param name="parameters">The parameters to this action.</param>
    /// <returns>The list of sample entity.</returns>
    [HttpPost]
    [CommerceAuthorization(CommerceRoles.Anonymous, CommerceRoles.Customer, CommerceRoles.Device,
CommerceRoles.Employee)]
    public System.Web.OData.PageResult<SampleEntity> GetSampleEntity(ODataActionParameters
parameters)
    {
        if (parameters == null)
        {
            throw new ArgumentNullException("parameters");
        }
        var runtime = CommerceRuntimeManager.CreateRuntime(this.CommercePrincipal);
        QueryResultSettings queryResultSettings = QueryResultSettings.SingleRecord;
        queryResultSettings.Paging = new PagingInfo(10);
        var request = new CRTDataRequest((string)parameters["key"]) { QueryResultSettings =
queryResultSettings };
        PagedResult<SampleEntity> sample = runtime.Execute<CRTDataResponse>(request, null);
        return this.ProcessPagedResults(sample);
    }
}

```

5. Create an **EdmModelExtender** (EDM) class that extends the **IEdmModelExtender** interface. This class contains the abstract data model that is used to describe the data that a Commerce Scale Unit API exposes. An Open Data Protocol (OData) Metadata Document is a representation of a service's data model that is exposed for client consumption. The central concepts in the EDM are entities, relationships, entity sets, actions, and functions.

The **IEdmModelExtender** interface contains the abstract **ExtendModel** method. When you extend this interface, you must implement the **ExtendModel** method. Inside the **ExtendModel** method, you build the EDM entities and functions that will be exposed to the client by using the **CommerceModelBuilder** class.

The **CommerceModelBuilder** class contains the build method that is used to build the entities and functions.

METHOD NAME	RETURN TYPE	DESCRIPTION
BuildEntity<TEntity>() where TEntity : class	EntityTypeConfiguration<TEntity>	This method builds an entity.

METHOD NAME	RETURN TYPE	DESCRIPTION
BuildEntitySet<TEntity>(string entitySetName) where TEntity : class	EntitySetConfiguration<TEntity>	This method builds an entity set.
BuildComplexType<TComplexType>() where TComplexType : class	ComplexTypeConfiguration<TComplexType>	This method builds a complex entity type.
BuildEnumType<TEnumType>()	EnumTypeConfiguration<TEnumType>	This method builds an enumeration type.
BindAction(string actionName)	ActionConfiguration	This method binds an action in the model builder. An action represents an HTTP POST request.
BindEntityAction<TEntity>(string actionName) where TEntity : class	ActionConfiguration	This method binds an entity action of the model. An action represents an HTTP POST request.
BindEntitySetAction<TEntity>(string actionName) where TEntity : class	ActionConfiguration	This method binds an entity set action. An action represents an HTTP POST request.
BindFunction(string functionName)	FunctionConfiguration	This method binds a function in the model builder. A function represents a HTTP GET request.
BindEntityFunction<TEntity>(string functionName) where TEntity : class	FunctionConfiguration	This method binds an entity function of the model. A function represents an HTTP GET request.
BindEntitySetFunction<TEntity>(string functionName) where TEntity : class	FunctionConfiguration	This method binds an entity set function. A function represents an HTTP GET request.

The following example shows how to extend the EDM model.

```

/// <summary>;
/// The class to extend the EDM model.
/// <summary>;
[Export(typeof(IEdmModelExtender))]
[ComVisible(false)]
public class EdmModelExtender : IEdmModelExtender
{
    /// <summary>;
    /// Extends the EDM model.
    /// <summary>;
    /// <param name="builder">The builder to build the EDM model.</param>
    public void ExtendModel(CommerceModelBuilder builder)
    {
        ThrowIf.Null(builder, "builder");
        // Extends entity sets.
        builder.BuildEntitySet<SampleEntity>("SampleEntity");
        // Extends entity set actions.
        var action = builder.BindEntitySetAction<SampleDataModel.StoreDayHours>("GetSampleEntity");
        action.Parameter<string>("Key");
        action.ReturnsCollectionFromEntitySet<SampleEntity>("SampleEntity");
    }
}

```

## NOTE

Do not duplicate the entity name in the EdmModelExtender class for the same entity. This will create multiple manager and Adapter classes during proxy generation. For example, if **CustomEntity1** is the new entity created by the extension code, in the EdmModelExtender, if the entity is named **CustomEntity1Sample**, then use the same name wherever it used. Do not use a different name for the same entity.

```
builder.BuildEntitySet< CustomEntity1>(**"CustomEntity1Sample");  
action.ReturnsCollectionFromEntitySet< CustomEntity1>(**"CustomEntity1Sample");
```

6. Build the extension project, and drop the binary into the `\RetailServer\webroot\bin\Ext` folder.
7. Update the Commerce Scale Unit web.config file in the `\RetailServer\webroot` folder by adding the new extension library name in the **extensionComposition** section.

```
<extensionComposition>  
<!-- Please use fully qualified assembly names for ALL if you need to support loading from the Global  
Assembly Cache.  
If you host in an application with a bin folder, this is not required. -->  
<add source="assembly" value="SampleExtension" >;  
</extensionComposition>
```

8. In Microsoft Internet Information Services (IIS), restart Commerce Scale Unit to load the new extension.
9. To verify that the extension was successfully loaded, you can browse the Commerce Scale Unit metadata, and confirm that your entities and methods appear in the list.

To browse the metadata, open a URL in the following format in a web browser:

```
https://Your Commerce Scale Unit URL/Commerce/$metadata
```

10. To call the Commerce Scale Unit extension in your client, you must generate the Commerce proxy. You can then use the proxy to call your new Commerce Scale Unit APIs from the client.

For information about how to generate the proxy, see [Typescript and C# proxies for Retail point of sale \(POS\)](#).

## NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Typescript and C# proxies for Retail point of sale (POS)

2/18/2021 • 6 minutes to read • [Edit Online](#)

When you create a new Retail Server API, you must generate the Commerce proxy by using the tools that are available as part of the Retail software development kit (SDK). For example, you must generate the Commerce proxy if you add a new Retail Server API.

## The Commerce proxy and when to use it

All clients use the proxy API to interact with Retail Server. The Commerce proxy abstracts the interface between Retail Server and the Commerce runtime (CRT). For example, you create a new entity and some business logic as request/response operations in CRT, and you add a new Retail Server API to expose that entity and those request/response operations. You now want to access the entity and the request/response operations in the point of sale (POS) to do some client logic. You can manually create all the entities and request/response metadata in the POS, and access the Retail Server API by using the correct parameters. However, this approach involves lots of additional overhead, because you must duplicate the entities, manager, and request/response code in two places, and you must also write lots of code.

The Commerce proxy reduces this effort by automatically generating the proxy for all the custom entities and request/response operations that are added in Retail Server. The proxy tool generates the required interface and all the required metadata, and abstracts the actual implementation. In that way, you can include the files in the extension projects, and can access the Retail Server APIs and the entities by using the metadata and interface that are generated.

## Proxy types

There two types of proxy to support cross-platform scenarios:

- **Typescript proxy** – The POS uses the Typescript proxy to access the Retail Server APIs and CRT entities. If the POS uses Retail Server, it requires the Typescript proxy. Otherwise, the POS can't communicate with the Retail Server for any operations or workflows.
- **C# proxy** – The POS uses the C# proxy when it's offline. (When the POS is offline, it communicates directly with CRT, without using Retail Server.) The POS also uses this proxy for the Dynamics e-Commerce platform. If you want your customization to work when the POS is offline, and you want your e-Commerce client to access the Retail Server APIs, you must generate the C# proxy.

The steps to generate the Typescript proxy and the C# proxy differ. The rest of this topic explains how to generate each type of proxy.

## Generate the Typescript proxy (10.0.11 or lower) Retail Server

If you are using Microsoft Dynamics Commerce version 10.0.12 or greater, follow the steps mentioned in [Create a new Retail Server extension API](#).

## IMPORTANT

Run MSBuild from the Retail SDK root folder to restore the CommerceProxyGenerator.exe package. Use the Visual Studio developer command prompt or MSBuild command prompt to restore all the packages in the reference folder before generating the proxy. If you do not perform this step, the CommerceProxyGenerator.exe package will not be available in the RetailSDK\Reference folder.

Use the CommerceProxyGenerator.exe file from the Retail SDK\Reference\Microsoft.Dynamics.Commerce.Tools.CoreProxyGenerator.<version\_number>\tools folder to generate the typescript proxy for the POS.

1. Before you generate the proxy, copy the customized Retail Server API, CRT, and other dependent libraries to the **Retail SDK\Reference** folder.
2. Open a Command Prompt window as an administrator, and go to the ...\**Retail SDK\Reference\Microsoft.Dynamics.Commerce.Tools.CoreProxyGenerator <version\_number>\tools** folder. Run the following command to generate the proxy. The proxy files will be generated in the same folder.

### POS Typescript proxy

```
CommerceProxyGenerator.exe <Path>\Microsoft.Dynamics.Retail.RetailServerLibrary.dll  
<FilePathNameForRetailServerExtensionDLL> /application:typescriptextensions
```

### e-Commerce Typescript proxy

```
CommerceProxyGenerator.exe <Path>\Microsoft.Dynamics.Retail.RetailServerLibrary.dll  
<FilePathNameForRetailServerExtensionDLL> /application:typescriptmoduleextensions
```

## NOTE

Use the Microsoft.Dynamics.Retail.RetailServerLibrary.dll file from  
\RetailSDK\References\Microsoft.Dynamics.Commerce.Tools.ExtensionsProxyGenerator.<version\_number>\build.

### Example

```
CommerceProxyGenerator.exe  
C:\\RetailSDK\\References\\Microsoft.Dynamics.Commerce.Tools.ExtensionsProxyGenerator.9.21.20042.5\\build\\M  
icrosoft.Dynamics.Retail.RetailServerLibrary.dll  
C:\\RetailSDK\\References\\Microsoft.Dynamics.RetailServer.CrossLoyaltySample.dll /a:typescriptextensions
```

In the above command, replace **Microsoft.Dynamics.RetailServer.CrossLoyaltySample.dll** with the name of your custom Retail Server extension library. Include the generated files in your POS project. The command generates two files that are based on your extension libraries: DataServiceEntities.g.ts and DataServiceRequests.g.ts.

## Generate the C# proxy (Commerce version 10.0.11 or lower)

## IMPORTANT

Retail Server extension built using this the Microsoft.Dynamics.Commerce.Runtime.Hosting.Contracts API can be used in and offline implementation, no need to generate separate C# proxy library. This step is required only for Commerce version 10.0.11 or lower or Retail Server extensions not using the Microsoft.Dynamics.Commerce.Runtime.Hosting.Contracts.

For each Retail Server extension, you must generate a separate proxy.

1. Navigate to **RetailSDK\SampleExtensions\RetailProxy\RetailProxy.Extensions.StoreHoursSample**.
2. In Microsoft Visual Studio, open the **Proxies.RetailProxy.Extensions.StoreHoursSample** project file.
3. Right-click, and select to unload the project.
4. Right-click the project, and select to edit the **Proxies.RetailProxy.Extensions.StoreHoursSample.csproj** file.
5. In the first property group section, update the following nodes:
  - **<RootNamespace>** – Specify your custom namespace.
  - **<AssemblyName>** – Specify your custom output library name for the proxy.
6. Update the **CommerceProxyGeneratorExtendedAssemblyPaths** element by specifying the name of your Retail Server extension library.

Here is an example.

```
<CommerceProxyGeneratorExtendedAssemblyPaths
Include="..\..\RetailServer\Extensions.StoreHoursSample\bin\$(Configuration)\net451\$(AssemblyNamePrefix).RetailServer.StoreHoursSample.dll" />
```

## NOTE

**.RetailServer.StoreHoursSample.dll** is the name of the Retail Server extension assembly, and the rest of the value is the prefix (if there is a prefix) and the path of the assembly where the proxy engine can find this assembly. The proxy is generated based on this assembly.

7. Save the file, and load the project again.
8. Rename the project according to your extension pattern.
9. After the project is loaded, delete the **StoreDayHoursManager.cs** file from the **Adapters** folder.
10. Add all the relevant CRT and Retail Server libraries to the proxy project as project or assembly references.
11. Rebuild the project.

You will see that a new **Interfaces.g.cs** file is generated inside the **Adapters** folder.

## NOTE

Before you build the proxy project, rebuild all your CRT and Retail Server extension libraries, and drop them into the **RetailSDK\References** folder.

12. Include the new **Interfaces.g.cs** file in the proxy project. However, don't modify this file.

13. Under the **Adapters** folder, add a new class file, and name it according to your extension pattern.
14. Extend the class from the interface manager class, and implement only the interface methods that are required.

#### NOTE

You can find the name of the interface manager class in the `Interfaces.g.cs` file.

In the following example, **IStoreDayHoursManager** is the name of the interface.

```
public interface IStoreDayHoursManager : Microsoft.Dynamics.Commerce.RetailProxy.IEntityManager
{
}
```

For the full sample code, see the **Proxies.RetailProxy.Extensions.StoreHoursSample** project under **RetailSDK\SampleExtensions\RetailProxy\RetailProxy.Extensions.StoreHoursSample**.

15. Inside the methods you will call the actual CRT request/response. Avoid including any logic in the proxy project. The proxy project should just call the CRT request/response.
16. Build the project.
17. Copy the output assembly, and paste it in the **RetailSDK\References** folder.
18. Navigate to the **RetailSDK\Assets** folder, and open the **RetailProxy.MPOSOffline.ext.config** file.
19. In the **composition** section, register the name of your new proxy library (that is, the assembly that was generated after you built your proxy project).

Here is an example.

```
<add source="assembly" value="Contoso.Commerce.RetailProxy.StoreHoursSample" />
```

In this example, the proxy library is named **Contoso.Commerce.RetailProxy.StoreHoursSample**. You should specify the name of your proxy library in the **value** field.

20. For manual testing, open the **RetailProxy.MPOSOffline.ext.config** file under **C:\Program Files (x86)\Microsoft Dynamics 365\70\Retail Modern POS\ClientBroker\ext**. Update the **composition** section with the name of the custom proxy library.

Here is an example.

```
<add source="assembly" value="Contoso.Commerce.RetailProxy.StoreHoursSample" />
```

21. For e-Commerce, you must initialize the proxy for the extensions before you call it from your e-Commerce project. In your e-Commerce **Startup.cs** file (or an equivalent, such as web project initialization), initialize **RetailServerContext** with the Extended data model(EDM) model for your Retail proxy extension. Otherwise, you will receive a runtime error when you try to call the proxy. You must complete this step only one time.

Here is an example.

```
RetailServerContext.Initialize(new IEdmModelExtension[]
{
    // /* BEGIN SDKSAMPLE_STOREHOURS
    new Contoso.Commerce.RetailProxy.StoreHoursSample.EdmModel(),
    // END SDKSAMPLE_STOREHOURS */
});
```

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Consume Retail Server APIs in external applications

2/18/2021 • 7 minutes to read • [Edit Online](#)

This topic describes how to consume the Retail Server APIs in external applications. Retail Server exposes the Open Data Protocol (OData) web endpoint so that external applications can consume the APIs. Retail Server also hosts the Commerce runtime (CRT) business logic and exposes it as OData endpoints. These endpoints are known as the Retail Server APIs.

External applications can consume the OData service through HTTPS. Retail Server provides multiple options for consuming the APIs:

- **Retail server proxy** – A strongly typed way to consume the APIs. There are multiple types of proxies that are exposed for different application types:
  - **C# proxy** – Server-side and native applications can consume the C# binary (class library) to access the APIs and other entities.
  - **TypeScript proxy** – Client applications can consume the .ts proxy file to access the APIs and other entities.
- **OData client** – The APIs can also be consumed from OData clients, from Postman, or by generating an HTTPS request to the Retail Server URL.

To browse the metadata, open the Retail Server URL in the following format in a web browser. The result is a list of all the Retail Server APIs, and input and output parameters.

```
https://RS-URL/Commerce/$metadata
```

## Security and authentication that are required to consume APIs

Access to each API is natively restricted according to the following roles:

- **Employee** – Access to APIs that are associated with this role requires point of sale (POS) device activation (a device token) and an authenticated employee.
- **Customer** – Access to APIs that are associated with this role requires an authenticated customer. E-Commerce sites generally use these APIs for operations such as retrieving order history and changing customer details.
- **Application** – Access to APIs that are associated with this role requires application-level authentication, such as Azure Active Directory (Azure AD) service-to-service authentication.
- **Anonymous** – APIs that are associated with this role are primarily used by e-Commerce sites without user authentication.
- **Customized APIs** – Access to APIs that are associated with this role can be restricted by using methods such as POS device activation, customer authentication, and anonymous authentication.

External application integration requires only the **Application** authentication flow to access the APIs. E-commerce application integration requires **Customer** authentication.

This topic is focused on setting up **Application** authentication flows and accessing the APIs by using the **Application** authentication context.

Before the APIs are accessed from an external application, the external application must be registered in Azure App registration, and details of the registered application must be added in the Commerce Headquarters.

For more information about authentication flows, see [Dynamics 365 Commerce authentication flows](#).

## Register the application in Azure App registration

Application registration establishes a trust relationship between your app and the Microsoft identity platform. The trust is unidirectional: your app trusts the Microsoft identity platform, not the other way around. Follow these steps to create the app registration.

1. Sign in to the [Azure portal](#).
2. If you have access to multiple tenants, use the **Directory + subscription** filter on the top menu to select the tenant that you want to register an application in.
3. Search for and select **Azure Active Directory**.
4. Under **Manage**, select **App registrations**, and then select **New registration**.
5. Enter a name for your application. Users of your app might see this name, and you can change it later.
6. Specify who can use this application as **Accounts in this organizational directory only (Microsoft only - Single tenant)**.
7. In the **Redirect URI** field, leave the default value. Don't change anything.
8. Select **Register** to complete the initial app registration.

When registration is completed, the Azure portal shows the **Overview** pane for the app registration. This pane includes the **Application (client) ID** field. The value of this field is known as the *application ID* or the *client ID*. It uniquely identifies your application in the Microsoft identity platform.

### Add a client secret

The client secret is also known as an *application password*. It's a string value that your app can use instead of a certificate to identify itself.

1. In the Azure portal, in **App registrations**, select your application.
2. Select **Certificates & secrets > New client secret**.
3. Add a description for your client secret.
4. Select a duration.
5. Select **Add**.
6. Be sure to record the secret's value so that you can use it in your client application code. **The secret's value is never displayed again after you leave this page.**

## Register the app in the Finance and Operations app so that Retail Server trusts it

1. In Commerce Headquarters, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
2. On the **Identity providers** FastTab, select the provider that begins with `HTTPS://sts.windows.net/`. The values on the **Relying parties** FastTab are set based on your selection.
3. On the **Relying parties** FastTab, select **Add**. Enter the client ID that was generated during the app registration in Azure. Set the **Type** field to **Confidential** and the **UserType** field to **Application**.
4. On the Action Pane, select **Save**.
5. Select the new relying party, and then, on the **Server resource IDs** FastTab, select **Add**. In the **Server Resource ID** column, enter the Retail Server URL.
6. On the Action Pane, select **Save**.
7. Go to **Retail and commerce > Retail and commerce IT > Distribution Schedule**, and run Commerce Data Exchange (CDX) job 1110.

# Access the APIs by using Postman

Several third-party tools let you authenticate with Azure services, compose and send Web API requests, and view responses. Postman is one of the most popular tools. [Download and install the Postman client tool.](#)

For this example, you will access the **GetOrderHistory** API. This API retrieves the order history for a specific customer. It can be accessed only by using the **Employee**, **Customer** or **Application** authorization context. Because your application has the **Application** authorization context, it can access the **GetOrderHistory** API and get the customer order history.

To access the API, you first generate the authorization token. You then use that authorization token to access the API.

For the full list of APIs, see [Commerce Scale Unit customer and consumer APIs.](#)

## Generate an authorization token by using Postman

1. In Postman, create a GET request that has the following request URL and body parameters.

### Request URL

```
https://login.microsoftonline.com/{tenant-id}/oauth2/token
```

#### NOTE

You can get the tenant ID from the **Overview** pane for your Azure app registration. It's shown next to the client ID.

### Body parameters

KEY	VALUE
grant_type	client_credentials
client_id	The client ID that was generated during Azure app registration
client_secret	The client secret that was generated during Azure app registration
resource	The Retail Server URL

2. After the request has finished running, the **access\_token** value will be generated in the response body. Copy this token value. You will use it to connect to the Retail Server.

## Get order history by using Postman

- In Postman, create a POST request that has the following request URL, parameters, and header parameters.

### Request URL

```
<https://Retail>serverurl/Commerce/Customers('2001')/GetOrderHistory
```

#### NOTE

Replace 2001 with your own customer ID.

#### Parameters

KEY	VALUE
\$top	10
api-version	7.3

#### Header parameters

KEY	VALUE
OUN	The operating unit number of this retail channel
authorization	id_token { access_token }

#### NOTE

For `access_token`, copy and paste the `access_token` value that was generated in the authorization request. Then prefix it with `id_token`.

After the request has finished running, the response body will contain the customer order history.

## Access the Retail Server APIs by using a console application

1. Use Visual Studio 2017 to create a console application.
2. In the `app.config` file, include the following configuration.

```
<appSettings>
  <add key="aadClientId" value="client id generated during app registration in Azure" />
  <add key="aadClientSecret" value="client secret generated during app registration in Azure" />
  <add key="aadAuthority" value="https://sts.windows.net/tenant id/" />
  <add key="retailServerUrl" value="https://RetailserverURL/Commerce" />
  <add key="resource" value="https://REtailServerURL" />
  <add key="operatingUnitNumber" value="OUN value" />
</appSettings>
```

#### Update the configuration settings with actual values

1. Use the NuGet package manager for the project to add the following NuGet packages.
  - Microsoft.IdentityModel.Clients.ActiveDirectory
  - Microsoft.Dynamics.Commerce.RetailProxy

The `Microsoft.Dynamics.Commerce.RetailProxy` NuGet package can be added from the `RetailSDK\pkgs` folder. In the NuGet manager, add a local repository for the `RetailSDK\pkgs` folder.

2. In the `Program.cs` file, add the following variables.

```
private static string clientId;
private static string clientSecret;
private static Uri retailServerUrl;
private static string resource;
private static string operatingUnitNumber;
private static Uri authority;
```

3. In the **Program.cs** file, add the **GetConfiguration** method to read the app settings.

```
private static void GetConfiguration()
{
    clientId = ConfigurationManager.AppSettings["aadClientId"];
    clientSecret = ConfigurationManager.AppSettings["aadClientSecret"];
    authority = new Uri(ConfigurationManager.AppSettings["aadAuthority"]);
    retailServerUrl = new Uri(ConfigurationManager.AppSettings["retailServerUrl"]);
    operatingUnitNumber = ConfigurationManager.AppSettings["operatingUnitNumber"];
    resource = ConfigurationManager.AppSettings["resource"];
}
```

4. In the **Program.cs** file, add the **CreateManagerFactory** method to get the access token.

```
private static async Task<ManagerFactory> CreateManagerFactory()
{
    Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationContext authenticationContext = new
    Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationContext(authority.ToString(), false);
    AuthenticationResult authResult = null;
    authResult = await authenticationContext.AcquireTokenAsync(resource, new
    ClientCredential(clientId, clientSecret));

    ClientCredentialsToken clientCredentialsToken = new
    ClientCredentialsToken(authResult.AccessToken);
    RetailServerContext retailServerContext = RetailServerContext.Create(retailServerUrl,
    operatingUnitNumber, clientCredentialsToken);
    ManagerFactory factory = ManagerFactory.Create(retailServerContext);
    return factory;
}
```

5. Add the **GetOrderHistory** method to read the orders.

```
private static async Task<Microsoft.Dynamics.Commerce.RetailProxy.PagedResult<SalesOrder>>
GetOrderHistory(string customerId)
{
    QueryResultSettings querySettings = new QueryResultSettings
    {
        Paging = new PagingInfo() { Top = 10, Skip = 10 }
    };

    ManagerFactory managerFactory = await CreateManagerFactory();
    ICustomerManager customerManage = managerFactory.GetManager<ICustomerManager>();
    return await customerManage.GetOrderHistory(customerId, querySettings);
}
```

6. In the **main** method, call the **GetOrderHistory** method to read the records.

```
static void Main(string[] args)
{
    GetConfiguration();
    Microsoft.Dynamics.Commerce.RetailProxy.PagedResult<SalesOrder> orderHistory = Task.Run(async ()
=> await GetOrderHistory("2001")).Result;
    Console.WriteLine(orderHistory.FirstOrDefault<SalesOrder>().Id);
}
```

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

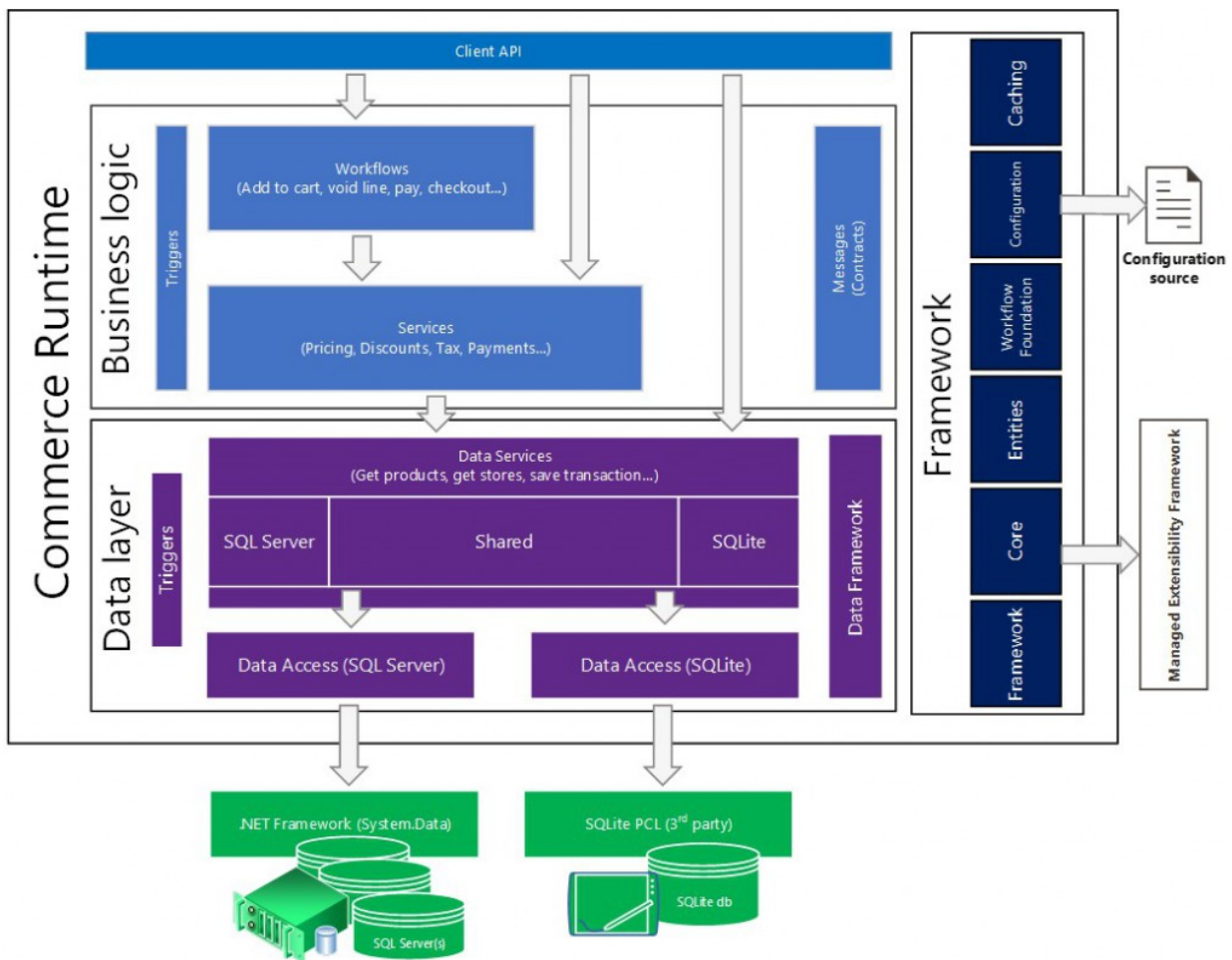
# Commerce runtime (CRT) architecture and configuration

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides information about the architecture and configuration of Commerce Runtime (CRT). The CRT is a collection of portable .NET libraries that encapsulate business logic. It serves as the engine for the commerce channel.

## Commerce Runtime architecture

The following diagram shows the components of the Microsoft Dynamics 365 Commerce Runtime (CRT).



### Data access

On top of the database is a data access layer. In the data access layer, raw data is translated into objects in memory. For example, an object might be a product. Products have attributes, such as price and color. The data access layer has functions that you can use to manipulate the objects. Stored procedures pass packets of data from the database to data entities that can be used in services and workflows. You can update the packets of data to include new fields that you add in Commerce.

### Services

On top of the data access layer is a services layer. Services query for real-time data. You can use these services to customize existing functionality, or you can add your own services that include new functionality.

### Workflows

On top of the services layer is the workflow layer. A workflow is a collection of services and business logic that, together, define business processes. For example, when a customer adds an item to the cart, you can use a workflow to get the price, perform validation, check the inventory quantity, calculate shipping charges, calculate tax, and calculate discounts. You can use the workflows that are included in Commerce, or you can create new workflows. You can even use a workflow to connect to a third-party system as part of your business processes.

## API

On top of the workflow layer is the application programming interface (API) layer. You can use the API for tasks such as getting information about items, calculating prices, calculating shipping charges, and placing orders. You can extend the API to fit your business processes.

## Commerce Runtime configuration

Services are enumerated as types in the CRT configuration file. You can add types in the CRT configuration file to control which services are loaded in the CRT. Services are loaded in the order in which they are listed in the configuration file. All the default services are loaded automatically. However, if you add a new service above one of the default services, the new service replaces the default service.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Commerce runtime (CRT) services

2/18/2021 • 12 minutes to read • [Edit Online](#)

Commerce runtime (CRT) is a collection of portable .NET libraries that contain the core business logic for the commerce channel and pricing functionality. To add or modify any business logic, you should customize CRT. Retail Modern POS or Cloud POS calls CRT to request that it perform business logic. CRT processes the request and then sends the response back to point of sale. POS is like a thin client, all the business logic should be done in CRT.

A CRT service is a group of requests/responses. Any time that you do something in POS, POS sends a request to Commerce Scale Unit, and Commerce Scale Unit calls CRT. CRT processes the request and sends back the response.

This topic shows some important requests/responses that you can customize for your business scenario.

There are three main layers in CRT:

- Services
- Workflow
- Data Access

All CRT customization for business logic can be done in Services, Workflow or Data Access layers, other core layers that are required for CRT to work are runtime, authentication, and data access managers and you should avoid any customization on these layers.

## NOTE

Detailed information about all CRT classes can be found in the Retail SDK, which is found at RetailSDK\Documents\CommerceRuntimeMessages.chm.

## Overall flow

The overall flow looks like this:

CRT service request < > Zero or more workflow requests < > Zero or more data access requests

For example, multiple workflow and data access requests are called to perform the **Create sales order** service request.

## CRT services

Each CRT service contains one or more requests/responses. For example, the Customer service in CRT contains all the customer-related requests/responses. Each request/response is run in a different flow.

### Default CRT services

Many services in CRT support the functionality of the channel and store operations. You can add your own services or extend the existing services. The following table describes some of the services that you might customize for various business services.

For more information about each service, see the CRT request/response document in the Retail software development kit (SDK), at ...RetailSDK\Code\Documents\CommerceRuntimeMessages.chm.

SERVICE	DESCRIPTION
AddressService	This service verifies addresses and gets location information, such as cities, counties, or states.
BarcodeService	This service processes the barcode that was scanned, based on the mask and barcode types, and calculates the quantity and price from the barcode.
CartService	This service gets the cart from the transaction and sales transaction service from the transaction tables.
ChargeService	This service implements logic that calculates automatic charges, price charges, and shipping charges for transactions.
CouponService	This service validates and updates coupon-related requests.
CurrencyService	This service converts currencies, based on exchange rates.
CustomerService	This service contains customer related operations such as Save customer, purchase history, get customer and customer balance.
EmployeeService	This service gets employee-related information and employees by store.
FormattingService	This service implements logic for the format of numbers, currencies, and dates.
GiftCardService	This service provides information about internal activities that are related to gift cards, such as issuing the gift card, getting the balance, and adding value.
LoyaltyService	This service implements a program that rewards repeat customers.
NotificationService	This service maintains the POS notification service.
PaymentService	This service lets you connect your online store to a payment service to provide credit card authorization and use preconfigured payment processing. You can also extend the payment service to add additional third-party payment processors.
ProductService	This service gets product-related and variant-related information.
ProductsService	This service gets information that is related to products and variants.
PricingService	This service gets the price of an item in real time. The price is adjusted, based on the base price and any applicable discounts. Discounts can be customized for each retailer.

SERVICE	DESCRIPTION
ProductAvailabilityService	This service calculates the quantities of products that are available to sell.
ReasonCodeService	This service calculates and gets the required reason code for POS operations or any workflow.
ReceiptService	This service gets and formats the receipt details.
RoundingService	This service rounds the tender amount, based on the tender type and store.
SalesOrderService	This service creates a sales order, based on a customer shopping cart.
SearchProductsService	This service searches products, based on the input text.
ShippingService	This service calculates shipping costs and determines shipping options for the current order.
StockCountService	This service creates, commits, and synchronizes stock journals.
StoreOperationService	This service maintains store-related operation services, such as Save and Drop, Tender declaration, and Search journal.
TaxService	This service calculates the sales tax for the current order. You can use sales tax information provided, or from a third-party sales tax service.
TotalingService	This service calculates the totals on the sales transactions and sales lines.

For extension scenarios, you can add CRT triggers, create new services, and override any of the requests in the service class. For information about how to extend and understand CRT extension patterns, see [Commerce runtime \(CRT\) extensibility](#).

#### NOTE

CRT extension code should not refer to or use any of the CRT business logic classes, methods, or handlers (such as classes from `Runtime.Workflow`, `Runtime.Services`, or `Runtime.DataServices`). These classes are not backward compatible, which could break extensions during an upgrade. Extensions should only use request, response, and entity classes from `Runtime.*.Messages`, `Runtime.Framework`, `Runtime.Data`, and `Runtime.Entities`.

## AddressService

The Address service supports the following requests/responses for various extension scenarios.

REQUEST	PURPOSE
GetCountryRegionsServiceRequest	This request gets the list of countries and regions that are supported.

REQUEST	PURPOSE
GetStateProvincesServiceRequest	This request gets the list of states or provinces that are supported for the country or region.
GetCitiesServiceRequest	This request gets the list of cities that are supported for the state or region.
GetDistrictServiceRequest	This request gets the list of districts that are supported.
GetZipCodesServiceRequest	This request gets the list of ZIP Codes that are supported.
GetFromZipPostalCodeServiceRequest	This request gets the list of postal codes supported.
GetAddressFormattingServiceRequest	This request gets the address format for the specified country or region.

### BarcodeService

REQUEST	PURPOSE
ProcessMaskSegmentsServiceRequest	This request processes the barcode, based on the barcode mask configuration.
GetBarcodeTypeServiceRequest	This request gets the types of barcodes that are supported.
CalculateQuantityFromPriceServiceRequest	This request calculates the price and quantity, based on the barcode that is scanned or entered.

### CartService

REQUEST	PURPOSE
GetSalesTransactionsServiceRequest	This request gets the sales transaction from Headquarters.
GetCartServiceRequest	This request gets the cart from the sales transaction table by using the cart ID.
CalculateSalesTransactionServiceRequest	This request calculates the various sales transaction totals, based on the specified calculation mode.
CalculateEstimatedShippingAuthorizationAmountServiceRequest	This request calculates the estimated shipping authorization amount on the transaction.
ConvertSalesTransactionToCartServiceRequest	This request converts the sales transaction to a cart transaction, based on transaction ID that is passed.

### CouponService

REQUEST	PURPOSE
UpdateCouponCodesOnCartServiceRequest	This request updates the coupon codes status in Headquarters, based on the coupons that are used in the cart.

REQUEST	PURPOSE
ValidateCouponCodesServiceRequest	This request validates the coupon code that is entered in the transactions.
UpdateCouponUsageServiceRequest	This request updates coupon usage for the transaction.

### CustomerService

REQUEST	PURPOSE
SaveCustomerServiceRequest	This request is called when you save a customer from the POS.
GetCustomersServiceRequest	This request gets the selected customer details.
CustomersSearchServiceRequest	This request is run when you search for a customer from the POS.
GetCustomerGroupsServiceRequest	This request gets the customer group details.
InitiateLinkToExistingCustomerServiceRequest	This request is an internal request for backward compatibility.
FinalizeLinkToExistingCustomerServiceRequest	This request is an internal request for backward compatibility.
UnlinkFromExistingCustomerServiceRequest	This request is an internal request for backward compatibility.
GetCustomerBalanceServiceRequest	This request gets the balance of the customer's account.
GetOrderHistoryServiceRequest	This request gets the customer's order history.
GetPurchaseHistoryServiceRequest	This request gets the history of the customer's recent purchases.
CustomerSearchByFieldsServiceRequest	This request is run when you search customer by using fields such as name, phone number etc. (hint search).
GetCustomerSearchFieldsServiceRequest	This request gets the list of customer search fields (hint fields).

### PricingService

REQUEST	PURPOSE
CalculatePricesServiceRequest	This request calculates the price for each item that is added to the cart and on the search screen.
CalculateDiscountsServiceRequest	This request calculates the discount for each item that is added to the cart.
GetIndependentPriceDiscountServiceRequest	This request calculates the price for each item on the price check screen and in other non-transaction-related flows.

REQUEST	PURPOSE
ValidateDiscountsServiceRequest	This request validates the discount that is entered, based on the employee.
GetAllPeriodicDiscountsServiceRequest	This request gets all the periodic discounts that are configured.
GetDiscountCodesServiceRequest	This request gets all the discount codes that are configured.

### ProductService

REQUEST	PURPOSE
ProductSearchServiceRequest	This request gets the product list, based on the search text from the POS. (This is an old request. For customization, use the new <b>SearchProductsServiceRequest</b> request instead.)
GetProductServiceRequest	This request gets the product, based on the product ID.
GetProductRefinersRequest	This request retrieves the product refiner values.

### ProductsService

REQUEST	PURPOSE
GetProductsServiceRequest	This request gets the products, based on the products IDs that are provided.
GetVariantProductsServiceRequest	This request gets specific variations of master type products.

### ReasonCodeService

REQUEST	PURPOSE
GetReasonCodesServiceRequest	This request gets the required reason code for the workflow or POS operation.
CalculateRequiredReasonCodesServiceRequest	This request calculates the required reason code for the workflow or POS operation.
CalculateCartRequiredReasonCodesServiceRequest	This request calculates the required reason code for the cart workflow.
CalculateDropAndDeclareTransactionRequiredReasonCodesServiceRequest	This request calculates the required reason code for the drop and tender declaration transaction.
CalculateNonSalesTransactionRequiredReasonCodesServiceRequest	This request calculates the required reason code for the non-sales transaction, such as tender declaration.
GetReturnOrderReasonCodesServiceRequest	This request gets the required reason code for the return transaction.
ValidateReasonCodeLineForUpdateServiceRequest	This request validates the reason code lines that are added during the save cart request.

## ReceiptService

REQUEST	PURPOSE
GetReceiptServiceRequest	This request gets the receipt type and generates the receipt data, based on the receipt type.
GetCustomReceiptFieldServiceRequest	Override this request, and add custom logic for your custom fields that are added in the receipt.
GetEmailReceiptServiceRequest	This request gets the formatted receipts that will be used for print and email scenarios.
PopulateTaxSummaryIndiaServiceRequest	This request populates the tax summary for India transaction.

## SearchProductsService

REQUEST	PURPOSE
SearchProductsServiceRequest	This request is run when you search for a product from the POS.

## TaxService

REQUEST	PURPOSE
CalculateTaxServiceRequest	This request calculates taxes on the transaction.
AssignTaxCodesServiceRequest	This request assigns tax codes on the transaction's taxable items before the tax calculation. The default tax calculation handler of the <b>CalculateTaxServiceRequest</b> request uses this message.
PopulateTaxRatesRequest	This request populates tax rates for the recalled transaction. No tax percentage information is persisted for tax lines. This message handler restores the information.

## Workflow layer

On top of the Services layer is the Workflow layer. A workflow is a collection of services and business logic that together define business processes. For example, when a customer adds an item to the cart, you can use a workflow to get the price, do validation, check the inventory quantity, calculate shipping, calculate tax, and calculate discounts. You can customize the existing workflows, or you can create new workflows. You can even use a workflow to connect to a third-party system as part of your business processes.

Just like services, workflows use the request/response pattern. The request object inherited from the base CRT [Request](#) class. The response object inherited from the base CRT [Response](#) class. A workflow also has a request handler class that extends the [WorkflowRequestHandler<TRequest, TResponse>](#) class. To create a workflow, you create a request class and a response class, and you then create a request handler class that contains the business logic for the workflow.

For example, when you create a cash-and-carry transaction or a customer order, many different steps or workflows are completed before the order is created. One of the workflow steps in the order process is the Save cart request. The Save cart request workflow is responsible for saving any changes that are made to the cart from the POS. For example, when you add an item to the cart, change the quantity, and so on, anything that you

do in the POS will cause a call to the SaveCart to save the changes to the POS and database.

### Default workflows and handlers

The following table lists the default workflow requests and response. CRT services call the workflows request and response based on the operation you perform in POS. You can customize any of these workflows request and response according to your business scenario.

REQUEST	HANDLER	PURPOSE
AddOrderInvoiceToCartRequest	AddOrderInvoiceToCartRequestHandler	This request adds the invoice to the cart as a cart line.
AddOrRemoveDiscountCodesRequest	AddOrRemoveDiscountCodesRequestHandler	This request adds discount codes to the cart or removes them from the cart.
CancelOrderRequest	CancelOrderRequestHandler	This request is triggered when you cancel an order from the POS.
CopyCartRequest	CopyCartRequestHandler	This request creates an identical shopping cart that has the specified cart type.
GetAddressRequest	GetAddressRequestHandler	This request gets the address from the list.
GetCardPaymentAcceptPointRequest	GetCardPaymentAcceptPointRequestHandler	This request shows the accepting page for card payment.
GetCartRequest	GetCartRequestHandler	This request gets the shopping cart, based on the cart search criteria.
GetDiscountCodesRequest	GetDiscountCodesRequestHandler	This request gets the discount codes.
GetOrdersRequest	GetOrdersRequestHandler	This request gets sales orders.
GetPromotionsRequest	GetPromotionsRequestHandler	This request gets the shopping cart together with promotion lines for the cart and the cart items.
GetScanResultRequest	GetScanResultRequestHandler	This request gets the entity details, based on the details that are scanned or typed.
GetSupportedCardTypesRequest	GetSupportedCardTypesRequestHandler	This request retrieves the card types that are supported for the specified currency.
IssueOrAddToGiftCardRequest	IssueOrAddToGiftCardRequestHandler	This request issues a gift card or adds to a gift card's balance.
PickAndPackOrderRequest	PickAndPackOrderRequestHandler	This request represents a request for picking list creation and packing slip creation for a customer order.



REQUEST	HANDLER	PURPOSE
PickupAtStoreRequest	PickupAtStoreRequestHandler	This request represents a request for pick-up at the store.
RecalculateOrderRequest	RecalculateOrderRequestHandler	This request represents a request for the recalculate order workflow.
RecallCustomerOrderRequest	RecallCustomerOrderRequestHandler	This request gets sales orders and converts them into a cart.
RecallSalesInvoiceRequest	RecallSalesInvoiceRequestHandler	This request gets invoices.
ResumeCartRequest	ResumeCartRequestHandler	This request represents a request to resume a cart that was suspended.
SaveCartLinesRequest	SaveCartLinesRequestHandler	This request represents a request for create, update, delete, or void operations on the cart line.
SaveCartRequest	SaveCartRequestHandler	This request is triggered when you make any changes in that POS that affect the cart.
SaveCustomerOrderRequest	SaveCustomerOrderRequestHandler	This request initiates a request to save the customer order, based on the specified cart identifier and payment card information.
SaveReasonCodeLineRequest	SaveReasonCodeLineRequestHandler	This request adds or updates a reason code on the cart line.
SaveTenderLineRequest	SaveTenderLineRequestHandler	This request adds, removes, or updates a tender line for the given shopping cart.
SaveVoidTransactionRequest	SaveVoidTransactionRequestHandler	This request voids a transaction.
SubmitSalesTransactionRequest	SubmitOrderRequestHandler	This request initiates a request to submit the sales transaction, based on the specified cart identifier.
SuspendCartRequest	SuspendCartRequestHandler	This request represents a request to suspend the cart.
TransferCartRequest	TransferCartRequestHandler	This request transfers the specified shopping cart.
UpdateCommissionSalesGroupRequest	UpdateCommissionSalesGroupHandler	This request encapsulates a request for updating the sales representative on the cart or the cart line.
UploadOrderRequest	UploadOrderRequestHandler	This request uploads the sales order.
ValidateCartForCheckoutRequest	ValidateCartForCheckoutRequestHandler	This request validates the cart for checkout.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Commerce runtime (CRT) extensibility

2/18/2021 • 22 minutes to read • [Edit Online](#)

This topic describes various ways that you can extend the commerce runtime (CRT). It explains the concept of extension properties, and shows how to add them to a CRT entity so that they are persistent and so that they aren't persistent.

CRT contains the core business logic. If you want to add or modify any business logic, you should customize CRT. All the CRT code is developed by using C#, and then it's compiled and released as class libraries (.NET assemblies). Point of sale (POS) is a thin client. All the business logic is done in CRT. POS calls CRT to perform any business logic, and then CRT processes the information and sends it back to POS.

Every CRT service consists of a group of one or more requests and responses. POS sends a request to Retail Server, and Retail Server calls CRT. CRT then processes the request and sends back the response.

For example, the Product service in CRT contains all the product-related requests and responses, each of which is run in a different flow. Likewise, the Customer service in CRT contain all the customer-related requests and responses. The following table shows the requests in the Customer service.

REQUEST	PURPOSE
SaveCustomerServiceRequest	This request is called when you save a customer from POS.
GetCustomersServiceRequest	This request is used to get the details of the selected customer.
CustomersSearchServiceRequest	This request is run when you do a customer search from POS.
GetCustomerGroupsServiceRequest	This request is used to get the details of the customer group.
InitiateLinkToExistingCustomerServiceRequest	This internal request is for backward compatibility.
FinalizeLinkToExistingCustomerServiceRequest	This internal request is for backward compatibility.
UnlinkFromExistingCustomerServiceRequest	This internal request is for backward compatibility.
GetCustomerBalanceServiceRequest	This request gets the balance for the customer account.
GetOrderHistoryServiceRequest	This request gets the customer's order history.
GetPurchaseHistoryServiceRequest	This request gets the history of the customer's recent purchases.
CustomerSearchByFieldsServiceRequest	This request is run when you search for customers by using fields (hint search).
GetCustomerSearchFieldsServiceRequest	This request gets the list of customer search fields (hint fields).

# CRT extension patterns

Before you learn about the CRT extension patterns, you should understand how a CRT extension can be created. CRT is just a collection of C# class libraries (.NET assemblies). You can create a class library project in C# and do all the CRT extension by using the patterns that are shown in the following subsections. Always use the samples that Microsoft provides as templates for your extension, because these samples have the correct assembly references, Microsoft .NET Framework version, output type, and build parameters. Additionally, all the other required parameters are preconfigured. You can find the CRT sample extension in the Retail software development kit (SDK), at ...\\RetailSDK\\SampleExtensions\\CommerceRuntime.

## NOTE

As of version 10.0.16 of Finance and Operations apps, all class libraries for CRT extension projects must use .NET Standard 2.0 for the target framework.

## Create a new CRT service

You can create new functionality or a new feature.

## Override the existing service

You can completely override existing functionality or customize it according to your business flow. Here are some examples:

- You want to override the POS search functionality to search from an external system instead of searching in a local database or Commerce headquarters. Alternatively, you can do an override, call the standard functionality, and do some additional custom logic.
- Search for a customer in a local database or Commerce headquarters, search for the customer in an external system, and then merge or modify the results.

Avoid overriding the handler. You can implement most of the CRT extension scenarios by using pre-triggers or post-triggers. Overrides are required only when you want to completely replace the existing functionality.

## CRT data service and data service with entities

You can use the CRT data service to read data or entities from the channel database.

## NOTE

CRT extension code should not refer to or use any of the CRT business logic classes, methods, or handlers (such as classes from `Runtime.Workflow`, `Runtime.Services`, or `Runtime.DataServices`). Because these classes aren't backward compatible, extensions could be broken during an upgrade. Extensions should use only request, response, and entity classes from `Runtime.*.Messages`, `Runtime.Framework`, `Runtime.Data`, and `Runtime.Entities`.

## Triggers

You can run additional logic before or after any request.

In the pre-trigger, you can do some validation, custom logic, and so on.

In the post-trigger, you can add some custom information to the request and send it to POS. Alternatively, you can modify the result that is returned from the standard functionality or do some additional business logic.

For information about how to create CRT trigger extensions, see [Commerce runtime \(CRT\) triggers extension](#).

## Extension properties

You can add custom properties to any CRT entity and send it to POS. Extension properties are key-value pairs. If you set an extension property in POS, it will be available in CRT. Likewise, if you set an extension property in CRT, it will be available in POS. You can also set extension properties at the level of the request, the response, or the

request context. By default, extension properties aren't stored in and read from the database. To read or set extension properties, you must write custom code. However, that custom code is automatically sent between POS and CRT.

For example, you want to capture and show some additional information to the Customer entity in POS. In this case, you can add a post-trigger to fetch all your custom properties for customers, add them to the Customer entity as extension properties, and then send those extension properties to POS.

You can also send extension properties from POS to CRT and store them in your custom table. Alternatively, you can do some custom logic based on those properties, or send it to Commerce headquarters.

All CRT entities, such as products, customers, transactions, and parameters, support extension properties.

#### NOTE

Attributes are also supported (configuration-driven development). For extension properties, you must create a custom table and store the data. However, attributes are configuration driven and aren't required to create table fields. Therefore, no code is required for read and update operations.

For details about the attributes, see the following topics:

- [Order attributes](#)
- [Customer attributes](#)

#### Extend Commerce Data Exchange - Real-time Service classes

You can do synchronous call from CRT to Commerce headquarters.

For information about how to extend Commerce Data Exchange - Real-time service, see [Extend Commerce Data Exchange - Real-time Service](#).

## Retail Server extension

For information about how to create new Retail Server APIs, see [Create a new Retail Server extension API](#).

## Register the CRT extension

### Online

For online mode (that is, when POS is connected to Retail Server), after you've finished extending CRT, put the extension library in the `\RetailServer\webroot\bin\Ext` folder. In the `CommerceRuntime.Ext.config` file, update the **composition** section with the information about the custom library, as shown in the following example.

```
<add source="assembly" value="your custom library name" />
```

For example, if the name of your custom library is `Contoso.Commerce.Runtime.CustomerSearchSample`, add the following line in the **composition** section.

```
<add source="assembly" value="Contoso.Commerce.Runtime.CustomerSearchSample" />
```

### Offline

For offline mode, after you've finished extending CRT, put the custom library in the `\Program Files (x86)\Microsoft Dynamics 365\70\Retail Modern POS\ClientBroker\ext` folder. In the `CommerceRuntime.MPOSOOffline.ext.config` file, update the **composition** section with the information

about the custom library, as shown in the following example.

```
<add source="assembly" value="your custom library name" />.
```

## Register the extension configuration in the CommerceRuntime.Ext.config file

Your extension can register key-value configurations in the **CommerceRuntime.Ext.config** file in the **<settings>** node.

The **<settings>** node is a collection of key-value pairs that is used to configure the CommerceRuntime components. The convention is to prefix the keys for settings with a representation of the functional area that the keys are configuring, and to use a period (.) as the separator between prefixes and keys. Extension configuration values must be prefixed with **ext**, because the CRT initialization enforces this convention. Any extension that doesn't have that prefix won't be loaded. More prefixes can be added to represent the subarea that the keys are configuring. The following example shows a key-value pair.

```
<add name="ext.SalesTransaction.Storage.CartSerializationFormat" value="XML" />
```

For HTTP binding time-outs on calls to Commerce Data Exchange - Real-time Service, configure the time-out in seconds per method. This time-out value is limited by the **maxExtensionTimeoutInSeconds** value, which is set in the **CommerceRuntime.config** file.

```
<add  
name="ext.RealTimeServiceClient.TimeoutInSeconds.InvokeExtensionMethod.ContosoRetailStoreHours_UpdateStoreHours" value="300" />
```

To read the key value from the CRT extension configuration, use **RequestContext.Runtime.Configuration**. The following example shows how to retrieve a value.

```
string key =  
context.Runtime.Configuration.GetSettingValue("ext.SalesTransaction.Storage.CartSerializationFormat") ??  
string.Empty;
```

The preceding steps are used for manual deployment and testing on your development box. To package the extension and deploy it to production or user acceptance testing (UAT), use the information in the packaging document.

## Debug CRT

### Attach the CRT extension project online

To debug CRT from POS, attach the CRT extension project to the **w3wp.exe** process (the Internet Information Services [IIS] process for Retail Server) when POS is connected to Retail Server.

### Attach the CRT extension project offline

For offline debugging, attach the CRT extension project to the **dllhost.exe** process.

## NOTE

CRT extension code should not refer to or use any of the CRT business logic classes, methods, or handlers (such as classes from `Runtime.Workflow`, `Runtime.Services`, or `Runtime.DataServices`). Because these classes aren't backward compatible, extensions could be broken during an upgrade. Extensions should use only request, response, and entity classes from `Runtime.*.Messages`, `Runtime.Framework`, `Runtime.Data`, and `Runtime.Entities`.

## Sample to create a new CRT service class

You can create a new service class, and implement one or more requests and responses. Use this approach to create a new feature.

Implement the following classes for a new CRT service:

- **Request class** – This class is the POS, Retail Server, e-commerce, or CRT workflow class that is the request to do something.
- **Response class** – This class returns the response, based on the request to the caller.
- **Handler class** – This class contains the core logic for the request. In the handler class, you can call other requests, do custom logic, and perform other tasks.

For serialization to work, the new request type must implement the `[DataContract]` and `[DataMember]` attributes.

### Request class

```
using System.Runtime.Serialization;
using Microsoft.Dynamics.Commerce.Runtime.Messages;

[DataContract]
public sealed class GetStoreHoursDataRequest : Request
{
    public GetStoreHoursDataRequest(string storeNumber)
    {
        this.StoreNumber = storeNumber;
    }

    [DataMember]
    public string StoreNumber { get; private set; }
}
```

### Response class

The new response type resembles the request type.

```
[DataContract]
public sealed class GetStoreHoursDataResponse : Response
{
    public GetStoreHoursDataResponse(PagedResult dayHours)
    {
        this.DayHours = dayHours;
    }

    [DataMember]
    public PagedResult DayHours { get; private set; }
}
```

Next, you must create a new CRT service that uses the request and response types.

# Create a new CRT service class

1. Implement the new service.

```
public class StoreHoursDataService : IRequestHandlerAsync
```

2. Implement two members of the interface. The **SupportedRequestTypes** member returns a list of all requests that this service can handle. The **execute** method is the method that CRT calls if a request for this service is run.

```
public IEnumerable<Type> SupportedRequestTypes
{
    get
    {
        return new[]
        {
            typeof(GetStoreHoursDataRequest),
            typeof(UpdateStoreDayHoursDataRequest),
        };
    }
}

public async Task<Response> Execute(Request request)
{
    if (request == null)
    {
        throw new ArgumentNullException("request");
    }

    Type reqType = request.GetType();
    if (reqType == typeof(GetStoreHoursDataRequest))
    {
        return await
this.GetStoreDayHoursAsync((GetStoreHoursDataRequest)request).ConfigureAwait(false);
    }
    else if (reqType == typeof(UpdateStoreDayHoursDataRequest))
    {
        return await
this.UpdateStoreDayHoursAsync((UpdateStoreDayHoursDataRequest)request).ConfigureAwait(false);
    }
    else
    {
        string message = string.Format(CultureInfo.InvariantCulture, "Request '{0}' is not
supported.", reqType);
        Console.WriteLine(message);
        throw new NotSupportedException(message);
    }
}

private async Task<Response> GetStoreDayHoursAsync(GetStoreHoursDataRequest request)
{
    ThrowIf.Null(request, "request");

    using (DatabaseContext databaseContext = new DatabaseContext(request.RequestContext))
    {
        var query = new SqlPagedQuery(request.QueryResultSettings)
        {
            DatabaseSchema = "ext",
            Select = new ColumnSet("DAY", "OPENTIME", "CLOSINGTIME", "RECID"),
            From = "CONTOSORETAILSTOREHOURSVIEW",
            Where = "STORENUMBER = @storeNumber",
        };

        query.Parameters["@storeNumber"] = request.StoreNumber;
        return new GetStoreHoursDataResponse(await
```



```

databaseContext.ReadEntityAsync<DataModel.StoreDayHours>(query).ConfigureAwait(false));
    }
}

private async Task<Response> UpdateStoreDayHoursAsync(UpdateStoreDayHoursDataRequest request)
{
    ThrowIf.Null(request, "request");
    ThrowIf.Null(request.StoreDayHours, "request.StoreDayHours");
    if (request.StoreDayHours.DayOfWeek < 1 || request.StoreDayHours.DayOfWeek > 7)
    {
        throw new
DataValidationException(DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_ValueOutOfRange);
    }

    InvokeExtensionMethodRealtimeRequest extensionRequest = new
InvokeExtensionMethodRealtimeRequest(
        "ContosoRetailStoreHours_UpdateStoreHours",
        request.StoreDayHours.Id,
        request.StoreDayHours.DayOfWeek,
        request.StoreDayHours.OpenTime,
        request.StoreDayHours.CloseTime);
    InvokeExtensionMethodRealtimeResponse response = await
request.RequestContext.ExecuteAsync<InvokeExtensionMethodRealtimeResponse>
(extensionRequest).ConfigureAwait(false);
    ReadOnlyCollection<object> results = response.Result;

    long recId = Convert.ToInt64(results[0]);

    using (var databaseContext = new DatabaseContext(request.RequestContext))
    {
        ParameterSet parameters = new ParameterSet();
        parameters["@bi_Id"] = recId;
        parameters["@i_Day"] = request.StoreDayHours.DayOfWeek;
        parameters["@i_OpenTime"] = request.StoreDayHours.OpenTime;
        parameters["@i_ClosingTime"] = request.StoreDayHours.CloseTime;
        await databaseContext.ExecuteStoredProcedureNonQueryAsync("
[ext].UPDATESSTOREDAYHOURS", parameters, resultSettings: null).ConfigureAwait(false);
    }

    return new UpdateStoreDayHoursDataResponse(request.StoreDayHours);
}

```

3. Register the CRT extension as described earlier in this topic.

The preceding sample code is missing the implementation of **UpdateStoreDayHoursDataRequest** and **UpdateStoreDayHoursDataResponse**. The full sample code is available in the Retail SDK, at **RetailSDK\SampleExtensions\CommerceRuntime\Extensions.StoreHoursSample**.

## Implement a new CRT service that handles a single new request

It's slightly easier to create a single-request service.

```

namespace Contoso
{
    namespace Commerce.Runtime.CrossLoyaltySample
    {
        using System;
        using System.Threading.Tasks;
        using Messages;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;

        /// <summary>
        /// Service class responsible executing the service requests.
        /// </summary>
        public class CrossLoyaltyCardService : SingleAsyncRequestHandler<GetCrossLoyaltyCardRequest>
        {
            /// <summary>
            /// Process method.
            /// </summary>
            /// <param name="request">The request with the loyalty number.</param>
            /// <returns>The discount value.</returns>
            protected override async Task<Response> Process(GetCrossLoyaltyCardRequest request)
            {
                if (request == null)
                {
                    throw new ArgumentNullException("request");
                }

                var serviceResponse = new GetCrossLoyaltyCardResponse(0);

                //Business logic

                return await Task.FromResult(serviceResponse);
            }
        }
    }
}

```

## Implement a CRT service that overrides the functionality of an existing request

In some cases, the request and response types are sufficient, but you must change the out-of-box service implementation logic to perform different logic, or you must integrate with an external service to perform that logic. An override of the default implementation must be done only when you want to completely replace the logic. For example, if you don't want to use the out-of-box tax implementation but want to use third-party tax logic instead, override the tax service request. Most other scenarios can be achieved by adding a pre-trigger or post-trigger to the request. Try to avoid overriding the request. When you override the request, the custom logic will be run, and you might not get the future enhancements that are done in this overridden request.

Additionally, registration in the **commerceRuntime.ext.Config** file must precede registration of the service that should be overridden. This registration order is important because of the way that the Managed Extensibility Framework (MEF) loads the extension dynamic-link libraries (DLLs). The types that are higher in the file take precedence.

To override any CRT request, follow the pattern in the following example that overrides the out-of-box **CreateOrUpdateCustomerDataRequest** request.

```

namespace Contoso
{
    namespace Commerce.Runtime.EmailPreferenceSample
    {
        using System.Threading.Tasks;
        using System.Transactions;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.Data;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        using Microsoft.Dynamics.Commerce.Runtime.DataServices.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;

        /// <summary>
        /// Create or update customer data request handler.
        /// </summary>
        public sealed class CreateOrUpdateCustomerDataRequestHandler :
            SingleAsyncRequestHandler<CreateOrUpdateCustomerDataRequest>
        {
            /// <summary>
            /// Executes the workflow to create or update a customer.
            /// </summary>
            /// <param name="request">The request.</param>
            /// <returns>The response.</returns>
            protected override async Task<Response> Process(CreateOrUpdateCustomerDataRequest request)
            {
                ThrowIf.Null(request, "request");

                return new SingleEntityDataServiceResponse<Customer>(customer);
            }
        }
    }
}

```

## Run the base handler in the extension

### NotHandledResponse

If the overridden logic runs the base handler for some scenarios, execution of the base handler can be achieved by returning **new NotHandledResponse()**. If **NotHandledResponse** is returned, the CRT framework will use the extension that is requesting to run the base or out-of-band logic, and will run the out-of-band handler.

**NotHandledResponse** can be used in scenarios where the extension runs the base handler logic. For example, if the overridden request runs the base handler logic, it can return **NotHandledResponse** for the base handler to run. Alternatively, if the extension runs custom logic and base logic, the extension code can return **NotHandledResponse** after it runs the custom logic.

```

private Response
GetCustomReceiptFieldForSalesTransactionReceipts(GetLocalizationCustomReceiptFieldServiceRequest request)
{
    ThrowIf.Null(request.SalesOrder, nameof(request.SalesOrder));

    string receiptFieldName = request.CustomReceiptField;
    string receiptFieldValue = string.Empty;

    if (request.SalesOrder.TaxCalculationType == TaxCalculationType.GTE)
    {
        switch (receiptFieldName)
        {
            case "Sample":
                receiptFieldValue = this.GetGstRegistrationNumber(request);
                break;
            default:
                return new NotHandledResponse();
        }
    }
    else
    {
        return new NotHandledResponse();
    }

    int receiptFieldLength = request.ReceiptItemInfo == null ? 0 : request.ReceiptItemInfo.Length;
    var returnValue = ReceiptStringUtils.WrapString(receiptFieldValue, receiptFieldLength);

    return new GetCustomReceiptFieldServiceResponse(returnValue);
}

```

## Run an extension request for a channel type

Sometimes, an extension request must be run only for a specific channel type. For example, the request might have to be run for the online channel but not for the retail channel (physical store). In these cases, before you run the request, check the channel type. Then run the custom logic, or run the base logic by calling **NotHandledResponse()**.

```

if (requestContext.GetChannel().OrgUnitType == RetailChannelType.RetailStore)
{
    // run your extension code here.
}
else
{
    return new NotHandledResponse();
}

```

## Implement a new CRT entity and use it in the new CRT service

Any new entity must inherit from the **CommerceEntity** type. When you use this type, lots of low-level functionality is automatically handled for you. The following example is taken from the **StoreHours** sample. It shows how to create an entity that is bound to the database table. This scenario is common.

```

public class StoreDayHours : CommerceEntity
{
    private const string DayColumn = "DAY";
    private const string OpenTimeColumn = "OPENTIME";
    private const string CloseTimeColumn = "CLOSINGTIME";
    private const string IdColumn = "RECID";

    public StoreDayHours()
        : base("StoreDayHours")
    {
    }

    [DataMember]
    [Column(DayColumn)]
    public int DayOfWeek
    {
        get { return (int)this[DayColumn]; }
        set { this[DayColumn] = value; }
    }

    [DataMember]
    [Column(OpenTimeColumn)]
    public int OpenTime
    {
        get { return (int)this[OpenTimeColumn]; }
        set { this[OpenTimeColumn] = value; }
    }

    [DataMember]
    [Column(CloseTimeColumn)]
    public int CloseTime
    {
        get { return (int)this[CloseTimeColumn]; }
        set { this[CloseTimeColumn] = value; }
    }

    [Key]
    [DataMember]
    [Column(IdColumn)]
    public long Id
    {
        get { return (long)this[IdColumn]; }
        set { this[IdColumn] = value; }
    }
}

```

When you want to use the new entity in a service, the process is straightforward. As was described earlier in this topic, you create a new service that is derived from **IRequestHandler**. You then either use or return the new entity. The following example shows how to read the entity from the database and return it as part of the response.

```

private async Task<Response> GetStoreDayHoursAsync(GetStoreHoursDataRequest request)
{
    ThrowIf.Null(request, "request");

    using (DatabaseContext databaseContext = new DatabaseContext(request.RequestContext))
    {
        var query = new SqlPagedQuery(request.QueryResultSettings)
        {
            DatabaseSchema = "ext",
            Select = new ColumnSet("DAY", "OPENTIME", "CLOSINGTIME", "RECID"),
            From = "CONTOSORETAILSTOREHOURSVIEW",
            Where = "STORENUMBER = @storeNumber",
        };

        query.Parameters["@storeNumber"] = request.StoreNumber;
        return new GetStoreHoursDataResponse(await databaseContext.ReadEntityAsync<DataModel.StoreDayHours>
(query).ConfigureAwait(false));
    }
}

```

#### NOTE

Commerce entities aren't thread safe. Therefore, the same object should not be read or modified when another thread is writing to it. This restriction applies to custom entities and out-of-box entities in the extension code. You should avoid having two different threads when you do synchronous read or write activities for the same shared object.

In the preceding example, the CRT runtime engine automatically makes a query to the channel database via the registered data adapter. It queries a type that has the name `crt.ISVRetailStoreHoursView`, and generates a **where** clause and columns as specified in the code. The customizer is responsible for providing the SQL objects as part of the customization.

## Using extension properties on CRT entities, requests, and responses

One way to add new data to an existing CRT entity is to use extension properties. Extension properties are key-value pairs on the entity. By default, these key-value pairs don't persist in the database. To make an extension property persist, you must write custom code.

## Using extension properties on CRT entities with persistence

Any extension property that you add to an entity stays in memory for POS and CRT for the lifetime of either the object or the transaction, depending on the scenario. The extension property also travels across application boundaries. For example, if you add an extension property in Retail Modern POS and then call Retail Server or CRT, the key-value pair is available during the whole flow. Additionally, if that entity is sent during a call to Commerce Data Exchange - Real-time Service, the key-value pair is available during the process.

#### NOTE

For Commerce headquarters, extension properties are sent only for customers and orders.

However, as was mentioned earlier, the extension property doesn't persist by default. If you want an extension property to persist, you must do data modeling to ensure that you make the correct design choices about where the data should reside. We recommend that you use a new table and a join. This approach fits most requirements well. The `EmailPreference` sample in the Retail SDK provides a good end-to-end example.

### Location of the sample code in the Retail SDK

...\RetailSDK\SampleExtensions\CommerceRuntime\Extensions.EmailPreferenceSample

## Location of the scripts

...\RetailSDK\Documents\SampleExtensionsInstructions\EmailPreference

## Syntax to set an extension property on an entity

```
public virtual void SetProperty(string key, object value);  
entity.SetProperty("key", value);
```

## Example

### Scenario

A new extension table and fields that were created for the Customer entity and extension must read those fields from the extension table and send them to POS.

When you extend the channel database, it's always a good idea to include the primary key from the main table in the extension table. That is, don't have a direct relation, and then read the data from the extension table by using the primary key.

### Steps

1. Create the extension table for the Customer entity in the channel database extension schema that has the primary key of the main table.
2. Identify the CRT data request that reads the Customer entity.
3. Add a post-trigger for the data request. You will then be able to use the primary key of the Customer table to query your extension table to read the custom fields.

### NOTE

You can get the primary key for the entity in the post-trigger request. That request will have the entity object, and that entity object will have all the required fields.

4. Add the custom fields as extension properties to the shared parameters entity in the CRT post-trigger, and send it to POS.

## Reading extension properties in triggers

The request that reads the data from the Customer table is `GetCustomerDataRequest`. The following example shows how you can add a post-trigger for this request.

```
namespace Contoso  
{  
    namespace Commerce.Runtime.EmailPreferenceSample  
    {  
        using System;  
        using System.Collections.Generic;  
        using System.Threading.Tasks;  
        using Microsoft.Dynamics.Commerce.Runtime;  
        using Microsoft.Dynamics.Commerce.Runtime.Data;  
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;  
        using Microsoft.Dynamics.Commerce.Runtime.DataServices.Messages;  
        using Microsoft.Dynamics.Commerce.Runtime.Messages;  
  
        /// <summary>  
        /// Class that implements a post trigger for the GetCustomerDataRequest request type.  
        /// </summary>  
        public class GetCustomerTriggers : IRequestTriggerAsync  
        {
```

```

    /// <summary>
    /// Gets the supported requests for this trigger.
    /// </summary>
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            return new[] { typeof(GetCustomerDataRequest) };
        }
    }

    /// <summary>
    /// Post trigger code to retrieve extension properties.
    /// </summary>
    /// <param name="request">The request.</param>
    /// <param name="response">The response.</param>
    public async Task OnExecuted(Request request, Response response)
    {
        ThrowIf.Null(request, "request");
        ThrowIf.Null(response, "response");

        var customer = ((SingleEntityDataServiceResponse<Customer>)response).Entity;

        if (customer == null)
        {
            return;
        }

        var query = new SqlPagedQuery(QueryResultSettings.SingleRecord)
        {
            DatabaseSchema = "ext",
            Select = new ColumnSet(new string[] { "EMAILOPTIN" }),
            From = "CUSTOMEREXTENSIONVIEW",
            Where = "ACCOUNTNUM = @accountNum AND DATAAREAID = @dataAreaId"
        };

        query.Parameters["@accountNum"] = customer.AccountNumber;
        query.Parameters["@dataAreaId"] =
request.RequestContext.GetChannelConfiguration().InventLocationDataAreaId;
        using (var databaseContext = new DatabaseContext(request.RequestContext))
        {
            var extensionsResponse = await databaseContext.ReadEntityAsync<ExtensionsEntity>
(request).ConfigureAwait(false);
            ExtensionsEntity extensions = extensionsResponse.FirstOrDefault();

            var emailOptIn = extensions != null ? extensions.GetProperty("EMAILOPTIN") : null;
            if (emailOptIn != null)
            {
                customer.SetProperty("EMAILOPTIN", emailOptIn);
            }
        }
    }

    /// <summary>
    /// Pre trigger code.
    /// </summary>
    /// <param name="request">The request.</param>
    public async Task OnExecuting(Request request)
    {
        // It's only stub to handle async signature.
        await Task.CompletedTask;
    }
}
}
}

```



#### NOTE

You must change this query, based on your new field and table, or whatever condition you're using. When you design the table, make sure that you have the primary key field from the parent table, so that you can query it by using the CRT entity. Most CRT entities should have the primary key field in them, such as the **RecId** field or another relevant unique field that is used to select the record.

## Saving extension properties by overriding the handler

### Scenario

You created a new extension table for Customer. When values for the extension field for Customer come from POS or CRT, you want to store the values in your custom table.

#### NOTE

You can set extension properties either in a trigger or by overriding the handler. If you just set some extension properties by reading from your extension table, you can use triggers.

### Steps

1. Create your extension table for Customer in the channel table that has the primary key relation to the main table. (In this case, the main table is CustTable.)
2. Identify the CRT data request that sets data in CustTable.
3. Override the handler, and call the base request to set values in the main table (CustTable) and then in the extension table.

#### NOTE

Extension code can pass additional properties that are required for the extension procedure or view. Extension properties can be saved in the CRT pre-triggers or post-triggers. You don't have to override the CRT handler. You should avoid overriding the handler, because most of the CRT extension scenarios can be achieved in pre-triggers or post-triggers.

The example that follows doesn't have the SQL scripts that are used. You can find the full sample code and scripts in the following locations in the Retail SDK.

### Location of the sample code in the Retail SDK

...\RetailSDK\SampleExtensions\CommerceRuntime\Extensions.EmailPreferenceSample

### Location of the scripts

...\RetailSDK\Documents\SampleExtensionsInstructions\EmailPreference

### Example

```

namespace Contoso
{
    namespace Commerce.Runtime.EmailPreferenceSample
    {
        using System.Threading.Tasks;
        using System.Transactions;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.Data;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        using Microsoft.Dynamics.Commerce.Runtime.DataServices.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;

        /// <summary>
        /// Create or update customer data request handler.
        /// </summary>
        public sealed class CreateOrUpdateCustomerDataRequestHandler :
        SingleAsyncRequestHandler<CreateOrUpdateCustomerDataRequest>
        {
            /// <summary>
            /// Executes the workflow to create or update a customer.
            /// </summary>
            /// <param name="request">The request.</param>
            /// <returns>The response.</returns>
            protected override async Task<Response> Process(CreateOrUpdateCustomerDataRequest request)
            {
                ThrowIf.Null(request, "request");

                using (var databaseContext = new DatabaseContext(request.RequestContext))
                using (var transactionScope = new TransactionScope())
                {
                    // Execute original functionality to save the customer.
                    var requestHandler = new
                    Microsoft.Dynamics.Commerce.Runtime.DataServices.SqlServer.CustomerSqlServerDataService();
                    var response =
                    (SingleEntityDataServiceResponse<Customer>)requestHandler.Execute(request);

                    // Execute additional functionality to save the customer's extension properties.
                    if (!request.Customer.ExtensionProperties.IsNullOrEmpty())
                    {
                        // The stored procedure will determine which extension properties are saved to which
                        tables.

                        ParameterSet parameters = new ParameterSet();
                        parameters["@TVP_EXTENSIONPROPERTIES"] = new
                        ExtensionPropertiesExtTableType(request.Customer.RecordId, request.Customer.ExtensionProperties).DataTable();
                        await databaseContext.ExecuteStoredProcedureNonQueryAsync("
                        [ext].UPDATECUSTOMEREXTENSIONPROPERTIES", parameters, resultSettings: null).ConfigureAwait(false);
                    }

                    transactionScope.Complete();

                    return response;
                }
            }
        }
    }
}

```

### Syntax to enable the property to be read later

```
var property = entity.GetProperty("EXTENSION_PROPERTY_ADDED");
```

## Using extension properties on CRT request and response types

Like entities, request and response types can be extended to set and get extension properties. However, they

persist only for the lifecycle of the request and won't be available in POS. If you want to save them to the database, you must write custom code.

```
request.SetProperty("PropertyName", true);
response.SetProperty("PropertyName2", true);
var PropertyName = request.GetProperty("PropertyName");
var BoolPropertyName2 = response.GetProperty("PropertyName2");
```

### Setting or reading extension properties in POS

You can set extension properties in POS and send them to CRT by using the POS APIs. Alternatively, you can send extension properties at the entity level.

To set an extension property on the cart line, you use `SaveExtensionPropertiesOnCartLinesClientRequest`. Likewise, for the cart header, you use `SaveExtensionPropertiesOnCartClientRequest`.

Here is an example.

```
let sampleExtensionProperty = <ProxyEntities.CommerceProperty>{
    Key: "sampleExtensionProperty",
    Value: <ProxyEntities.CommercePropertyValue>{
        BooleanValue: false
    }
};
this.context.runtime.executeAsync(new
SaveExtensionPropertiesOnCartClientRequest([sampleExtensionProperty]));
```

## Reading the extension property from the cart in POS

```
let getCartRequest: GetCurrentCartClientRequest<GetCurrentCartClientResponse> = new
GetCurrentCartClientRequest<GetCurrentCartClientResponse>();
return this.context.runtime.executeAsync(getCartRequest).then((value:
ICancelableDataResult<GetCurrentCartClientResponse>) => {
    let cart: Commerce.Proxy.Entities.Cart = (<GetCurrentCartClientResponse>value.data).result;

    // Gets the extension property from the cart.

    let sampleExtensionPropertyValue: boolean;
    if (!ObjectExtensions.IsNullOrEmpty(cart) &&
!ObjectExtensions.IsNullOrEmpty(cart.ExtensionProperties)) {
        let SampleCommerceProperties: ProxyEntities.CommerceProperty[] =
cart.ExtensionProperties.filter((extensionProperty: ProxyEntities.CommerceProperty) => {
            return extensionProperty.Key === "sampleExtensionProperty";
        });
        sampleExtensionPropertyValue = SampleCommerceProperties.length > 0 ?
SampleCommerceProperties[0].Value.BooleanValue : false;
    } else {
        sampleExtensionPropertyValue = false;
    }

    //Resolve the promise according to your scenario and method.

    return Promise.resolve({ canceled: false });
});
```

Likewise, you can read the extension property from all the other entities, such as products, customers, and addresses.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Manage secrets for retail channels

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how to manage secrets when you're using an extension with channels that require access to secrets. Extensions will not be able to deploy any custom certificates in Commerce scale unit or add thumbprints or secrets in web.config files. The recommended approach for managing secrets is to use the Azure Key Vault, as noted in this topic.

## Key Vault setup

1. The extension developer follows these development steps:
  - a. Create a test secret in Microsoft Azure Key Vault.
  - b. Configure the head-office client to connect to Key Vault.
  - c. On the **Key Vault Parameters** page (**Head Office > Key Vault Parameters**), specify an extension key name for the Key Vault secret. The same name must be used in the next step.
  - d. Use the **GetUserDefinedSecretStringValueServiceRequest** Commerce Runtime (CRT) application programming interface (API) to get the secret. Pass a unique secret name to identify the secret.
  - e. As part of the documentation of the extension setup, state the secret name that is referenced in the extension. We recommend that the extension developer use a namespace for the secret name, because this approach helps prevent conflicts with other extensions.
2. The IT pro or implementation partner follows these deployment and configuration steps:
  - a. Apply the extension to the customer environment. For details, see [Apply updates to cloud environments](#).
  - b. Upload the desired secrets to Key Vault (or enter them). For details, see [What is Azure Key Vault?](#)
  - c. On the **Key Vault Parameters** page (**Head Office > Key Vault Parameters**), configure the head-office client to connect to Key Vault.
  - d. On the **Key Vault Parameters** page, specify the extension secret name for the Key Vault secret in the head-office client.

## Consume the secret in the CRT extension

To consume the secret in the extension, add the following request and response.

REQUEST/RESPONSE	PARAMETERS	DESCRIPTION
<code>GetUserDefinedSecretStringValueServiceRequest</code>	string <b>secretName</b>	The request class that is used to get user-defined secrets from Headquarters.
<code>GetUserDefinedSecretStringValueServiceResponse</code>	string <b>SecretStringValue</b>	The response class that is used to get user-defined secrets from Headquarters. The response returns a <b>SecretStringValue</b> value, and extensions can type-cast this value to <b>X509Certificate2</b> or use it as string value.

To read the secret in the CRT extension, follow these steps.

1. Create a new CRT extension project (C# class library project type). Use the sample templates from the Retail software development kit (SDK) (**RetailSDK\SampleExtensions\CommerceRuntime**).
2. In the CRT extension, you can create a new request/response, or you can add a pre-trigger or post-trigger for the existing CRT request, and then call it. In the following example, a trigger was added for **SaveCartRequest**. It calls **GetUserDefinedSecretStringValueServiceRequest** to read the secret by passing the secret key that is configured in Headquarters. You don't have to write custom code to read the secret from Headquarters. You can use the request and response to read the value.

```

public class CustomSaveCartTrigger : IRequestTrigger
{
    /// <summary>
    /// Gets the list of supported request types.
    /// </summary>
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            return new[]{ typeof(SaveCartRequest) };
        }
    }

    /// <summary>
    /// Pre trigger code.
    /// </summary>
    /// <param name="request">The request.</param>
    public void OnExecuting(Request request)
    {
        ThrowIf.Null(request, "request");
        Type requestedType = request.GetType();
        if (requestedType == typeof(SaveCartRequest))
        {
            // Sample code to get the secret in string format.

            string result = null;

            GetUserDefinedSecretStringValueServiceRequest keyVaultRequest = new
            GetUserDefinedSecretStringValueServiceRequest("SecretName");
            GetUserDefinedSecretStringValueServiceResponse keyVaultResponse =
            request.RequestContext.Execute<GetUserDefinedSecretStringValueServiceResponse>(keyVaultRequest);
            result = keyVaultResponse.SecretStringValue;

            GetUserDefinedSecretCertificateServiceRequest
            getUserDefinedSecretCertificateServiceRequest = new
            GetUserDefinedSecretCertificateServiceRequest(profileId: null, secretName: "SecretName", thumbprint:
            null, expirationInterval: null);
            GetUserDefinedSecretCertificateServiceResponse
            getUserDefinedSecretCertificateServiceResponse =
            request.RequestContext.Execute<GetUserDefinedSecretCertificateServiceResponse>
            (getUserDefinedSecretCertificateServiceRequest);

            X509Certificate2 Certificate =
            getUserDefinedSecretCertificateServiceResponse.Certificate;

            // custom code to additional processing with secrets.
        }
    }

    /// <summary>
    /// Post trigger code.
    /// </summary>
    /// <param name="request">The request.</param>
    /// <param name="response">The response.</param>

    public void OnExecuted(Request request, Response response)
    {
    }
}

```

3. Build the CRT extension project.
4. Copy the output class library, and paste it into ...\**RetailServer**\webroot\bin\Ext for manual testing.
5. In the **CommerceRuntime.Ext.config** file, update the extension composition section with the custom library information. Here is an example.

```
<add source="assembly" value="your custom library name" />
```

## Credential rotation

When this approach is used for credential management, credential rotation is more streamlined. To update a secret, an IT admin just has to update the secret in Key Vault. No change is required to the extension. After a secret is updated, the new value starts to be used when the cache expires.

## Offline support

Offline support for credentials requires that the extension code handle failover to offline when Key Vault credentials aren't available or accessible.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Commerce runtime (CRT) extensibility and triggers

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic explains trigger support for the Dynamics 365 commerce runtime (CRT). CRT supports pre-triggers and post-triggers for every request.

## CRT trigger overview

Commerce runtime (CRT) triggers give you a way to extend the CRT workflow, and let you add business logic before and after every CRT request is executed. The following two methods are used:

- **OnExecuting** – This method is invoked before a request has been processed by a corresponding **IRequestHandler** implementation.
- **OnExecuted** – This method is invoked after the request has been processed by a corresponding **IRequestHandler** implementation.

## CRT trigger extension

To implement a trigger, you must complete these tasks, as shown in the code example that follows:

1. Implement **IRequestTriggerAsync**.
2. Specify **SupportedRequestTypes** to define the request types that the trigger must be executed for.
3. Write a trigger implementation in the **OnExecuting** method if business logic must be run before the request is addressed.
4. Write a trigger implementation in the **OnExecuted** method if business logic must be run after the request is addressed.

**Sample trigger implementation for Get customer data request:**

```

using Microsoft.Dynamics.Commerce.Runtime;
using Microsoft.Dynamics.Commerce.Runtime.DataServices.Messages;
using Microsoft.Dynamics.Commerce.Runtime.Messages;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

public class GetCustomerTriggers : IRequestTriggerAsync
{
    /// <summary>
    /// Gets the supported requests for this trigger.
    /// </summary>
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            return new[] { typeof(GetCustomerDataRequest) };
        }
    }

    /// <summary>
    /// Post trigger code.
    /// </summary>
    /// <param name="request">The request.</param>
    /// <param name="response">The response.</param>
    public async Task OnExecuted(Request request, Response response)
    {
        //Custom logic

        // The only stub to handle async signature
        await Task.CompletedTask;
    }

    /// <summary>
    /// Pre trigger code
    /// </summary>
    /// <param name="request">The request.</param>
    public async Task OnExecuting(Request request)
    {
        // custom logic
        await Task.CompletedTask;
    }
}

```

### Register the extension

Copy and paste the extension library to ...\**RetailServer\webroot\bin\ext folder** and update the **commerceRuntime.ext.config** file with the custom extension library information under composition section. In this example, **Contoso.Commerce.Runtime.Services** is the custom extension name.

For the CRT extension to work in offline mode, update ...\**Microsoft Dynamics 365\70\Retail Modern POS\ClientBroker\ext\CommerceRuntime.MPOSOOffline.ext.config** with the extension library information under the composition section. Then copy and paste the extension library to ...\**Microsoft Dynamics 365\70\Retail Modern POS\ClientBroker\ext**.

### Debugging CRT

To debug CRT from POS, attach the CRT extension project to the w3wp.exe (IIS process for Retail server) when POS is connected to Retail server. For offline mode, attach the CRT extension project to the dllhost.exe process.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Block transactions by using triggers

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic shows how you can use a trigger to block an invoice or credit transaction.

This topic shows how you can block an invoice or credit transaction.

1. Open Visual Studio as an administrator. Create a new Visual C# Class Library (Portable) project and name it CRTTriggerExtension. If you get a message that the selection makes this project incompatible with Visual Studio 2010, click **OK**.
2. In Solution Explorer, rename default class1.cs to GetCustomersServiceRequestTrigger.cs.
3. Right-click the **Reference** node in the project and add the following references. The location of the references will depend on the deployment topology.
  - Microsoft.Dynamics.Commerce.Runtime.Entities.dll
  - Microsoft.Dynamics.Commerce.Runtime.Framework.dll
  - Microsoft.Dynamics.Commerce.Runtime.Services.Messages.dll
4. Add the following **using** statement to the GetCustomersServiceRequestTrigger.cs file.

```
using Microsoft.Dynamics.Commerce.Runtime.Messages;  
using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;  
using Microsoft.Dynamics.Commerce.Runtime;
```

5. Rename class1.cs in the code to GetCustomersServiceRequestTrigger and then add the IRequestTrigger interface declaration.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using Microsoft.Dynamics.Commerce.Runtime;  
using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;  
using Microsoft.Dynamics.Commerce.Runtime.Messages;  
  
namespace CRTTriggerExtension  
public class GetCustomersServiceRequestTrigger : IRequestTrigger  
{
```

6. Implement the IRequestTrigger interface trigger. Right-click the IRequestTrigger class, select **Quick Actions**, and then click **Implement Interface**. Visual Studio will implement the interface. You can also place the cursor on IRequestTrigger, press **Ctrl+**, and select **Implement Interface**.
7. The empty interface members SupportedRequestTypes, OnExecuted, and OnExecuting methods are shown in the following code example.

```

public class GetCustomersServiceRequestTrigger : IRequestTrigger
{
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            throw new NotImplementedException();
        }
    }

    public void OnExecuted(Request request, Response response)
    {
        throw new NotImplementedException();
    }

    public void OnExecuting(Request request)
    {
        throw new NotImplementedException();
    }
}

```

8. Commerce Scale Unit uses the GetCustomersServiceRequest object to get the customer details from Commerce Runtime (CRT) and uses the GetCustomersServiceRequest object to add the customer to the transaction. Before adding the customer to the transaction you need to check whether the customer is blocked. To do this, implement a post trigger for this request and check whether the customer is blocked. If the customer is blocked, then throw the exception to MPOS.
9. In the SupportedRequestTypes method tell the CRT that you are going to add the trigger for GetCustomersServiceRequest. The following code example shows how to add GetCustomersServiceRequest as a supported type.

```

public IEnumerable<Type> SupportedRequestTypes
{
    get
    {
        return new[] { typeof(GetCustomersServiceRequest) };
    }
}

```

10. Check if the customer is blocked in the OnExecuted (post trigger) method with the following code.

```

public void OnExecuted(Request request, Response response)
{
    if (response == null)
    {
        throw new ArgumentNullException("request");
    }

    var getCustomersServiceResponse = (GetCustomersServiceResponse)response;
    if(getCustomersServiceResponse.Customers.FirstOrDefault().Blocked == true)
    {
        string message = string.Format("Failed to add customer '{0}' to cart. Blocked customers are not allowed for transactions.",
            getCustomersServiceResponse.Customers.FirstOrDefault().AccountNumber);
        throw new
CartValidationException(DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_CustomerAccountIsBlocked, message);
    }
}

```

11. Finally, update the OnExecuting method with the following code.

```
public void OnExecuting(Request request)
{
    if (request == null)
    {
        throw new ArgumentNullException("request");
    }
}
```

12. Click **Save**.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Loyalty extension sample

2/18/2021 • 4 minutes to read • [Edit Online](#)

## Scenario

A retailer wants customers to be able to earn loyalty points and pay by using loyalty points in a single transaction.

## Scenario details

The retailer has set up a loyalty scheme for the channel and associated that scheme with a loyalty program. The loyalty scheme includes some earning and redemption rules. The retailer wants customers to be able to partially pay a transaction amount by using their loyalty points. The customers should then be able to earn loyalty points for the remaining transaction amount that they pay by using other payment methods.

## Assumptions

Because of the flexibility that the loyalty setup provides, this scenario can quickly become very complex. We have made some assumptions to reduce the complexity of this sample. However, these assumptions aren't far removed from the real-world examples. Here are the assumptions:

- No tiers are associated with the loyalty program.
- There is a single loyalty scheme and a single loyalty reward point type.
- There is a single earning rule that applies to all product categories. For example, this rule might specify that, for every \$1 that the customer spends, the customer earns 0.1 reward point.
- There is a single redemption rule that applies to all product categories. For example, this rule might specify that one reward point is equivalent to \$1.

### NOTE

The earning and redemption rules that are mentioned here are just examples. For this sample, we have hard-coded the values. For a production scenario, the values should be read from the database instead.

## Customization approach

We implement this customization in two steps. In step 1, points will be earned as though loyalty points weren't used in the transaction. Then, in step 2, the extra points that were earned will be reduced, based on the points that were redeemed.

There are six service requests for the loyalty feature:

- GetLoyaltyCardStatusServiceRequest
- CalculateLoyaltyRewardPointsServiceRequest
- IssueLoyaltyCardServiceRequest
- FillInLoyaltyRewardPointLinesForSalesServiceRequest
- FillInLoyaltyRewardPointLinesForReturnServiceRequest
- FillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest

To calculate the reward points for a cash-and-carry transaction, the system uses

**FillInLoyaltyRewardPointLinesForSalesServiceRequest** for sales transaction lines. This request, in turn, uses **FillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest** to do the actual reward calculation. Similarly, the system uses **FillInLoyaltyRewardPointLinesForReturnServiceRequest** for return transaction lines, and this request also uses **FillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest** to do the actual reward calculation. In the out-of-box implementation of the loyalty feature, customers can't both redeem and earn loyalty points in the same transaction. In other words, if the sales transaction has a tender line for the loyalty card, **FillInLoyaltyRewardPointLinesForSalesServiceRequest** doesn't call **FillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest** to calculate the reward points.

To enable loyalty points to be redeemed and earned in the same transaction, we will work with one service request, **FillInLoyaltyRewardPointLinesForSalesServiceRequest**.

### **Step 1**

For our scenario, we must implement **FillInLoyaltyRewardPointLinesForSalesServiceRequest** in such a way that it always calls **FillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest** to calculate the reward points. However, this change will cause the system to calculate extra earned points, based on the amount that the customer pays by using loyalty points. Therefore, we must be able to reduce the earned points by an appropriate amount. This adjustment is done in step 2.

### **Step 2**

We take advantage of the post trigger mechanism that the extensibility framework provides on **FillInLoyaltyRewardPointLinesForSalesServiceRequest**. In the post trigger, we will use the earning and redemption rules to determine how many extra points were earned for each reward point that was redeemed. We will then subtract those extra points to get the correct amount.

**Implement FillInLoyaltyRewardPointLinesForSalesServiceRequest**



```

namespace Contoso
{
    namespace Commerce.Runtime.EarnRedeemLoyalty
    {
        using System;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        using Microsoft.Dynamics.Commerce.Runtime.DataServices.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;
        using System.Collections.Generic;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;
        public class FillInLoyaltyRewardPointLinesForSalesHandler : IRequestHandler
        {
            /// <summary>
            /// Gets the collection of supported request types by this handler.
            /// </summary>
            public IEnumerable<Type> SupportedRequestTypes
            {
                get
                {
                    return new[]
                    {
                        typeof(FillInLoyaltyRewardPointLinesForSalesServiceRequest),
                    };
                }
            }
            /// <summary>
            /// Entry point to FillInLoyaltyRewardPointLinesForSalesServiceRequest service.
            /// </summary>
            /// <param name="request">The request to execute.</param>
            /// <returns>Returns the SalesTransaction object with the one or more LoyaltyRewardPointLines.
            </returns>
            public Response Execute(Request request)
            {
                ThrowIf.Null(request, "request");
                var LoyaltyRewardPointLinesForSalesServiceRequest =
                (FillInLoyaltyRewardPointLinesForSalesServiceRequest)request;
                SalesTransaction salesTransaction =
                LoyaltyRewardPointLinesForSalesServiceRequest.SalesTransaction;
                // Call the service to calculate the loyalty reward points.
                var fillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest = new
                FillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest(salesTransaction,
                LoyaltyRewardPointLinesForSalesServiceRequest.EarnSchemeLines, LoyaltyRewardPointEntryType.Earn);
                var fillInLoyaltyRewardPointLinesForEarnOrDeductServiceResponse =
                request.RequestContext.Execute<SingleEntityDataServiceResponse<SalesTransaction>>
                (fillInLoyaltyRewardPointLinesForEarnOrDeductServiceRequest);
                salesTransaction = fillInLoyaltyRewardPointLinesForEarnOrDeductServiceResponse.Entity;
                return new SingleEntityDataServiceResponse<SalesTransaction>(salesTransaction);
            }
        }
    }
}

```

**Implement the post trigger for FillInLoyaltyRewardPointLinesForSalesServiceRequest**

```

namespace Contoso
{
    namespace Commerce.Runtime.Sample.EarnRedeemLoyalty
    {
        using System;
        using System.Collections.Generic;
        using System.Linq;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        class AdjustLoyatyRewardsTrigger : IRequestTrigger
        {
            public IEnumerable<Type> SupportedRequestTypes
            {
                get { return new[] { typeof(FillInLoyaltyRewardPointLinesForSalesServiceRequest) }; }
            }
            public void OnExecuted(Request request, Response response)
            {
                ThrowIf.Null(request, "request");
                var LoyaltyRewardPointSalesServiceRequest =
(FillInLoyaltyRewardPointLinesForSalesServiceRequest)request;
                SalesTransaction salesTransaction = LoyaltyRewardPointSalesServiceRequest.SalesTransaction;

                if (salesTransaction.LoyaltyRewardPointLines != null)
                {
                    decimal totalRedeemedPoints = 0m;
                    decimal extraEarnedPoints = 0m;

                    // Find the redeemed reward lines in the transaction and calculate the total redeemed
reward points
                    IEnumerable<LoyaltyRewardPointLine> redeemLoyaltyRewardPointLines =
salesTransaction.LoyaltyRewardPointLines.Where<LoyaltyRewardPointLine>(line => line.EntryType ==
LoyaltyRewardPointEntryType.Redeem);
                    if (redeemLoyaltyRewardPointLines.Count() > 0)
                    {
                        foreach (var rewardPointLine in redeemLoyaltyRewardPointLines)
                        {
                            totalRedeemedPoints += rewardPointLine.RewardPointAmountQuantity;
                        }
                        // Calculate the number of extra earned points for every redeemed point.
                        // If the earning rules stated that for every $1 spent, the user earns X points and
redemption rule was that Y points equal $1 then, for every redemption point the user earns X/Y extra earn
points.
                        // Based on the above logic, in this sample, for every redeemed point the user earns
.1/1 = .1 extra earned points.
                        extraEarnedPoints = .1m * totalRedeemedPoints;
                    }
                    // Reduce the amount of earned points by the extraEarnedPoints calculated above.
                    IEnumerable<LoyaltyRewardPointLine> earnLoyaltyRewardPointLines =
salesTransaction.LoyaltyRewardPointLines.Where<LoyaltyRewardPointLine>(line => line.EntryType ==
LoyaltyRewardPointEntryType.Earn);
                    if (earnLoyaltyRewardPointLines.Count() > 0)
                    {
                        earnLoyaltyRewardPointLines.FirstOrDefault().RewardPointAmountQuantity -=
extraEarnedPoints;
                    }
                }
            }
            public void OnExecuting(Request request)
            { }
        }
    }
}

```

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Extension points for packing slips during order fulfillment

2/18/2021 • 4 minutes to read • [Edit Online](#)

When you generate packing slips for customer orders from the point of sale (POS), you might want to print the item weight on the packing slip. You might also want to add custom information about the package, information about how to call the shipping carrier for the shipping rates, or information about how to return the package. To handle these scenarios, extension points for the packing slip process for order fulfillment have been added in Headquarters, in the commerce runtime (CRT), which is the business logic layer, and in the POS. These extension points let you add custom flow and logic through extensions.

## POS

In the POS, a new request handler that is named **PrintPackingSlipClientRequestHandler** lets you override extensions. This request is called when you select the **Print packing slip** button in the POS. By overriding this request, you can use custom logic in the POS. For example, you can call your own printing methods and print additional information.

For example, by default, you can print the packing slip in PDF format. By overriding the POS client request, you can print the packing slip in a different format. Alternatively, you can check for a specific condition, and then print or stop printing the packing slip, based on that condition. You can also do validation before you print, and prevent printing or custom logic. However, if you want to change the actual data that is printed, you must modify the business logic in CRT and Headquarters.

### Override PrintPackingSlipClientRequestHandler

To override any request in the POS, use the following override handler pattern in the POS.

### Example

```
import { PrintPackingSlipClientRequestHandler } from
"PosApi/Extend/RequestHandlers/StoreFulfillmentRequestHandlers";
import { PrintPackingSlipClientRequest, PrintPackingSlipClientResponse } from "PosApi/Consume/SalesOrders";
import { ClientEntities } from "PosApi/Entities";
/**
 * Override request handler class for printing packing slip request.
 */
export default class PrintPackingSlipClientRequestHandlerExt extends PrintPackingSlipClientRequestHandler {
  /**
   * Executes the request handler asynchronously.
   * @param {PrintPackingSlipClientRequest<PrintPackingSlipClientResponse>} The request containing the
   response.
   * @return {Promise<ICancelableDataResult<PrintPackingSlipClientResponse>>} The cancelable promise
   containing the response.
   */
  public executeAsync(request: PrintPackingSlipClientRequest<PrintPackingSlipClientResponse>):
  Promise<ClientEntities.ICancelableDataResult<PrintPackingSlipClientResponse>> {
    // Do the extension logic here before calling the default handler.
    return this.defaultExecuteAsync(request);
  }
}
```

After you override the request, update the manifest.json file with the new request handler information.

```
"requestHandlers": [  
  {  
    "modulePath": "Handlers/PrintPackingSlipClientRequestHandlerExt"  
  }  
]
```

## CRT

Similar to the POS client request handler, new requests have been added in the CRT to enable extension of custom logic.

### **MarkAsPickedRealtimeRequest**

The **MarkAsPickedRealtimeRequest** request marks the fulfillment lines as picked. You can add custom logic before the line is marked as picked. For example, you can change a property. Later, the picked line information will be used to print the packing slip. Based on your scenario, you can override this request, or add pre-triggers or post-triggers.

For example, for the fulfillment lines, you want to calculate item weight, and so on. In this case, you can customize the service.

#### **NOTE**

If the logic that you require for your customization must be determined in Headquarters, you can customize at the Headquarters level instead of customizing in the CRT. For example, if you want to get the item weight for each line, but that information is available only in Headquarters. In this case, you can customize the relevant real-time service methods and then return the result from Headquarters to the CRT.

### **PackFulfillmentLinesRealtimeRequest**

The **PackFulfillmentLinesRealtimeRequest** request updates the status of the fulfillment lines to **Partially packed** or **Packed**. Based on your scenario, you can override this service, or add pre-triggers or post-triggers.

## Headquarters

In Headquarters new real-time service methods have been added to get the packing slip information, and to add custom information or logic through extension. If your customization requires that data or logic be done in Headquarters, you can customize these methods by using the extension patterns. Additionally, if you require an extension scenario that uses the transaction service client class, you can call these methods from the CRT.

### **RetailTransactionServiceFulfillment**

All the new methods that are related to the packing slip process for order fulfillment were added in the **RetailTransactionServiceFulfillment** class.

#### **GetPackingSlipsData**

The **GetPackingSlipsData** method returns all the packing slip information by passing the sales ID as a parameter.

#### **GetFulfillmentLinesByPackingSlipId**

The **GetFulfillmentLinesByPackingSlipId** method returns the fulfillment lines, based on the packing slip ID. For example, you can use this method if you have the packing slip ID, and you want to get the sales lines that are relevant to that packing slip ID.

#### **MarkFulfillmentLinesAsPacked**

The **MarkFulfillmentLinesAsPacked** method updates the status of the fulfillment lines to packed by taking the fulfillment details as a parameter in XML string format. The POS calls the CRT to mark the selected lines from

the POS user interface (UI) as picked. The **MarkAsPickedRealtimeRequest** request in the CRT then calls the **MarkFulfillmentLinesAsPacked** real-time service method in Headquarters to set the line status as **Packed**.

To add custom logic or return additional information when the line is marked as packed and returned for printing, you can customize the **packingSlipExtensionPoint** method.

### **packingSlipExtensionPoint**

You can customize the **packingSlipExtensionPoint** method to add custom logic or return custom information together with the packing slip ID. This custom information includes packing information or a delivery note. This method is typically added for extensions for custom scenarios. This method is called from the **MarkFulfillmentLinesAsPacked** method by using the chain of command extensibility point. The **MarkFulfillmentLinesAsPacked** method is run from the real-time service call that the CRT code makes.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Channel database extensions

2/18/2021 • 12 minutes to read • [Edit Online](#)

The channel database (channel DB) holds transactional and master data from one or more commerce channels, such as an online store or a brick-and-mortar store. The master data is pushed down from the Headquarters (HQ) to the channel database using the commerce data exchange (CDX). The transactional data stored in the channel database is pulled back to the headquarters using the CDX.

In this topic we explain how to extend the channel database for different scenarios. The steps here apply only to Dynamics 365 Finance and Commerce.

Before discussing the different scenarios for extension, it's important to understand the recent enhancements to channel DB extensions.

We have made some improvements to how extensions are handled during an upgrade. We recommend using one of the following environment configurations:

- Microsoft Dynamics 365 for Finance and Operations, Enterprise edition (July 2017) with application update 5.
- Microsoft Dynamics 365 Retail 7.2 with application update 5, which will be available soon.
- Microsoft Dynamics 365 Retail 7.3, which includes application update 5.
- Microsoft Dynamics 365 for Finance and Operations 7.3, which includes application update 5.

## Ext schema

In Finance and Commerce there is a now schema called the **ext schema** to support extensions. In previous versions, if you wanted to add an extension to channel DB, you would add it to the CRT or AX schema. In both Finance and Commerce, you cannot change the CRT, AX, or DBO schemas. All changes must be made in the **ext schema**. If you modify anything in the CRT or AX schemas, then deployment in Lifecycle Services (LCS) will fail. An error message states that you don't have permission to modify the CRT, AX, and DBO schemas. Extensions will not have permission to read the CRT, AX, and DBO schema definition during deployment, do not include any queries in the extension script to read the CRT, AX, and DBO schema definition.

### NOTE

If you want to increase any channel DB field length, you must create an extensibility request in LCS, increasing the EDT length or decimal precision. Changes will not be automatically pushed to the channel DB, and extensions will not have permissions to change or modify anything in the channel DB - CRT, AX or DBO schema. If you modify anything in the CRT or AX schemas, then deployment in LCS will fail.

## Best practices for channel DB extensions

- Don't modify anything in the CRT, AX, or DBO schemas. Use the **ext schema** for all extension scenarios.
- If available, we recommend getting data through commerce runtime data services, as opposed to accessing channel DB artifacts directly from CRT, AX, or DBO objects.

### Don't do this

The following is an example of what you should not do. Instead, you should use the CRT data service to get the primary key value and then use the primary key to insert into your extension table.

```

MERGE INTO [ax].RETAILCUSTPREFERENCE  --DONT access ax schema object
USING (SELECT DISTINCT
tp.PARENTRECID, tp.PROPERTYVALUE as [EMAILOPTIN], ct.ACCOUNTNUM, ct.DATAAREAID
FROM @TVP_EXTENSIONPROPERTIESTABLETYPE tp
JOIN [ax].CUSTTABLE ct on ct.RECID = tp.PARENTRECID  --DONT access ax schema object
WHERE tp.PARENTRECID <> 0 and tp.PROPERTYNAME = 'EMAILOPTIN') AS SOURCE
ON [ax].RETAILCUSTPREFERENCE.RECID = SOURCE.PARENTRECID
and [ax].RETAILCUSTPREFERENCE.DATAAREAID = SOURCE.DATAAREAID --DONT access ax schema object
and [ax].RETAILCUSTPREFERENCE.ACCOUNTNUM = SOURCE.ACCOUNTNUM
WHEN MATCHED THEN
UPDATE SET [EMAILOPTIN] = source.[EMAILOPTIN]
WHEN NOT MATCHED THEN
INSERT
(
    RECID
    ,DATAAREAID
    ,EMAILOPTIN
    ,ACCOUNTNUM
)
VALUES
(
    SOURCE.PARENTRECID
    ,SOURCE.DATAAREAID
    ,SOURCE.EMAILOPTIN
    ,SOURCE.ACCOUNTNUM
);
SELECT @i_Error = @@ERROR;
IF @i_Error <> 0
BEGIN
    SET @i_ReturnCode = @i_Error;
    GOTO exit_label;
END;

```

### Don't do this

- Don't create any new extension tables, views, or procs in crt, ax, or dbo schema. All extension artifacts must be done in ext schema.
- Don't use any of the crt, ax or dbo schema data types in ext schema. Create custom types in ext schema and use it.
- Don't modify any views, procedures, functions, or any of the database artifacts.
- Avoid accessing or calling database artifacts from your extensions, if possible. Instead, use the CRT data service to get data. The benefits of using the data service are that it will continue to be supported until the SLA, even if breaking changes are made to the database schema in the future. However, there will be instances in which the CRT data service does not expose the data that you need. In these cases, it is still possible to access this data by creating a view which joins on a channel DB artifact. Creating views can be a powerful tool to structure the data in a format you need at a database level, as opposed to doing it in memory through CRT extensions.
- Don't access any dbo.objects from extension scripts because dbo schema objects will not be available in Commerce scale unit deployments.



```

CREATE VIEW [ext].[CONTOSORETAILSTOREHOURSVIEW] AS
(
    SELECT
        sdht.DAY,
        sdht.OPENTIME,
        sdht.CLOSINGTIME,
        sdht.RECID,
        rst.STORENUMBER
    FROM [ext].[CONTOSORETAILSTOREHOURSTABLE] sdht
    INNER JOIN [ax].RETAILSTORETABLE rst ON rst.RECID = sdht.RETAILSTORETABLE
)

```

## Adding extensions

1. If you are creating an extended table and want to sync the data back to HQ, then the table must have the same primary key and clustered index as the HQ table in the extended table, if not, the CDX sync will fail. If you need to pull the data from the extension table to HQ, then the REPLICATIONCOUNTERFROMORIGIN identity column ([REPLICATIONCOUNTERFROMORIGIN] [int] IDENTITY(1,1) NOT NULL,) is required in the extension table.
2. All extension table columns must have the NOT NULL constraint enforced. During upgrade, if the column value is blank it will be updated with NULL values and it may cause a runtime exception in CRT if the null value is not handled properly.
3. All the extension tables should have grant permission on **UserRole** and **DeployExtensibilityRole**.

```

--Tables:

GRANT SELECT, INSERT, UPDATE, DELETE ON OBJECT::[ext].[RETAILCUSTPREFERENCE] TO [UsersRole]
GO

GRANT SELECT, INSERT, UPDATE, DELETE ON OBJECT::[ext].[RETAILCUSTPREFERENCE] TO
[DeployExtensibilityRole]
GO

--Stored procedures:

GRANT EXECUTE ON [ext].[EXTSTOREDPROCEDURE] TO [UsersRole];
GO

GRANT EXECUTE ON [ext].[EXTSTOREDPROCEDURE] TO [PublishersRole];
GO

GRANT EXECUTE ON [ext].[EXTSTOREDPROCEDURE] TO [DeployExtensibilityRole];
GO

```

4. Grant **DataSyncUsersRole** permission if your table is going to send or receive data from HQ.

```

GRANT SELECT, INSERT, UPDATE, DELETE, ALTER ON OBJECT::[ext].[EXTTABLENAME] TO [DataSyncUsersRole]
GO

```

5. Always prefix your table, for example **ContosoRetailTransactionTable**, so that you can avoid conflicts with other customizations.

## Attributes

We extended the attribute framework in HQ to support attributes for Customers, Customer orders, cash and carry transactions and call center orders.

## Customer attributes

With the new customer attribute framework, you can use configurations to add new fields to the customer add/edit or customer details screens in POS or HQ. After configuring the customer attribute group in commerce parameters, POS and HQ will automatically show up the new attribute without any code change or customization. The screen layout designer will also be configured to show the customer attributes in the transaction screen - **Customer** panel.

## Order attributes

The attribute framework was extended to support attributes in cash and carry transactions, customer orders, and call center orders. You can edit and set values directly in HQ or in CRT. All this can be done through configurations, without any database changes. (You can customization the attribute values for core business logic, not required for basic CRUD operations.) Previously, you had to create new tables in HQ and channel DB, and then modify CRT to do this. Now all the attribute creation can be done through configuration.

## Adding a new table

In this scenario we will explain how to create a new table and add it to the channel DB. All extension code has access to the **ext schema**.

- Create a new table in the channel database in the **ext schema** either using SQL Server Management Studio Designer or using SQL scripts. The following is an example SQL script.

```
-- Create the extension table to store the custom fields.
IF (SELECT OBJECT_ID('[ext].[CONTOSORETAILSTOREHOURSTABLE]')) IS NULL
BEGIN
CREATE TABLE [ext].[CONTOSORETAILSTOREHOURSTABLE](
[RECID] [bigint] NOT NULL,
[DAY] [int] NOT NULL DEFAULT ((0)),
[OPENTIME] [int] NOT NULL DEFAULT ((0)),
[CLOSINGTIME] [int] NOT NULL DEFAULT ((0)),
[RETAILSTORETABLE] [bigint] NOT NULL DEFAULT ((0)),
[REPLICATIONCOUNTERFROMORIGIN] [int] IDENTITY(1,1) NOT NULL,
[ROWVERSION] [timestamp] NOT NULL,
[DATAAREAID] [nvarchar](4) NOT NULL,
CONSTRAINT [I_CONTOSORETAILSTOREHOURSTABLE_RECID] PRIMARY KEY CLUSTERED
(
[RECID] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
ALTER TABLE [ext].[CONTOSORETAILSTOREHOURSTABLE] WITH CHECK ADD CHECK (([RECID]<>(0)))
END
GO
GRANT SELECT, INSERT, UPDATE, DELETE ON OBJECT::[ext].[CONTOSORETAILSTOREHOURSTABLE] TO
[DataSyncUsersRole]
GO
```

### NOTE

If the new extension table data needs to be pulled to Retail headquarters using Commerce Data Exchange (CDX), then the extension table must include the

```
REPLICATIONCOUNTERFROMORIGIN identity column ([REPLICATIONCOUNTERFROMORIGIN] [int] IDENTITY(1,1) NOT NULL,), [ROWVERSION] [timestamp] NOT NULL
```

and [DATAAREAID] [nvarchar](4) NOT NULL (required if the table data is per company). This is required for a CDX pull job. REPLICATIONCOUNTERFROMORIGIN is not required if the data is pushed from Retail headquarters to channel database, this is only needed if the data is pulled from channel database to Retail headquarters.

## Extending an existing table

If you are extending existing table, then you must either use attributes if supported for that entity or create and extended table (new table) with same primary key as the parent table. The following script extends a table.

```
CREATE TABLE [ext].[RETAILTRANSACTIONTABLE](
  [TRANSACTIONID] [nvarchar](44) NOT NULL, -- FK to [crt].RETAILTRANSACTIONTABLE
  [ISB2BSALES] [int] NOT NULL DEFAULT (0),
  [EXTERNALID] [nvarchar](20) NOT NULL DEFAULT (''),
  CONSTRAINT [EXT_RETAILTRANSACTIONTABLE_PK] PRIMARY KEY CLUSTERED
  (
    [TRANSACTIONID]
  ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
  ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
GRANT INSERT ON [ext].[RETAILTRANSACTIONTABLE] TO [DataSyncUsersRole];
GO
GRANT DELETE ON [ext].[RETAILTRANSACTIONTABLE] TO [DataSyncUsersRole];
GO
GRANT UPDATE ON [ext].[RETAILTRANSACTIONTABLE] TO [DataSyncUsersRole];
GO
GRANT SELECT ON [ext].[RETAILTRANSACTIONTABLE] TO [DataSyncUsersRole];
GO
```

## Adding new views, stored procedure, functions, and defined types

All new stored procedures, views or functions must be created in the **ext** schema. Don't access or call our database artifacts from your procedures, views, or functions.

## Deployment checks

The deployment process determines if there are any modification to the database artifacts. If you have attempted to modify the CRT, AX, or DBO schema objects, or access them for any scenario directly in SQL, then deployment will fail.

## Extension scripts and deployment

Channel Database extensions are provided by authoring one or more T-SQL script files and including them in a [deployable package](#). This process is described in the [Retail SDK](#) documentation.

Extension script files must be written using [T-SQL](#) and compatible with [Azure SQL Database](#). The script files must end with the `.sql` file extension, any other files will be ignored or may induce a packaging or deployment failure. If you intend to deploy your Channel Database extensions as part of Commerce Scale Unit or Modern POS offline, the scripts must also be compatible with the version of SQL Express and/or SQL Server that will be used for those components.

During deployment and installation, the extension scripts are executed in alphabetical order based on the script file name. Each script is run to completion and then a metadata record is added to the Channel Database's CRT.RETAILUPGRADEHISTORY table to track the completion of that extension script. The script will not be executed again for the same Channel Database in subsequent deployments if that metadata record is present. If a script fails during execution and does not complete successfully, its metadata will not be stored and the script will be rerun on subsequent deployments.

If the deployment or installation is combined with an update of the product, the extension scripts are run after the product update.

To author a successful Channel Database extension, you must adhere to the following guidelines.

## Use a naming convention that ensures stable order when sorted alphabetically

Because extension scripts are executed in alphabetical order based on the file name, you should establish a naming convention that ensures that the correct execution order is used when sorted.

One example would be naming files with the following pattern: `<ISO 8601 date>_<descriptio>.sql`, where `<ISO 8601 date>` is a ISO 8601 formatted date and `<description>` is descriptive text to identify the purpose of the script. For instance, `"20180501_CustomerDetails.sql"` and `"20181102_CustomerDetailsIndex.sql"`. The former would represent an extension script authored on May 1, 2018 that is related to "Customer Details" feature and the latter an extension script associated to indexes related to the previous feature authored on November 2, 2018.

Another simpler alternative is to use an incremental numeric prefix, such as `"0001_CustomerDetails.sql"` and `"0002_CustomerDetailsIndex.sql"`.

If one script depends on another having executed successfully, you must name then in a way that ensures that the file name in alphabetical order matches the required execution order.

## Do not alter extension scripts that have been published

If you have released a deployable package or installer extension that contains Channel Database extension scripts, do not alter those scripts. Extension scripts are run only once per Channel Database instance. If you alter published scripts and those scripts may have already been run against a Channel Database, the modifications to an already executed script will not be applied to the database.

Instead, provide the modifications in a new script file. Consider the naming convention noted above to ensure that it runs after its dependencies.

## Do not remove old extension scripts that have been published

Your deployable package or installer extension must represent a cumulative update for your database extensions. There should be no dependencies on previous versions of an extension package or installer. A customer should be able to apply your extension package or installer without depending on a previous version of your package or extension.

If your extension scripts have been published as part of a deployable package or installer extension, do not remove them from subsequent updates in your package or installer. To account for disaster recovery, upgrade and scale out scenarios, extension packages may be used to bring a new instance of the Channel Database to the same version of the last deployed extension package.

## Extension scripts must be idempotent and reentrant

Although extension scripts are run only once per Channel Database, scripts may fail due to authoring errors or transient SQL errors, like time outs or transaction deadlocks. The extension scripts must be idempotent and reentrant to account for those failure scenarios. If the extension script fails due to any error, it may be rerun. Rerunning the script should not adversely affect the database.

## Do not assume that the Channel Database data is perennial

The Channel Database is a transactional database that provides storage support for operations performed by Commerce Scale Unit. All data that is stored in the Channel Database that must be kept for long periods of time must be uploaded to the headquarters through the [Commerce Data Exchange](#). Data uploaded to the headquarters can be accessed by the [Commerce Data Exchange Real-time Service](#).

## Do write backward compatible channel database extensions

The Channel Database is expected to be backward compatible. This means that updating only the Channel Database without updating Commerce Scale Unit or POS must not prevent existing Commerce Scale Unit or POS operations from functioning correctly. During deployment flows, the different components of your Commerce Scale Unit and Modern POS are updated in the inverse order of dependency. This means that the Channel Database is the first component to be updated, and Commerce Scale Unit or POS are updated next. If

Commerce Scale Unit or POS fails to update successfully, those components are rolled back to restore them to their previous working state. However, in such situations, the Channel Database is not rolled back to prevent data loss. If your extensions are not backward compatible, they may fail to work properly until a successful deployment is performed.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable custom Commerce Data Exchange synchronization via extension

2/18/2021 • 19 minutes to read • [Edit Online](#)

This topic explains how you can extend the Commerce initialization class to support custom Commerce Data Exchange (CDX) synchronization. For this extension, you use the new extension points that were added in Microsoft Dynamics 365 for Finance and Operations platform update 8 or Microsoft Dynamics 365 Retail platform update 8.

CDX is a system that transfers data between Commerce Headquarters (HQ) and channels, such as online stores or brick-and-mortar stores. The data transfer between HQ and the channel database is controlled by scheduler jobs. Each scheduler job contains a list of scheduler subjobs. The scheduler subjobs contain the names of the source tables and destination tables, and the transfer field mapping of those tables. There are two ways to configure the data synchronization between HQ and the channel database:

- Configure all the custom jobs and subjobs by using the configuration user interface (UI) for CDX.
- Extend the Commerce initialization class by using the extension points that are provided to support custom jobs and subjobs for both push and pull.

The advantage of using the Commerce initialization class is that you don't have to configure the custom jobs in different environments (dev, test, and production). Instead, you can run the CDX initialization by using the **Initialize commerce scheduler** dialog box from **Retail and Commerce > Headquarters setup > Commerce scheduler > Initialize commerce scheduler**. Information about the custom job for the data synchronization is then automatically created in CDX.

There are various scenarios for data transfer between HQ and the channel database:

- Send data from a new HQ table to a new channel database table by using a download job.
- Pull data from a new channel database table to a new HQ table by using a push job.

## Send data from a new HQ table to a new channel database table by using a download job

Before you push or pull data, you must understand various metadata definitions in the XML resource file. The resource file contains the custom job information that will be initialized in your environment to push and pull data. Here is a list of the resource files that you must configure:

- **ChannelDBSchema** – The extension schema that you created in the channel database.
- **TargetTableSchema** – The extension schema that you created in the channel database to add your custom tables.
- **AxTableName** – The table name.
- **IsUpload** – A flag that determines whether the job is a push job or a pull job. (In other words, the flag indicates whether you want to send data from HQ to the channel database or pull data from the channel database to HQ). The default value is **false**, which indicates that you're sending data from HQ to the channel database.
- **ScheduledByJob** – This resource file contains one or more subjobs.
- **Subjob** – Each table is added as a subjob, and each subjob is scheduled by one or more scheduler jobs.
- **TargetTable** – The name of the channel database table. This table is the target table that the push job or pull job must send data to. If a value isn't specified, the system assumes that name of the target table and the

name of the source table are the same.

If you created a new HQ table and a new channel database table, follow these steps to push the data between the two tables.

1. Create a custom project and use the Application Object Tree (AOT) to add a custom table.
2. Create a new resource file to add all custom job information. Here is the template for the resource file.

```
<RetailCdxSeedData ChannelDBMajorVersion="7" ChannelDBSchema="ext" Name="AX7">
  <Jobs>
  </jobs>
  <Subjobs>
    <Subjob Id="" TargetTableSchema="" TargetTableName="">
  </Subjobs>
</RetailCdxSeedData>
```

#### NOTE

The **DataAreaid** column name should not be explicitly included in the field mapping. This is automatically added by Commerce Data Exchange (CDX). If added, an error will occur during initialization of the retail scheduler.

3. Use the AOT to create a new XML resource. In the XML file for the resource, specify the new table and new job details, as shown in the following example.

#### NOTE

You can either add the new table as part of the existing job, or create a new job and add this table. In this case, we are creating a new job, where the job ID is **7000** and the custom table is named **ContosoRetailSeatingArrangementData**.

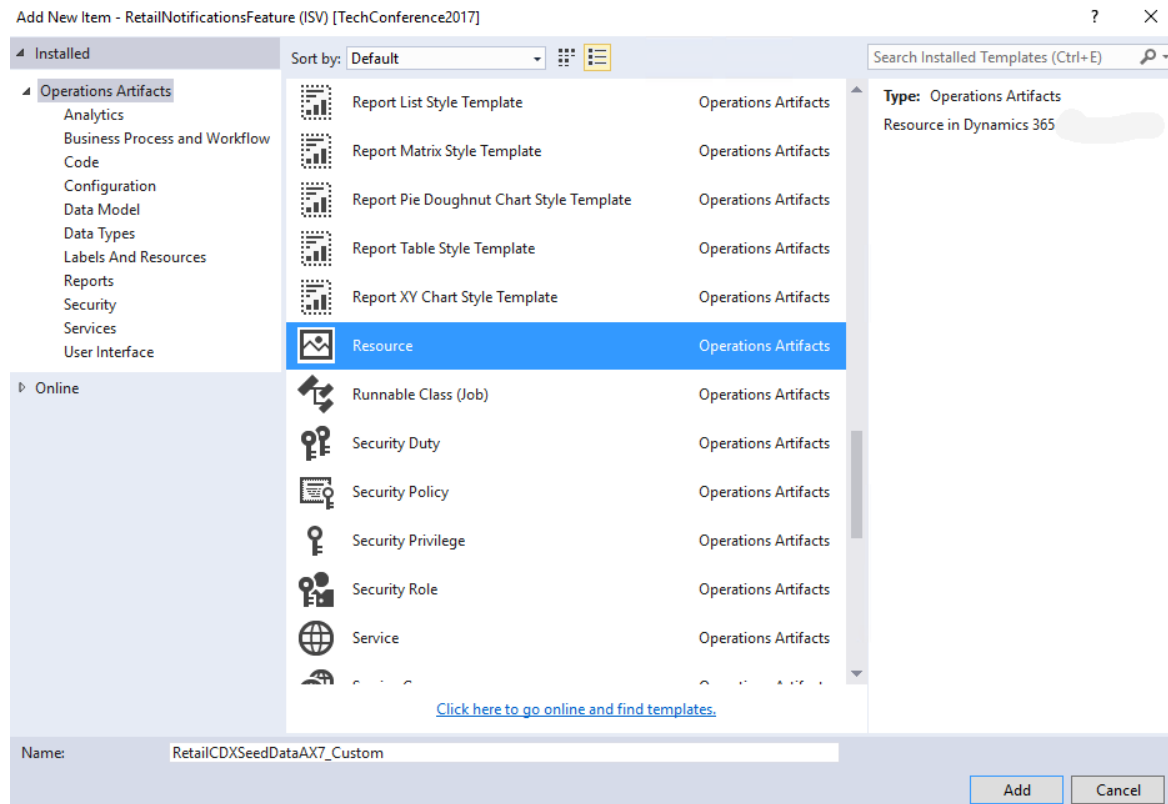
```
<RetailCdxSeedData ChannelDBMajorVersion="7" ChannelDBSchema="ext" Name="AX7">
  <Jobs>
    <Job DescriptionLabelId="REX4520710" Description="Custom job" Id="7000"/>
  </Jobs>
  <Subjobs>
    <Subjob Id="ContosoRetailSeatingArrangementData" TargetTableSchema="ext"
AxTableName="ContosoRetailSeatingArrangementData">
      <ScheduledByJobs>
        <ScheduledByJob>7000</ScheduledByJob>
      </ScheduledByJobs>
      <AxFields>
        <Field Name="seatNumber"/>
        <Field Name="capacity"/>
        <Field Name="channelRecId"/>
        <Field Name="RecId"/>
      </AxFields>
    </Subjob>
  </Subjobs>
</RetailCdxSeedData>
```

By default, the name of the target table isn't specified here. The system assumes the name of the target table on the channel side is the same as the name of the source table on the Commerce side (**AXTableName**). However, the name of the target table on the channel side might sometimes differ from the name of the source table. In this case, in the **<Subjob>** node, you can use the **<TargetTableName>** attribute to set the name of the target table on the channel side.

Similarly, in the mapping section, only the names of fields on the Commerce side are specified (**AxFields**). By default, it's assumed that the same field name is also used on the channel side. However, the field name on the corresponding channel table might sometimes differ from the field name on the

Commerce side. In this case, in the mapping, you can use the **ToName** attribute of the <Field> node to set the name of the field on the channel side.

4. Right-click the project, and then select **Add > New Item**.
5. In the **Add New item** dialog box, select **Resources**, name the resource file **RetailCDXSeedDataAX7\_Custom**, and then select **Add**.



6. In the **Select a Resource file** dialog box, find the resource file that you created in step 2, and then select **Open**.
7. Add a new class that should be used to handle the **registerCDXSeedDataExtension** event. Search for the **RetailCDXSeedDataBase** class, and then open it in the designer. Right-click the **registerCDXSeedDataExtension** delegate, and then select **Copy event handler**.
8. Go to the event handler class that you created and paste the following event handler code into it.

```
if (originalCDXSeedDataResource == resourceStr(RetailCDXSeedDataAX7))
{
    resources.addEnd(resourceStr(RetailCDXSeedDataAX7_Custom));
}
```



## NOTE

Because there are two definitions for CDX seed data in the system, you must specify that your extension CDX seed data should be added only if the CDX seed data that is being generated is the version that you're trying to extend. If the `if` condition is removed, your extension CDX seed data could also be applied on top of the N-1 CDX seed data and cause unintended results. As a best practice, try to avoid any other customization on CDX/Retail scheduler sync framework class in X++. This could impact the flow of data when extra processing is performed. The suggested pattern is to have a separate class and batch job to process the uploaded data.

You don't have to create separate resource files for the various scenarios that are mentioned later. You can have one file that contains all the custom job information and register that file from the extension class.

When the initialization class runs, it looks for any extension that implements this handler. If an extension is found, the runtime will also initialize the custom information that is found in the resource file.

9. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Initialize commerce scheduler**.
10. Run the CDX initialization by clicking the **OK** button on **Initialize commerce scheduler** dialog.

## Pull data from a new channel database table to a new HQ table by using a push job

To pull data from a new channel table to HQ, you have two options:

- Create a new resource file and add the new resource to the event handler as a second line, as shown here.

```
if (originalCDXSeedDataResource == resourceStr(RetailCDXSeedDataAX7))
{
    resources.addEnd(resourceStr(RetailCDXSeedDataAX7\_Custom));
    resources.addEnd(resourceStr(RetailCDXSeedDataAX7\_Custom1));
}
```

- Update the existing resource file with the new information, so that you don't have to add a new line. To upload you set the **IsUpload** attribute to **true** in the resource file and add information about your custom pull job, as shown in the following example.

```
<Subjob Id="ContosoRetailSeatReservationTrans" TargetTableSchema="ext" IsUpload="true"
ReplicationCounterFieldName="ReplicationCounterFromOrigin"
AxTableName="ContosoRetailSeatReservationTrans">
    <ScheduledByJobs>
        <ScheduledByJob>P-1000</ScheduledByJob>
    </ScheduledByJobs>
    <AxFields>
        <Field Name="transactionId"/>
        <Field Name="storeId"/>
        <Field Name="terminalId"/>
        <Field Name="contactPhoneNo"/>
        <Field Name="numberOfCustomers"/>
        <Field Name="customerName"/>
        <Field Name="reservationDate"/>
        <Field Name="reservationTime"/>
        <Field Name="replicationCounterFromOrigin"/>
    </AxFields>
</Subjob>
```

## NOTE

If you are creating an extended table and want to sync the data back to HQ, then the table must have the same primary key and clustered index as the HQ table in the extended table, if not, the CDX sync will fail. If you need to pull the data from the extension table to HQ, then the REPLICATIONCOUNTERFROMORIGIN identity column ([REPLICATIONCOUNTERFROMORIGIN] [int] IDENTITY(1,1) NOT NULL,) is required in the extension table.

You can either add this new table as part of the existing pull job (P-0001) or create a new pull job.

## Other scenarios

For the remaining push and pull scenarios, only the information for the sample resource file is described, because initialization is the same as we described in the previous sections.

### Push existing headquarters tables to channel database that are not part of CDX configurations

In this case, the extension should create a new sub job with the same name as the core table and create the same table in the channel database ext schema and map it. Because CDX doesn't support multiple sub jobs for the same table, the sub job name must match the core table name to avoid any future conflicts. In the future, the headquarters table may be added for CDX push/pull by out-of-band (OOB) products with the sub job name same as the core table. CDX framework will automatically merge if there are any duplicate sub job names.

### Push existing columns that aren't mapped as part of any subjobs

You can push the existing unmapped column to either new extension columns or existing columns in the channel database, as shown in the following example.

```
<Subjob Id="RetailChannelTable" TargetTableSchema="ext">
  <AxFields>
    <Field Name="Payment"/>
    <!-- Existing column which was not pushed to channel db-->
    <Field Name="PaymMode"/>
    <!-- Existing column which was not pushed to channel db-->
    <Field Name="ContosoRetailWallPostMessage"/>
    <!-- New column from the extended table -->
  </AxFields>
</Subjob>
```

If the table has a primary key that isn't **RecId**, your extension table on the channel side should also contain the non-**RecId** primary keys, as shown in the following example.

```
<Subjob Id="RetailCustTable" TargetTableSchema="ext">
  <AxFields>
    <Field Name="ReturnTaxGroup_w"/>
    <!-- Existing column which was not pushed to channel db-->
    <Field Name="SSNNumber"/>
    <!-- New column from the extended table-->
  </AxFields>
</Subjob>
</Subjobs>
```

### Pull new columns to an existing table

If you add new columns and want to pull in part of the existing table, use the following code.

```
<Subjob Id="RetailTransactionTable" TargetTableName="CONTOSORETAILTRANSACTIONTABLE" TargetTableSchema="ext"
OverrideTarget="false">
  <AxFields>
    <Field Name="ContosoRetailSeatNumber"/>
    <Field Name="ContosoRetailServerStaffId"/>
  </AxFields>
</Subjob>
```

### Move an existing subjob to another subjob

To move an existing subjob to another job, you can change the **ScheduledByJob** attribute in the resource file and it is run as part of the event handler.

```
<Subjob Id="DirPartyTable">
  <ScheduledByJobs>
    <ScheduledByJob>1000</ScheduledByJob>
    <!--add existing subjob to another job-->
  </ScheduledByJobs>
```

## CDX sample - Pull new columns to an existing table

In Microsoft Dynamics 365 Retail App update 5, we added a new sample in **RetailSDK\Documents\SampleExtensionsInstructions\ExtensionTables**, it has all the sample SQL scripts, ax project files for different CDX extension scenarios, please use it as a reference for different CDX extension scenarios.

In the next sections, we discuss the steps and best practices for customizing transactional tables by using extension tables. Another section shows how to customize CDX to upload the customized (extension) tables on the channel side back to Commerce. We have also included a section that describes how to test the customization.

### Setup steps

We recommend that you implement these changes on an untouched Retail software development kit (SDK). Alternatively, you can put the SDK under source control, such as Microsoft Azure DevOps, so that you can easily revert your changes at any step. To begin, you import the \*.axpp package that is located in the SDK. You then run the SQL update script on your channel database.

1. Import the package on the Commerce side that contains the customization code:
  - a. Copy the ExtensionTablesAndCDXCustomization.axpp file from the RetailSDK\Documents\SampleExtensionsInstructions\ExtensionTables folder and paste in your extension project folder.
  - b. Start Microsoft Visual Studio.
  - c. Select **Dynamics 365 > Import project**.
  - d. In the **Import project** dialog box, specify the path of the .axpp file you copied in step 1.
  - e. Select either **Current solution** or **New solution**, according to your preference.
  - f. Select **OK** to begin to import the package.

After the import is completed, you have the files in Solution Explorer.

- g. Build the solution.
- h. Right-click the project, and then select **Synchronize database**.

2. Run the SQL update script:

- a. Copy the **ContosoRetailExtensionTablesUpdate.sql** file from the Retail SDK folder. You can run the other sample files in a similar manner.
- b. Open the script in Microsoft SQL Server Browser, and run the script against your channel database.

This step creates the extension tables that are required in order to customize the transactional tables. Note that the script also creates other tables that are used for other sample scenarios.

### Extend the data in the sample

The table extension on the Commerce side is already created in the sample. To create it manually, follow these steps.

1. Start Visual Studio.
2. On the menu, select **View > Application Explorer**.
3. Select **Data Model > Tables > RetailTransactionTable**, right-click **RetailTransactionTable**, and then select **Create extension**.

As a best practice, you should change the default name to something like

**RetailTransactionTable.ContosoRetailExtension**. Always add your unique prefix. In this sample, **ContosoRetail** is used as a unique prefix. By using a unique prefix, you help prevent naming conflicts if a table is extended by multiple independent software vendors (ISVs).

4. In the new **RetailTransactionTable.ContosoRetailExtension** table, create two new fields:

**Type=string, name=ContosoRetailServerStaffId**: Set the **Extended data type** property to **RetailStaffId**. **Type=int, name=ContosoRetailSeatNumber**: Set the **Extended data type** property to **ContosoRetailSeatNumber**.

5. Save the changes, and build your project.
6. Right-click your project, and then select **Synchronize the database**.

#### NOTE

As a best practice, the unique prefix is added to the new column names to help prevent future naming conflicts. A naming conflict can occur if another ISV creates a column that has the same name, or if Microsoft releases an update that uses a column that has the same name. Even though the extension table is created in a different AOT asset, the new columns are added to the original table in SQL.

### Extend the database on the channel side

From the Retail SDK folder, open and run the SQL Server **ContosoRetailExtensionTablesUpdate.sql** script. Several items are created and configured:

- The **[ext].ContosoRetailTransactionTable** table that has the foreign key and custom (extension) fields is created. In addition to the extension columns that we added in the tables, the extension table on the channel side must have the same primary key columns as the original table on the channel side. Therefore, **[ext].RetailTransactionTable\_ContosoRetailExtension** has the four primary key columns that are used in **[ax].RetailTransactionTable**. As a best practice, when you add the primary key columns to the extension table on the channel side, keep the names of the columns the same as the names of the primary key column on the original table.
- CDX is configured to upload and pull the custom columns from the channel extension table back to Commerce. The **RetailCDXSeedDataAX7** resource contains the information for the table mapping from Commerce to the channel database. CDX uses this information to create the required data transfer

scheduler jobs and subjobs. To include your new extension tables or columns in the data transfer, you must provide a resource file that specifies the customization for the CDX data transfer. As a best practice, use the following naming convention to prevent conflicts:

**RetailCDXSeedDataAX7\_ContosoRetailExtension.** (Here, **ContosoRetail** is your unique extension.)

The sample CDX resource file in the Retail SDK contains additional customizations. However, for our example of RetailTransactionTable extension, the section in the following code is the only section that is required to pull data from the channel side back to HQ.

```
<RetailCdxSeedData Name="AX7" ChannelDBSchema="ext" ChannelDBMajorVersion="7">
  <Subjobs>
    <!--Adding additional columns to (existing) RetailTransactionTable and wants to pull it back to
    HQ.For upload subjobs, set the OverrideTarget property to "false", as illustrated below. This will tell CDX
    to use the table defined by TargetTableName and TargetTableSchema as extension table on this subjob.-->
    <Subjob Id="RetailTransactionTable" TargetTableName ="CONTOSORETAILTRANSACTIONTABLE"
    TargetTableSchema="ext" OverrideTarget="false">
      <!--Notice that there is no mention of the <ScheduledByJobs></ScheduledByJobs> because the
      subjob is already part of an upload job. -->
      <AxFields>
        <!--If you notice the existing columns are not listed here in the <Field> tag, it's because
        the existing fields are already mapped in the main RetailCdxSeedData resource file, we only add the delta
        here. -->
        <Field Name="ContosoRetailSeatNumber" />
        <Field Name="ContosoRetailServerStaffId" />
      </AxFields>
    </Subjob>
  </Subjobs>
</RetailCdxSeedData>
```

**Description of the fields used in this resource file:**

**ChannelDBSchema='ext'** – This field is included so that the resource reads from the extension schema in the channel database.

**Subjob Id= "RetailTransactionTable"** – You must make sure that the SubJob ID is the same as the original subjob id for that table. so that the extensibility framework can determine that you're customizing the existing subjob. If you use new subjob di, system will throw duplicate subjob error for the same table.

**TargetTableName = "CONTOSORETAILTRANSACTIONTABLE"** - Your channel extension table name.

**TargetTableSchema="ext"** - Your channel extension schema. Currently we support the extension schema name only as ext.

**OverrideTarget="false"** - For upload subjobs (the ones that bring data from the channel to the headquarters), OverrideTarget when set to "false" will tell CDX that the table defined by TargetTableName is an extension table and data will be uploaded along with the primary table already defined in the subjob.

If OverrideTarget is set to "true" the table defined by TargetTableName will override the primary table for the subjob (default value fields will be omitted during the pull job and only the extension fields will be considered). For instance, in this sample, if you set this value to true, this would mean that instead of uploading the data from ax.RetailTransactionTable, CDX would only upload the data from ext.CONTOSORETAILTRANSACTIONTABLE.

The **AxTableName** attribute isn't specified, because the framework can already determine the **AxTableName** value that the specified subjob uses as a sink. You only have to specify the differences when you customize the RetailCDXSeedDataAX7 resource. Any data that the framework can infer doesn't have to be added by extensions. Similarly, in the **</AXFields?>** section, you can see that we specified only the custom or new fields, because the extensibility framework can determine the list of remaining fields from the specified subjob ID.

- The CDX module that has the CDX customization resource is updated. To apply the customization that is specified in RetailCDXSeedDataAX7\_ContosoRetailExtension, you must subscribe to the

registerCDXSeedDataExtension delegate. By subscribing to this event, you help guarantee that the customization is applied when initialization of the CDX seed data is run.

#### Subscribe to the registerCDXSeedDataExtension delegate

1. Select **View > Application Explorer**.
2. Search for the **RetailCDXSeedDataBase** class.
3. Right-click the class, and then select **Open in designer**.
4. In the designer, in the list of delegates and methods, select the **registerCDXSeedDataExtension** delegate.
5. Right-click, and then select **Copy event handler**. The method signature that you must implement is copied, so that CDX picks up the customized resource for CDX seed data.
6. Create a new class, and give it a name, such as **ContosoRetailCDXSeedDataAX7EventHandler**. You can specify any name. However, as a best practice, be sure to prefix the class name with your prefix.
7. Paste the code that you copied in step 5.

```
class ContosoRetailCDXSeedDataAX7EventHandler
{
    /// <summary>
    /// Registers the extension CDX seed data resource to be used during CDX seed data generation.
    /// </summary>
    /// <param name="result">The result object which is used to return the resource name.</param>
    [SubscribesTo(classStr(RetailCDXSeedDataBase), delegateStr(RetailCDXSeedDataBase,
registerCDXSeedDataExtension))]
    public static void RetailCDXSeedDataBase_registerCDXSeedDataExtension(str
originalCDXSeedDataResource, List resources)
    {
    }
}
```

8. The CDX extensibility framework calls this method when you select the Commerce initialization. To help guarantee that the CDX extensibility module uses the CDX customization, paste the following code into the preceding method.

```
if (originalCDXSeedDataResource == resourceStr(RetailCDXSeedDataAX7))
{
    resources.addEnd(resourceStr(RetailCDXSeedDataAX7_ContosoRetailExtension));
}
```

Before you add your custom resource to the list, you must verify that the originalCDXSeedDataResource resource that is being processed is RetailCDXSeedDataAX7. Otherwise, you might cause unintended results.

9. To initialize or reinitialize the CDX module with the customized configuration, follow these steps:
  - a. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Scheduler jobs > Initialize commerce scheduler**.
  - b. In the dialog box that appears, select **Delete existing configuration**.
  - c. Select **OK** to start the initialization.

When the initialization is completed, the CDX scheduler jobs, subjob definitions, and distribution schedules are updated by using the original RetailCDXSeedDataAX7 resource and the customized RetailCDXSeedDataAX7\_ContosoRetailExtension resource.

## Validate the customization

1. Verify that your customization works correctly:
  - a. After the initialization is completed, go to **Retail and Commerce > Headquarters setup > Commerce scheduler**, and then select the **Scheduler subjobs** link.
  - b. On the subjobs table, search for the **RetailTransactionTable** subjob ID.
  - c. In the details area, under the **Channel field mapping** section, verify that the new custom (extension) columns are listed in the mapping.
2. Test that the CDX job will upload and pull from the original and extension tables on the channel side(a view combining original and extensible tables is generated by the CDX framework):
  - a. Create some transactions in Retail Modern POS (MPOS).
  - b. Because the extension table isn't used in the Commerce Runtime (CRT) and MPOS, you must manually insert data into the extension table. Run the following script after you change the required values.

```
INSERT INTO [ext].[CONTOSORETAILTRANSACTIONTABLE] (  
    [CONTOSORETAILSEATNUMBER],  
    [CONTOSORETAILSERVERSTAFFID],  
    [TRANSACTIONID],  
    [STORE],  
    [CHANNEL],  
    [TERMINAL],  
    [DATAAREAID])  
VALUES (  
    1, /*normally this needs to be an existing seat number from ContosoRetailSeatingData table,  
    but for this test add any number*/  
    '000160' /*add any staff ID here*/,  
    'HOUSTON-HOUSTON-11-101', /*add the transaction id you just created */  
    'HOUSTON', /*add the store used to create the transaction */  
    5637144592, /*add the channel RecId of the store used to create the transaction*/  
    'HOUSTON-11', /*add the terminalId used to create the transaction*/  
    'USRT' /*add the dataareaId used by the store*/)   
GO
```

Repeat this step for the other transactions. Don't add corresponding data in [ext].[CONTOSORETAILTRANSACTIONTABLE] for some of the transactions that you created in MPOS. In this way, you can verify that the data from [ax].RetailTransactionTable is pulled and uploaded even if there is no corresponding data in the extension table.

- c. Go to **Dynamics 365 > Retail and Commerce > Retail and Commerce IT**, and then select **Distribution schedule**.
  - d. In the list of distribution schedules, select **P-0001**. This distribution schedule contains the **RetailTransactionTable** subjob that you customized.
  - e. On the Action Pane, select **Run**. When the confirmation message appears, select **Yes**.
  - f. On the Action Pane, select **History** to open the **History** page, where you can verify that the uploaded session was completed successfully.
  - g. On the **History** page, verify that there is a new upload session record. Also verify that the status of the record is set to **Applied**, and that the **Rows Affected** value isn't **0** (zero).
3. If the upload session is applied successfully, go to **Retail and Commerce > Inquiries and reports > Store transactions**, and search for the new transactions that you just uploaded. Verify that the transactions, seat number, and server staff ID custom columns have the expected values.

Additionally, verify that the transactions that don't have a corresponding record in the

[ext].ContosoRetailTransactionTable extension table on the channel side are also uploaded. Verify that these transactions have default values for the seat number and server staff ID. The seat number should be set to **0** (zero), and the server staff ID should be set to **000160**.

#### **MPOS offline transaction sync**

Switch MPOS to offline mode and do some transaction. Switch back to online and verify that the data is synced properly from the offline database to the channel database and then to headquarters.

#### **NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Define and set order attributes

2/18/2021 • 14 minutes to read • [Edit Online](#)

Previously, the attribute framework supported attributes only in online orders. However, the framework has been extended so that it now supports attributes in cash-and-carry transactions, customer orders, and call center orders. This enhancement lets you edit and set attribute values for orders directly in Commerce Headquarters, the point of sale (POS), and the Commerce runtime (CRT).

Headquarters now includes pages for editing and updating attribute values, which means that you can set the values for call center orders in Headquarters. In POS, use the Attributes panel to set or update the attribute value in POS. If you don't need to use a user interface and just want to add business logic, you can add the business logic directly in CRT. You can create new attributes by using the Headquarters configurations. No database changes are required. Previously, you had to create new tables in Headquarters and the channel database, and then modify those tables.

## Why and when you should order attributes

If you want to add new fields to cash-and-carry transactions, customer orders, or call center orders, and if you want to capture the information in the POS or Headquarters, use order attributes. Previously, to add a new field to a cash-and-carry transaction (transaction header or lines) or a customer order in the POS, you had to create a new extension table in Headquarters and the channel database, and then make inline changes to CRT and POS code to handle the various screens and operations. You also had to configure Commerce Data Exchange to synchronize the data between the channel database and Headquarters. However, order attributes now let you complete all these actions through configuration. You don't have to write any code or create custom extension tables, but you still need to create the core business logic and the POS UI.

This first version supports only the **String** attribute type, but future versions will support other attribute types. If you want the data to come from the master table, and that data involves complex search logic and core business logic in X++, you should use extension properties.

### NOTE

We only support attributes on customer orders and cash-and-carry transactions, no other transaction types are supported.

## Define attribute types

First, you must define the attribute types and assign valid ranges to them.

1. Go to **Product information management > Setup > Categories and attributes > Attribute types**.
2. On the **Attribute types** page, select **New** to add a new attribute type.
3. Enter a name for the attribute type.
4. On the **General** FastTab, in the **Type** field, select the type of data that can be entered for attributes that are assigned to this data type.
5. If the attribute type is **Decimal** or **Integer**, select a unit of measure.
6. If the attribute type is **Text**, you can define a fixed list of values for it. Select the **Fixed list** check box, and then, on the **Values** FastTab, enter the list of values.
7. To define a range of valid values for the attribute type, select the **Value range** check box. Then, on the **Range** FastTab, enter the valid range of values.

## Define the attributes

Next, you must define the attributes. Follow these steps for each attribute that you want to define.

1. Go to **Product information management > Setup > Categories and attributes > Attributes**.
2. On the **Attributes** page, select **New** to add a new attribute.
3. Enter a name, friendly name, and description for the attribute. Additionally, enter any Help text that should be shown to the user for the attribute.
4. In the **Attribute type** field, select the attribute type to assign to the attribute.
5. Depending on the attribute type, in the **Default value** field, enter the value or range of values that is shown by default when the attribute is assigned to a channel.
6. Select **Translate**. Then, on the **Text translation** page, enter the name, description, friendly name, and Help text for the attribute in additional languages.

## Define attribute groups

1. Go to **Product information management > Setup > Categories and attributes > Attribute groups**.
2. On the **Attribute groups** page, select **New** to add a new attribute group.
3. Enter a name for the attribute group. Then, on the **General** FastTab, enter a friendly name, a description, and any Help text for the attribute group.
4. On the **Attributes** FastTab, select **Add** to add attributes to the attribute group. In the **Default value** field, you can enter a default value for the selected attributes.
5. Select **Translate**. Then, on the **Text translation** page, enter the description, friendly name, and Help text for the attribute group in additional languages.

## Link the attribute group to the channel

1. Go to **Retail and Commerce > Channels > Stores > All stores**.
2. Select the channel that the attributes on the **Channel** page should be linked to.
3. On the **Set up** tab, select **Sales order attributes** under **Attribute group**.
4. On the **Sales order attribute groups** page, select **New** to link the attribute group to the channel.
5. In the **Name** field, select the attribute group to link to the channel.
6. In the **Apply attributes to** field, select one of the following options:
  - **Header** – The attributes will apply only to the transaction header.
  - **Lines** – The attribute will apply only to the transaction lines.
  - **Default** – The attribute will apply to both the transaction header and the transaction lines.
7. Select **Save**.

## Run the distribution jobs

1. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Select **Products (1040)**, and then, on the Action Pane, select **Run now**. When you're prompted, select **Yes**. This step is required only if you added any new attributes, attribute types, or attribute groups.
3. Select **Channel configuration job (1070)**, and then, on the Action Pane, select **Run now**. When you're prompted, select **Yes**.

## Show order attributes in the POS transaction screen using the

# Attribute control (this feature is available in version 8.1.3 and later)

## Headquarters

1. Select **Retail and Commerce > Channel setup > POS Setup > POS > Screen layouts**.
2. On the screen layout page, click **New** to create a new screen layout, or select an existing screen layout.
3. Enter the ID and name for the screen layout.
4. On the **Layout sizes** FastTab, select the **Add** button to add new layout sizes for the POS.
5. In the **Name** field, select the POS screen resolution.
6. On the **Layout sizes** FastTab, click the **Layout designer** button.
7. If you're prompted, select **Yes** to download and install the Designer Host by using the **Install/Run** button.
8. When you're prompted, enter the Microsoft Dynamics 365 user name and password to start the designer.
9. After the designer is started, drag the Attributes panel anywhere in the screen layout designer and adjust the size according to your screen width.
10. When you've finished, select **OK** to save your changes.
11. Close the screen layout designer by clicking the **Close** button (X) in the upper-right corner. When you're prompted, select **Yes** to save your changes.
12. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
13. Select the Registers job (1090), and then, on the Action Pane, select **Run now**. When you're prompted, select **Yes**.

## POS

1. Start POS, and add any item to a transaction. You should see the Attribute panel in the transaction screen with the configured attributes both for header and lines.
2. Click the **Edit** icon in the attribute panel to update the attribute value.
3. Click the header or lines tab in the attribute panel to view the header or lines attribute.
4. The lines attribute will refresh automatically based on the lines selected in the transaction.

## Set attribute values for call center orders

After you configure the order attributes for the channel, go to **Customer service** or **All Sales orders**, and create a new call center order.

1. After or during the creation of a customer order, if you want to set an attribute value for the transaction header, on the Action Pane, go to the **Commerce** tab, select **Attributes**.
2. On the **Sales order attributes values** page, you can set the values for the attributes. The list of attributes on this page is based on the attribute group that you configured for the channel.
3. To set attribute values at the line level, on the **Sales order** page, select the **Lines** view, and then select the line to set the attribute value for. Under **Sales order lines group**, select **Retail and Commerce > Attributes**.
4. Repeat step 3 for all the sales lines that you want to set the values for.

## View the attributes values for cash-and-carry transactions in Headquarters

After you've run the distribution job and pulled a cash-and-carry transaction into Headquarters, you can view the attribute values for that transaction. The POS doesn't provide a UI for viewing order attributes. Therefore, to view the order attribute values, you must extend the POS.

1. Go to **Retail and Commerce > Inquires and reports > Store transactions**.
2. To view the transaction header attributes, on the Action Pane, select **Attributes**.
3. To view the transaction lines attributes, on the Action Pane, select **Transaction > Sales transaction**.

4. On the **Sales transactions** page, select any line, and then, on the Action Pane, select **Attributes** to view the line attributes.

#### NOTE

Only the attributes that are configured as part of your attribute group and linked to the channel will appear in the Headquarters UI.

## Extend attributes to add business logic in CRT

A new sample that has been added to the Retail SDK adds business logic for order attributes in CRT. This sample includes code only for the business logic. It doesn't show how to save or read the attributes, because read and write operations for attributes are automated.

The sample implements the following scenario: When you suspend a cart, you set an attribute value. When you resume the cart, you want to clear that value. A pre-trigger was added for **SuspendCartRequest**, and the business logic was written. You can extend any trigger or override any request in CRT to set the logic, based on your scenario.

#### NOTE

Before adding attribute to the cart, check whether the attribute already exists in the cart or cartline. If the attribute already exists, then don't add the attribute again, instead update it. If a duplicate attribute is added to the cart or cartline, then CRT will display a runtime error. Sample code for this scenario can be found in sample code section below.

You can find the full sample code in the Retail SDK at Retail SDK\SampleExtensions\CommerceRuntime\Extensions.TransactionAttributesSample.

- Create a new C# portable class library project, and paste in the following code.

```
public class CustomSuspendCartTrigger : IRequestTrigger
{
    // summary
    // Gets the list of supported request types.
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            return new [ ] { typeof(SuspendCartRequest)};
        }
    }

    // Pre-trigger code.

    // summary
    // param name="request" The request param
    public void OnExecuting(Request request)
    {
        ThrowIf.Null(request, "request");
        Type requestedType = request.GetType();
        if (requestedType == typeof(SuspendCartRequest))
        {
            SuspendCartRequest suspendCartRequest = request as SuspendCartRequest;
            // Get the cart.
            var getCartServiceRequest = new GetCartServiceRequest(
                new CartSearchCriteria(suspendCartRequest.CartId), QueryResultSettings.SingleRecord);
            Cart cart = request.RequestContext.Execute<GetCartServiceResponse>
            (getCartServiceRequest).Carts.Single();
            // Update the transaction header attribute for customer order.
            if (cart.CartType == CartType.CustomerOrder)
```

```

        if (cart.CartType == CartType.CustomerOrder)
        {
            bool cartUpdated = CustomCartHelper.CreateUpdateTransactionHeaderAttribute(
                cart, reserveNow: false, updateAttribute: true);
            if (cartUpdated)
            {
                // Save the cart after updating the header attribute.
                var saveCartRequest = new SaveCartRequest(cart);
                request.RequestContext.Execute<SaveCartResponse>(saveCartRequest);
            }
        }
    }
}

// Post-trigger code.

// summary
// param name="request" The request param
// param name="response" The response param
public void OnExecuted(Request request, Response response)
{
}

// Sample code to check for the duplicate attribute, before adding attributes to the cart check
whether the attribute already exists if so then don't add the attribute again, instead update it.

public static class CustomCartHelper
{
    /// <summary>
    /// Updates the transaction header attribute.
    /// </summary>
    /// <param name="cart">The cart.</param>
    /// <param name="reserveNow">The value of the transaction header attribute.</param>
    /// <param name="updateAttribute">A flag indicating whether or not to override an existing
attribute value.</param>
    /// <returns>A flag indication whether or not the cart was updated.</returns>
    public static bool CreateUpdateTransactionHeaderAttribute(Cart cart, bool reserveNow, bool
updateAttribute)
    {
        ThrowIf.Null(cart, "cart");
        bool cartUpdated = false;
        IList<AttributeValueBase> transactionAttributes = cart.AttributeValues;
        string reserveNowAttributeName = "Reserve now";
        string reserveNowAttributeValue = reserveNow ? "Yes" : "No";
        AttributeValueBase reserveNowAttribute = transactionAttributes.SingleOrDefault(attribute =>
attribute.Name.Equals(reserveNowAttributeName));

        if (reserveNowAttribute == null)
        {
            transactionAttributes.Add(new AttributeTextValue() { Name = reserveNowAttributeName,
TextValue = reserveNowAttributeValue });
            cartUpdated = true;
        }
        else if (updateAttribute && !
((AttributeTextValue)reserveNowAttribute).TextValue.Equals(reserveNowAttributeValue))
        {
            ((AttributeTextValue)reserveNowAttribute).TextValue = reserveNowAttributeValue;
            cartUpdated = true;
        }

        return cartUpdated;
    }
}

```

Extend attributes to do some business logic in the POS

## NOTE

The following changes are required only if you are running the application with version 8.1.2 or earlier. Starting in 8.1.3, you can use the Attributes panel to set or update the attribute value in POS. With this control you no longer need to write any additional code or create UI to set the attribute value in POS. In the attribute control UI has been added to set or update the attribute value. Refer to the **Show Order attributes in the POS transaction screen using the Attribute control** section in this document for more details.

A new sample that has been added to the Retail SDK sets the business logic for order attributes in the POS. This sample includes code only for the business logic. It doesn't show how to save or read attribute values, because read and write operations for attributes are automated. You can set the values for attributes in either CRT or the POS, based on your scenario. If your values are based on customer input, set them in the POS client. If some business logic is involved, set the values in CRT.

The sample implements the following scenario: You create a business-to-business (B2B) order and want to set the B2B attribute value (**Yes** or **No**), based on customer input. **PreEndTransactionTrigger** was extended in the POS to set the values. You can extend any POS trigger or override the request as appropriate.

You can find the full sample code in the Retail SDK at Retail SDK\POS\Extensions\B2BSample.

## NOTE

Only the configured attributes will appear in the Headquarters UI, even if you set or add attributes and attribute values in the code. If an attribute isn't part of the attribute group that you linked to the channel, it won't appear in the Headquarters UI.

1. From the Retail SDK, open **ModernPOS.sln\CloudPos.sln**.
2. In the Retail SDK, create a new extension folder under the **POS.Extension** project.
3. In the new extension folder, add a new **manifest.json** file.
4. Paste in the following code.

```
{
  "$schema": "..manifestSchema.json",
  "name": "Pos_Extensibility_B2BSample",
  "publisher": "Microsoft",
  "version": "7.2.0",
  "minimumPosVersion": "7.2.0.0",
  "components": {
    "resources": {
      "supportedUICultures": [ "en-US" ],
      "fallbackUICulture": "en-US",
      "culturesDirectoryPath": "Resources/Strings",
      "stringResourcesFileName": "resources.resjson"
    },
    "extend": {
      "triggers": [
        {
          "triggerType": "PreEndTransaction",
          "modulePath": "TriggerHandlers/PreEndTransactionTrigger"
        }
      ]
    }
  }
}
```

5. Create a new TypeScript file to implement **PreEndTransactionTrigger**, and add the following code.

```
...
```

```

/**
 * Executes the trigger functionality.
 * @param {Triggers.IPreEndTransactionTriggerOptions} options The options provided to the trigger.
 */

public execute(options: Triggers.IPreEndTransactionTriggerOptions):
Promise<ClientEntities.ICancelable {
    console.log("Executing PreEndTransactionTrigger with options " + JSON.stringify(options) + ".");
    let currentCart: ProxyEntities.Cart;
    return this.context.runtime.executeAsync<GetCurrentCartClientResponse>(new
GetCurrentCartClientRequest())
then((getCurrentCartClientResponse:
ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>):
    Promise<ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>> => {
        currentCart = getCurrentCartClientResponse.data.result;
        // Gets the current customer.
        let result: Promise<ClientEntities.ICancelableDataResult<GetCurrentCartClientResponse>>;
        if (!ObjectExtensions.isNullOrUndefined(currentCart) &&
!ObjectExtensions.isNullOrUndefined(currentCart.CustomerId)) {
            let getCurrentCustomerClientRequest: GetCustomerClientRequest<GetCustomerClientResponse =
new GetCustomerClientRequest(currentCart.CustomerId);
            result = this.context.runtime.executeAsync<GetCustomerClientResponse>
(getCurrentCustomerClientRequest);
        } else {
            result = Promise.resolve({ canceled: false, data: new GetCustomerClientResponse(null) });
        }
        return result;
    })
.then((getCurrentCustomerClientResponse:
ClientEntities.ICancelableDataResult<GetCustomerClientResponse>):
    Promise<ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>> => {
        let currentCustomer: ProxyEntities.Customer = getCurrentCustomerClientResponse.data.result;
        // If the cart is a customer order with a B2B customer, then we display a dialog to determine if
the order should be B2B.
        let result: Promise<ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>>;
        if (!ObjectExtensions.isNullOrUndefined(currentCart)
&& currentCart.CustomerOrderModeValue ===
ProxyEntities.CustomerOrderMode.CustomerOrderCreateOrEdit
&& !ObjectExtensions.isNullOrUndefined(currentCustomer)
&& this.isCustomerB2B(currentCustomer)) {
            let yesButton: ClientEntities.Dialogs.IDialogResultButton = {
                id: PreEndTransactionTrigger.DIALOG_YES_BUTTON_ID,
                label: this.context.resources.getString("string_0"), "Yes"
                result: PreEndTransactionTrigger.DIALOG_RESULT_YES
            };
            let noButton: ClientEntities.Dialogs.IDialogResultButton = {
                id: PreEndTransactionTrigger.DIALOG_NO_BUTTON_ID,
                label: this.context.resources.getString("string_1"), "No"
                result: PreEndTransactionTrigger.DIALOG_RESULT_NO
            };
            let showMessageDialogClientRequestOptions: ClientEntities.Dialogs.IMessageDialogOptions = {
                title: this.context.resources.getString("string_2"), "B2B Order"
                subTitle: StringExtensions.EMPTY,
                message: this.context.resources.getString("string_3"), "Do you want to mark this order
as a B2B order?"
                button1: yesButton,
                button2: noButton
            };
            let showMessageDialogClientRequest:
ShowMessageDialogClientRequest<ShowMessageDialogClientResponse> =
new ShowMessageDialogClientRequest(showMessageDialogClientRequestOptions);
            result = this.context.runtime.executeAsync<ShowMessageDialogClientResponse>
(showMessageDialogClientRequest);
        } else {
            result = Promise.resolve({ canceled: false, data: new ShowMessageDialogClientResponse(null)
});
        }
        return result;
    })
});

```

```

.then((showMessageDialogClientResponse:
ClientEntities.ICancelableDataResult<ShowMessageDialogClientResponse>):
    Promise<ClientEntities.ICancelableDataResult<SaveAttributesOnCartClientResponse>> => {
        // Save the B2B attribute value depending on the dialog result.
        let messageDialogResult: ClientEntities.Dialogs.IMessageDialogResult =
showMessageDialogClientResponse.data.result;
        let result: Promise<ClientEntities.ICancelableDataResult<SaveAttributesOnCartClientResponse>>;
        if (!ObjectExtensions.isNullOrUndefined(messageDialogResult)) {
            let attributeValue: ProxyEntities.AttributeTextValue = new
ProxyEntities.AttributeTextValueClass();
            attributeValue.Name = PreEndTransactionTrigger.B2B_CART_ATTRIBUTE_NAME;
            attributeValue.TextValue = messageDialogResult.dialogResult ===
PreEndTransactionTrigger.DIALOG_RESULT_YES?
            PreEndTransactionTrigger.B2B_ATTRIBUTE_VALUE_TRUE :
            PreEndTransactionTrigger.B2B_ATTRIBUTE_VALUE_FALSE;
            let attributeValues: ProxyEntities.AttributeValueBase[] = [attributeValue];
            let saveAttributesOnCartRequest:
SaveAttributesOnCartClientRequest<SaveAttributesOnCartClientResponse> =
                new SaveAttributesOnCartClientRequest(attributeValues);
            result = this.context.runtime.executeAsync(saveAttributesOnCartRequest);
        } else {
            result = Promise.resolve({ canceled: false, data: new
SaveAttributesOnCartClientResponse(null) });
        }
        return result;
    });
}
/**
 * Returns whether or not the given customer is a B2B customer.
 * @param {ProxyEntities.Customer} customer The customer.
 * @returns {boolean} Whether or not the given customer is a B2B customer.
 */
private isCustomerB2B(customer: ProxyEntities.Customer): boolean {
    let isB2B: boolean = false;
    if (!ObjectExtensions.isNullOrUndefined(customer.Attributes)) {
        for (let i: number = 0; i < customer.Attributes.length; i++) {
            let currentAttribute: ProxyEntities.CustomerAttribute = customer.Attributes[i];
            if (currentAttribute.Name === PreEndTransactionTrigger.B2B_CUSTOMER_ATTRIBUTE_NAME) {
                if
(!ObjectExtensions.isNullOrUndefined(currentAttribute.AttributeValue.BooleanValue)) {
                    isB2B = currentAttribute.AttributeValue.BooleanValue;
                }
                break;
            }
        }
    }
    return isB2B;
} } } }

```

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Customer attributes

2/18/2021 • 5 minutes to read • [Edit Online](#)

We extended the attribute framework in Headquarters to support attributes for customers, customer orders, cash-and-carry transactions, and call center orders.

## NOTE

The attributes are read-only. However, in the case of customer or order attributes, you can edit and set values at the level of the individual customer or order.

The new customer attribute framework lets you use configurations to add new fields to the customer master record. Those fields then automatically appear on the **Customer add/edit** or **Customer details** screen in Point of Sale (POS) or Headquarters. After you configure the customer attribute group in the Commerce parameters, POS and Headquarters automatically show the new attribute. No code change or customization is required. You can also use the screen layout designer to configure the customer card on the POS transaction screen so that it shows the customer attributes.

## Why and when you should configure customer attributes

If you want to add new fields to the customer master record, and capture the information in POS or Headquarters, you can use this feature. Previously, to add a new field to the customer master record and show it in POS and Headquarters, you had to create a new extension table in Headquarters and the channel database, and make inline modifications to Commerce runtime (CRT) and POS code. You had to write code in CRT and POS to read/write to the extension fields and show them in POS. You had to handle this in various POS views and scenarios, such as the **Customer details** screen and the **Customer** panel on the transaction screen. In addition, in CRT, you had to handle all insert, select, and update operations. However, the new functionality lets you complete all these steps through configuration so you don't have to write any code or create custom extension tables.

The first version of this functionality doesn't support **datetime** and **reference** attribute types. For those attribute types, you should use extension properties and custom controls to show the details in POS.

## Configure customer attributes in POS and Headquarters

### Define attribute types

1. Select **Product information management > Setup > Categories and attributes > Attribute types**.
2. On the **Attribute types** page, select **New** to add a new attribute type.
3. Enter a name for the attribute type.
4. On the **General** FastTab, in the **Type** field, select the type of data that can be entered for attributes that are assigned to this data type.
5. If the attribute type is **Decimal** or **Integer**, select a unit of measure.
6. To define a fixed list of values for the attribute type, select the **Fixed list** check box. Then, on the **Values** FastTab, add the list of values.

**NOTE**

The **Fixed list** check box is available only for the **Text** attribute type.

Alternatively, to define a range of valid values for the attribute type, select the **Value range** check box. Then, on the **Range** FastTab, enter the valid range of values.

**Define attributes**

1. Select **Product information management > Setup > Categories and attributes > Attributes**.
2. On the **Attributes** page, select **New** to add a new attribute.
3. Enter the name, friendly name, description, and any Help text that should be shown to the user for the attribute.
4. In the **Attribute type** field, select the attribute type to assign to the attribute.
5. Depending on the attribute type, in the **Default value** field, enter the value or the range of values that is shown by default when this attribute is assigned to a customer.
6. Select **Translate** to open the **Text translation** page, where you can enter the name, description, friendly name, and Help text for the attribute in additional languages.
7. Repeat steps 2 through 6 to add more attributes.

**Define an attribute group**

1. Select **Product information management > Setup > Categories and attributes > Attribute groups**.
2. On the **Attribute groups** page, select **New** to add a new attribute group.
3. Enter the name, and then, on the **General** FastTab, enter the friendly name, description, and any Help text for the attribute group.
4. On the **Attributes** FastTab, select **Add** to add attributes to the attribute group. In the **Default value** field, you can enter a default value for the selected attributes.
5. Select **Translate** to open the **Text translation** page, where you can enter the description, friendly name, and Help text for the attribute group in additional languages.

**Link the attribute group to the customers**

1. Select **Retail and Commerce > Headquarters setup > Parameters > Commerce parameters**.
2. On the **General** tab, in the **Customer attribute group** field, select the attribute group that should be shown in POS.

**Run the distribution jobs**

1. Select **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
2. Select the **Customers job (1010)**, and then, on the Action Pane, select **Run now**. When you're prompted, select **Yes**.
3. Select the **Global configuration job (1110)**, and then, on the Action Pane, select **Run now**. When you're prompted, select **Yes**.

**View customer attributes****Headquarters**

1. Select **Retail and Commerce > Customers > All customers**.
2. On the Action Pane, in **Retail and Commerce**, in the **Attribute** section, select **Retail attributes** to view or edit the attribute values.

**POS**

- Start POS, and then either open the **Customer Add/Edit** screen to set or update the attribute values for the customer, or open the **Customer details** screen to view the configured attributes.

**Show customer attributes in the POS transaction screen**

## Headquarters

1. Select **Retail and Commerce > Channel setup > POS Setup > POS > Screen layouts**.
2. On the **screen layout** page, select **New** to create a new screen layout, or select an existing screen layout.
3. Enter the ID and name for the screen layout.
4. On the **Layout sizes** FastTab, select the **Add** button to add new layout sizes for the POS.
5. In the **Name** field, select the POS screen resolution.
6. On the **Layout sized** FastTab, select the **Layout designer** button.
7. If you're prompted, select **Yes** to download and install the Retail Designer Host by using the **Install/Run** button.
8. When you're prompted, enter the Microsoft Dynamics 365 user name and password to start the designer.
9. After the designer is started, drag the **Customer** card anywhere in the screen layout designer.
10. Right-click the **Customer** card, and then select **Customize**.
11. When the page for the **Customization - Customer** card appears, select the required attributes in the **Available columns** section, and then select the right arrow button (>) to move them to the **Selected columns** section. You can move the attributes up or down by selecting the **Up** or **Down** buttons.
12. When you've finished, select **OK** to save your changes.
13. Close the screen layout designer by selecting the **Close** button (X) in the upper-right corner. When you're prompted, select **Yes** to save your changes.
14. Select **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
15. Select the **Registers** job (1090), and then, on the Action Pane, select **Run now**. When you're prompted, select **Yes**.

## POS

1. Start POS, and add a customer to a transaction.
2. Open the transaction screen to view the attributes that have been added.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Retail software development kit (SDK) samples

2/18/2021 • 9 minutes to read • [Edit Online](#)

This topic describes the samples that are included in the Retail SDK.

## Override message handler sample

**Scenario:** Sometimes, one of Fabrikam's customers is in the customer relationship management (CRM) system but isn't imported into Microsoft Dynamics 365 Commerce. Therefore, Fabrikam wants to look up the customer from the CRM system and the point of sale (POS). Here are the business requirements:

- Search for customers from the CRM system and the POS.
- Merge the results, and show a unified result set in Retail Modern POS (MPOS).

Here are some situations where you might use the override message handler:

- You want to use a third-party inventory system for stock updates and inquiries.
- You want to integrate with an external tax system for tax calculation.
- You want to integrate with a third-party loyalty system.

Here are the basic tasks in the sample:

1. Override and implement the existing customer search request, because we are changing the existing search behavior so that an external search is performed.
2. After the external search is completed, call the standard search request, and merge both results.

Here is the code for these tasks.

```
public sealed class CustomerSearchRequestHandler : SingleRequestHandler<CustomersSearchRequest,
CustomersSearchResponse>
{
    /// <summary>
    /// Executes the workflow to retrieve customer information.
    /// </summary>
    /// <param name="request">The request.</param>
    /// <returns>The response.</returns>
    protected override CustomersSearchResponse Process(CustomersSearchRequest request)
    {
        ThrowIf.Null(request, "request");
        ThrowIf.Null(request.Criteria, "request.Criteria");
        // Execute custom customer search logic here.
        CustomersSearchResponse externalResponse = this.ExternalCustomerSearch(request.Criteria.Keyword);
        // Execute original customer search logic.
        var requestHandler = new
Microsoft.Dynamics.Commerce.Runtime.Workflow.CustomerSearchRequestHandler();
        CustomersSearchResponse originalResponse =
            request.RequestContext.Runtime.Execute<CustomersSearchResponse>(request, request.RequestContext,
requestHandler, skipRequestTriggers: false);
        return new
CustomersSearchResponse(externalResponse.Customers.Union(originalResponse.Customers).AsPagedResult());
    }
}
```

The full sample code is in the RetailSDK\SampleExtensions\CommerceRuntime\Extensions.CustomerSearchSample folder of the software development kit (SDK).

## Best practice

If you're planning to completely change the behavior of an existing request or response, or if you want to use your logic in addition to the standard logic, override the standard message handler.

# Request handler triggers and extension properties sample

**Scenario:** Fabrikam wants to collect customer email preferences for email marketing. Here are the business requirements:

- Enable a customer's email preferences to be collected and updated from the POS.
- A customer's email preferences should become effective immediately.

Here are some situations where you might use extension properties:

- You want to extend entities such as the customer and sales order, but you don't want to create a new separate entity.
- As new entity fields are read from or written to the database, they should be sent between the commerce runtime (CRT) and the POS, and updated in the client.
- You want temporary internal flags that can be used to control the flow of custom logic.
- You want to set custom receipt fields that the receipt customization will access when receipts are generated.

The following steps show the CRT code changes. For MPOS and the channel database, see the full sample. Notice that the following samples differ from previous code, where changes to the standard database artifacts were required. (For example, to expose new columns as extension properties, changes to the view were required. To receive a list of extension properties and update these properties together with standard fields, changes to the stored procedure were required.) Eventually, as we move to a model that doesn't have inline changes, merge conflicts should not occur even when the database is updated. Therefore, our new recommendation is that you make separate database calls to read, write, and update entities.

1. **Read the entity.** Implement the post-trigger for `GetCustomerDataRequest`, read the value from channel database, and add the value to the extension property.

```

public class GetCustomerTriggers : IRequestTrigger
{
    public IEnumerable<Type> SupportedRequestTypes
    {
        get { return new[] { typeof(GetCustomerDataRequest) }; }
    }
    public void OnExecuted(Request request, Response response)
    {
        // Check if default handler found a customer.
        var customer = ((SingleEntityDataServiceResponse<Customer>)response).Entity;
        if (customer == null)
        {
            return;
        }
        // Read from a custom view mapped to a custom table.
        var query = new SqlPagedQuery(QueryResultSettings.SingleRecord)
        {
            Select = new ColumnSet(new string[] { "EMAILOPTIN" }),
            From = "CUSTOMEREXTENSIONVIEW",
            Where = "ACCOUNTNUM = @accountNum AND DATAAREAID = @dataAreaId"
        };
        query.Parameters["@accountNum"] = customer.AccountNumber;
        query.Parameters["@dataAreaId"] =
request.RequestContext.GetChannelConfiguration().InventLocationDataAreaId;
        using (var databaseContext = new SqlServerDatabaseContext(request))
        {
            // Use ExtensionEntity which will map all columns to extension properties.
            ExtensionsEntity extensions = databaseContext.ReadEntity<ExtensionsEntity>
(query).FirstOrDefault();
            var emailOptIn = extensions != null ? extensions.GetProperty("EMAILOPTIN") : null;
            // If the EmailOptIn is found, set it at a new extension property at the Customer.
            if (emailOptIn != null)
            {
                customer.SetProperty("EMAILOPTIN", emailOptIn);
            }
        }
    }
}

```

2. **Write the entity.** Override the handler for `CreateOrUpdateCustomerDataRequest` to run the original request handler and the custom stored procedure inside a transaction scope. If the database transaction isn't required, a post-trigger suffices here.

```

protected override SingleEntityDataServiceResponse<Customer>
Process(CreateOrUpdateCustomerDataRequest request)
{
    using (var databaseContext = new SqlServerDatabaseContext(request))
        using (var transactionScope = new TransactionScope())
            {
                // Execute original functionality to save the customer.
                var requestHandler = new
Microsoft.Dynamics.Commerce.Runtime.DataServices.SqlServer.CustomerSqlServerDataService();
                var response = (SingleEntityDataServiceResponse<Customer>)requestHandler.Execute(request);
                // Execute additional functionality to save the customer's extension properties.
                if (!request.Customer.ExtensionProperties.IsNullOrEmpty())
                    {
                        // The stored procedure will determine which extension properties are saved to which
tables.
                        ParameterSet parameters = new ParameterSet();
                        parameters["@TVP_EXTENSIONPROPERTIESTABLETYPE"] = new
ExtensionPropertiesTableType(request.Customer.RecordId,
request.Customer.ExtensionProperties).DataTable();
                        databaseContext.ExecuteStoredProcedureNonQuery("UPDATECUSTOMEREXTENSIONPROPERTIES",
parameters);
                    }
                transactionScope.Complete();
                return response;
            }
}

```

Before you try this sample, be sure to create the custom tables, views, and stored procedures in the channel database. Additionally, make the relevant changes to MPOS. The full sample code, together with additional comments, is in the RetailSDK\SampleExtensions\CommerceRuntime\Extensions.EmailPreferenceSample folder of the SDK. For information about how to create custom database artifacts, see the RetailSDK\Documents\SampleExtensionsInstructions\EmailPreference folder of the SDK.

#### NOTE

The above code sample and the sample script in the RetailSDK\Documents\SampleExtensionsInstructions\EmailPreference folder use [crt].EXTENSIONPROPERTIESTABLETYPE. Starting in version 7.3 we no longer support using crt or ax schema objects/data types in ext schema. You must create your custom extension table property type in ext schema and use it.

#### Best practice

Because the order of triggers isn't guaranteed when the triggers are chained, and because of the internal cache mechanism, the pre-triggers should not change the *request* message, and the post-triggers should not change the *response* message. Extension properties are allowed, because no core properties are being changed. You should use pre-triggers and post-triggers to handle extension properties. You should also use pre-triggers to do validation and post-triggers to do additional actions.

## Custom fields and custom receipt types sample

**Scenario:** Fabrikam wants to print a special receipt whenever products that have a warranty are sold. Sales receipts should include the warranty expiration date, the warranty ID, and other information. Here are the business requirements:

- Print special receipts.
- Print additional warranty information on sale receipts.

The following steps shows the HQ configuration and CRT code changes for the custom fields:

# Configure Headquarters for custom fields

At the headquarters (HQ), create two custom receipt fields: **EXPIRATIONDATE** for the warranty expiration date and **WARRANTYID** for the warranty ID. Add these fields to the receipt format layout.

1. Sign in to HQ.
2. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Language text**.
3. On the **POS** tab, select **Add** to add a new POS language text.

The text that appears in the **Receipt** panel can be localized. Therefore, you can create multiple texts in different languages for the same text ID. Here is an example.

LANGUAGE ID	TEXT ID	TEXT
en-US	1	WARRANTYID
en-UK	1	WARRANTYID

4. On the Action Pane, select **Save** to save your changes.
5. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Custom fields**.
6. On the Action Pane, select **New** to add a new custom field, and specify the following information:
  - a. In the **Name** field, enter the name of the custom field.
  - b. In the **Type** field, select **Receipt**.
  - c. In the **Caption text ID** field, specify the text ID that you used in step 3.

Here is an example.

NAME	TYPE	CAPTION TEXT ID
WARRANTYID	Receipt	1

7. On the Action Pane, select **Save** to save your changes.
8. Go to **Retail and Commerce > Channel setup > POS setup > POS > Receipt formats**.
9. Select an existing or create a new receipt format and then select **Designer** on the Action Pane.
10. If you're prompted to confirm that you want to open the application, select **Open**, and then follow the installation instructions.
11. After the designer is installed, you're asked for Azure Active Directory (Azure AD) credentials. Enter the information to start the designer.
12. In the designer, drag and drop the Custom field from the left pane to the receipt designer.
13. Save the changes.
14. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
15. Select the **Channel configuration (1070)** job, and then select **Run now**.

## Sample code to implement Custom fields

To add the custom fields to the sales receipts or any receipt format, implement `GetSalesTransactionCustomReceiptFieldServiceRequest` and the business logic for the custom fields in



CRT, as shown in the following code.

```
public IEnumerable<Type> SupportedRequestTypes
{
    get
    {
        return new[] { typeof(GetSalesTransactionCustomReceiptFieldServiceRequest) };
    }
}
public Response Execute(Request request)
{
    Type requestedType = request.GetType();
    if (requestedType == typeof(GetSalesTransactionCustomReceiptFieldServiceRequest))
    {
        return this.GetCustomReceiptFieldForSalesTransactionReceipts(
(GetSalesTransactionCustomReceiptFieldServiceRequest)request);
    }
    throw new NotSupportedException(string.Format("Request '{0}' is not supported.", request.GetType()));
}
```

### Add the business logic for the custom fields

```
private GetCustomReceiptFieldServiceResponse GetCustomReceiptFieldForSalesTransactionReceipts(
GetSalesTransactionCustomReceiptFieldServiceRequest request)
{
    string receiptFieldName = request.CustomReceiptField;
    string returnValue = null;
    switch (receiptFieldName)
    {
        case "WARRANTYID":
            {
                // Write your logic
            }
            break;
        case "EXPIRATIONDATE":
            {
                // Write your logic
            }
            break;
    }
    return new GetCustomReceiptFieldServiceResponse(returnValue);
}
```

### Custom receipt type configuration in HQ

1. Go to **Retail and Commerce > Channel setup > POS setup > POS > Receipt formats**.
2. Create new Receipt format and then select the **Receipt type** and choose one of the CustomReceiptType(1...20).
3. Save the changes.
4. Select **Designer** on the Action Pane.
5. If you're prompted to confirm that you want to open the application, select **Open**, and then follow the installation instructions.
6. After the designer is installed, you're asked for Azure Active Directory (Azure AD) credentials. Enter the information to start the designer.
7. In the designer, drag and drop the required receipt fields from the left pane to the receipt designer.
8. Save the changes.
9. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
10. Select the **Channel configuration (1070) job**, and then select **Run now**.

### Sample code to implement Custom receipt type

To add logic for the new receipt type, implement `GetCustomReceiptsRequest` in CRT.

```
protected override GetReceiptResponse Process(GetCustomReceiptsRequest request)
{
    Collection<Receipt> result = new Collection<Receipt>();
    // 2. Now we can handle any additional receipt here.
    switch (request.ReceiptRetrievalCriteria.ReceiptType)
    {
        // An example of getting custom receipts.
        case ReceiptType.CustomReceipt1:
            {
                IEnumerable<Receipt> customReceipts = this.GetCustomReceipts(salesOrder,
request.ReceiptRetrievalCriteria);
                result.AddRange(customReceipts);
            }
            break;
        default:
            // Add more logic to handle more types of custom receipt types.
            break;
    }
    return new GetReceiptResponse(new ReadOnlyCollection<Receipt>(result));
}
```

The full sample code is available in the `RetailSDK\SampleExtensions\CommerceRuntime\Extensions.ReceiptsSample` folder.

#### NOTE

You should call the printing of the custom receipt type from the client. For more information, see [Printing custom receipt from POS](#).

#### Best practice

Avoid making database calls for each custom receipt field. Instead, use extension properties that were previously set on entities. Custom receipt types can be called by any logic (per sales line, one time per some condition). See the sample for a more complete scenario.

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Log extension events to Application Insights

2/18/2021 • 6 minutes to read • [Edit Online](#)

This topic explains how to log events to [Customer Application Insights](#) from Commerce runtime (CRT) and POS extensions.

## Log an event to Application Insights

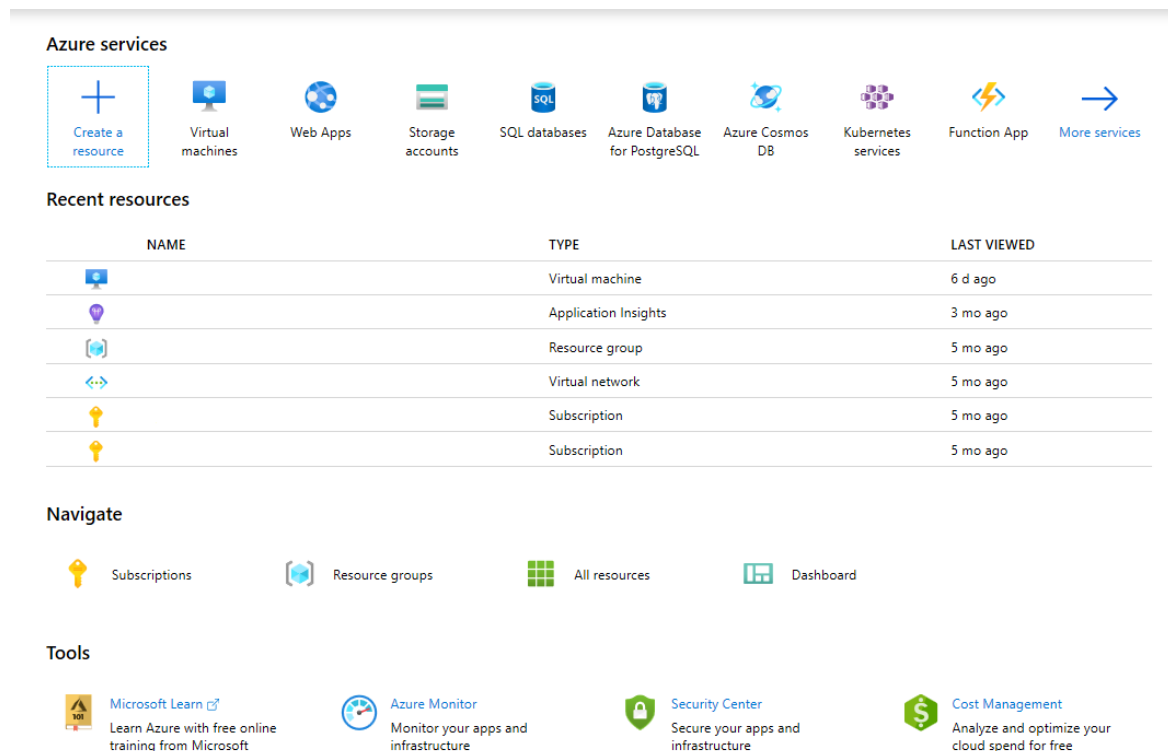
1. Set up Application Insights in the [Microsoft Azure portal](#), and generate the instrumentation key.
2. Extend CRT to log events to Application Insights by using the instrumentation key that you generated.

### NOTE

The `RetailLogger` class is no longer supported. Existing extensions that use this class must be migrated to the new model.

## Set up and configure Application Insights in Azure

1. Open the [Azure portal](#), and sign in by using your Azure subscription credentials.
2. Select **Create a resource**.



**Azure services**

- Create a resource
- Virtual machines
- Web Apps
- Storage accounts
- SQL databases
- Azure Database for PostgreSQL
- Azure Cosmos DB
- Kubernetes services
- Function App
- More services

**Recent resources**

NAME	TYPE	LAST VIEWED
	Virtual machine	6 d ago
	Application Insights	3 mo ago
	Resource group	5 mo ago
	Virtual network	5 mo ago
	Subscription	5 mo ago
	Subscription	5 mo ago

**Navigate**

- Subscriptions
- Resource groups
- All resources
- Dashboard

**Tools**

- [Microsoft Learn](#)  
Learn Azure with free online training from Microsoft
- [Azure Monitor](#)  
Monitor your apps and infrastructure
- [Security Center](#)  
Secure your apps and infrastructure
- [Cost Management](#)  
Analyze and optimize your cloud spend for free

3. Search for **Application Insights**.

Microsoft Azure (Preview) [Report a bug](#)

Home > New

## New

Application Insights

Azure Marketplace [See all](#)

Get started

Recently created

AI + Machine Learning

Analytics

Blockchain

Compute

Containers

Popular

Windows Server 2016 Datacenter  
[Quickstart tutorial](#)

Ubuntu Server 18.04 LTS  
[Learn more](#)

Web App  
[Quickstart tutorial](#)

SQL Database

4. Click **Create** in the Application Insights page.

5. On the **Basic** tab, set the **Subscription**, **Resource group**, **Name**, and **Region** fields.

Home > New > Application Insights > Application Insights

### Application Insights

Monitor web app performance and usage

[Basics](#) [Tags](#) [Review + create](#)

Create an Application Insights resource to monitor your live web application. With Application Insights, you have full observability into your application across all components and dependencies of your complex distributed architecture. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and Java EE, hosted on-premises, hybrid, or any public cloud. [Learn More](#)

**PROJECT DETAILS**

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Resource Group \* ⓘ  [Create new](#)

**INSTANCE DETAILS**

Name \* ⓘ  ✓

Region \* ⓘ

6. On the **Review + create** tab, select **Create**.


## Application Insights

Monitor web app performance and usage

✓ Validation passed

Basics Tags Review + create

### SUMMARY

 **Application Insights**  
by Microsoft

#### Subscription

Resource Group	RetailAppInsights
Name	RetailAppInsights
Region	(US) West US 2

Create

« Previous

[Download a template for automation](#)

7. Wait for the deployment to be completed.

### Microsoft.AppInsights - Overview

Deployment

Search (Ctrl+/)

Delete Cancel Redeploy Refresh

Overview

Inputs

Outputs

Template

✓ Your deployment is complete

Deployment name: Microsoft.AppInsights  
Subscription:  
Resource group: RetailAppInsights

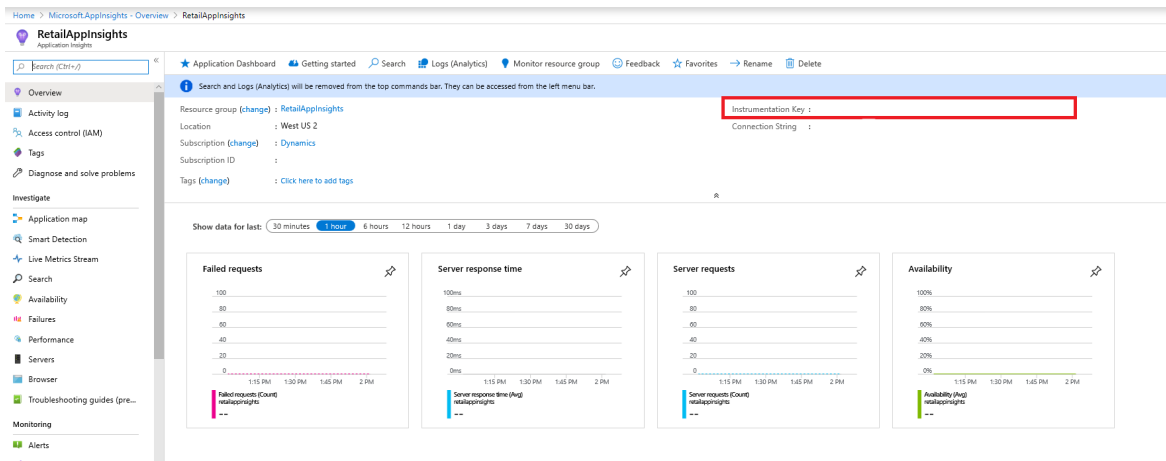
Start time: 11/13/2019, 1:59:55 PM  
Correlation ID:

Deployment details (Download)

Next steps

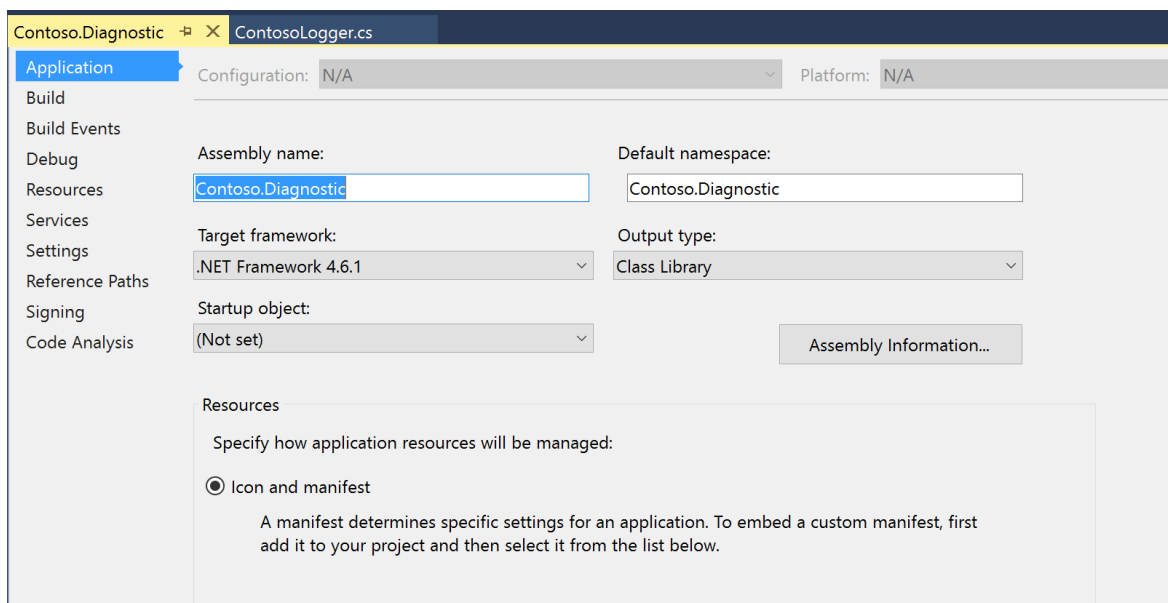
Go to resource

8. Go to the resource, and copy the **Instrumentation Key** value. You will use this value in the CRT code or CRT extension configuration file.



## Extend the CRT extension project to log events to Application Insights

1. Create a new C# class library project, and name it Contoso.Diagnostic.



2. Add references to the following libraries:

- Microsoft.ApplicationInsights
- Netstandard
- Microsoft.Dynamics.Commerce.Runtime.Framework

### NOTE

To install the **Microsoft.ApplicationInsights** assembly reference, install the [Application Insights SDK for ASP.NET Core](#). The reference to **Microsoft.Dynamics.Commerce.Runtime.Framework** can be added from the `..\RetailSDK\Reference` folder.

3. Add a new class file that is named **ContosoLogger**, and copy the following code into it.

```

using Microsoft.ApplicationInsights;
using Microsoft.ApplicationInsights.Extensibility;
using Microsoft.Dynamics.Commerce.Runtime;
using Microsoft.Dynamics.Commerce.Runtime.Extensions;
namespace Contoso.Diagnostic
{
    public static class ContosoLogger
    {
        private static readonly object lockObject = new object();
        private static TelemetryClient client = null;
        public static TelemetryClient GetLogger(RequestContext context)
        {
            if (client == null)
            {
                lock (lockObject)
                {
                    if (client == null)
                    {
                        string key =
context.Runtime.Configuration.GetSettingValue("ext.AppInsightsKey") ?? string.Empty;
                        client = new TelemetryClient(new TelemetryConfiguration(key));
                    }
                }
            }
            return client;
        }
    }
}

```

4. Build the project, and copy the output library and the **Microsoft.ApplicationInsights.dll** file to the **..\RetailServer\webroot\bin\Ext** folder for manual deployment and testing.
5. In the **..\RetailServer\webroot\bin\Ext** folder, open the **CommerceRuntime.Ext.config** file, and update the **<settings>** section with the Applications Insights instrumentation key that you generated earlier. Here is an example.

```
<add name="ext.AppInsightsKey" value="xxxxxxx"/>
```

6. Restart your Commerce Scale Unit.

## Consume the logger in the CRT extension

1. To consume the **ContosoLogger** in the extension, add the **ContosoDiagnostic** and **Microsoft.ApplicationInsights** assembly references to the extension project.
2. To log events, use the **TraceTelemetry** class, and create the traces. Here is an example.

```

using Contoso.Diagnostic;
using Microsoft.ApplicationInsights.DataContracts;
var trace = new TraceTelemetry("CRT executing request", SeverityLevel.Information);
trace.Properties.Add("CustomDimensionColumn1", request.RequestContext.GetTerminalId().ToString());
trace.Properties.Add("CustomDimensionColumn2", "CRT demo - Save Cart request");
ContosoLogger.GetLogger(request.RequestContext).TrackTrace(trace);

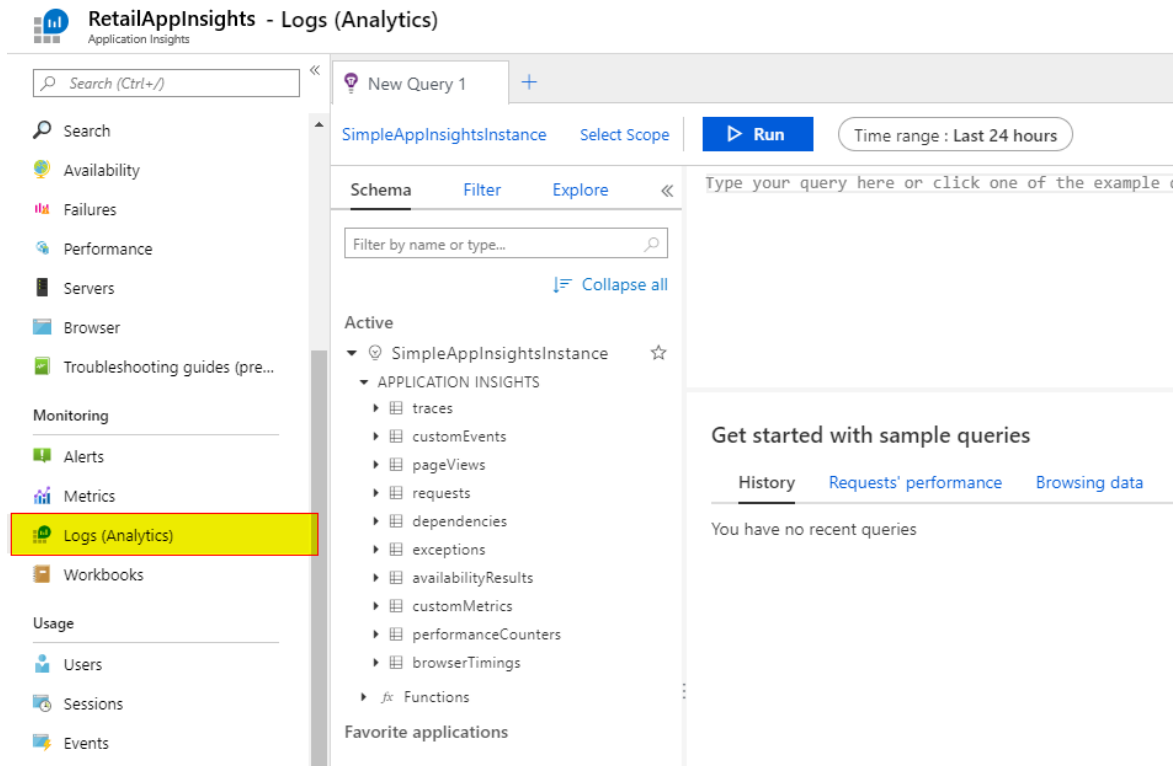
```

### NOTE

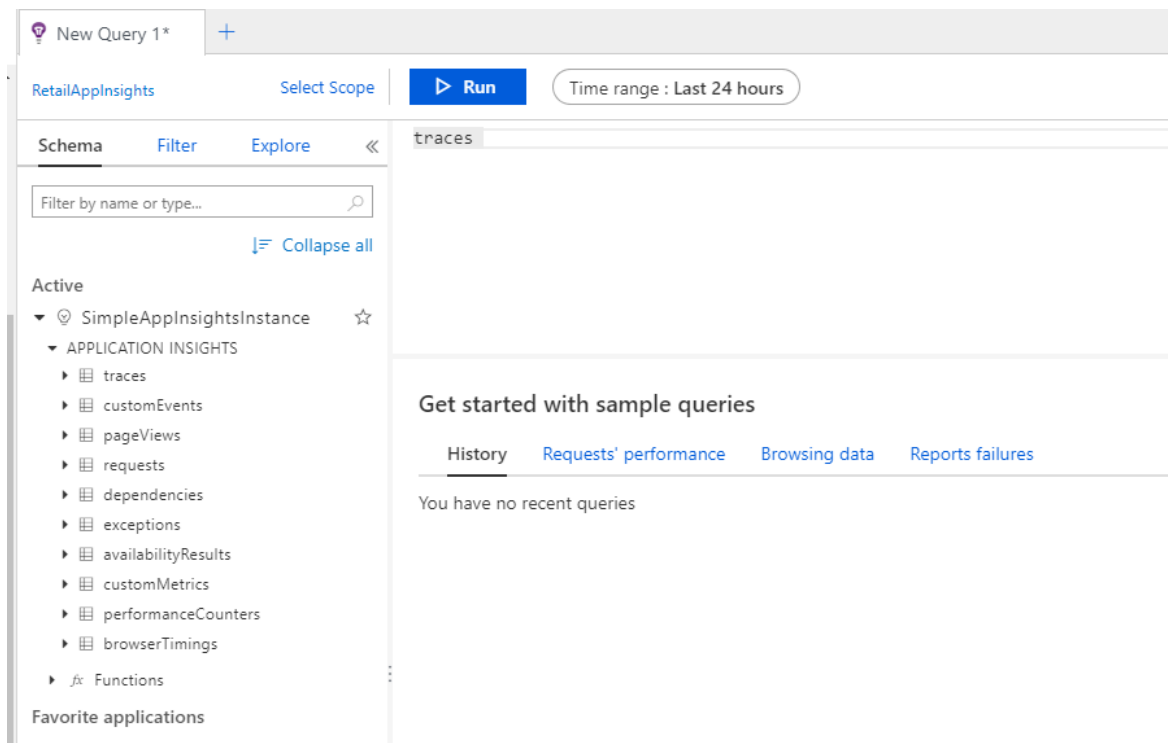
Trace properties are custom dimensions that you can easily add to query the traces.

# Validate the trace events

1. Open the [Azure portal](#), and sign in by using your Azure subscription credentials.
2. Go to the Application Insights instance, and then, under **Monitoring**, select **Logs (Analytics)** to open a new query editor.



3. On the **Schema** tab, double-click **traces** to add it to the query editor. The default time range is **Last 24 hours**.



4. Select **Run** to run the query. The logged event will appear in the results.



Completed. Showing results from the last 24 hours.

TABLE | CHART | Columns ▾

Drag a column header and drop it here to group by that column

timestamp [UTC]	message	severityLevel	itemType	customDimensions	customMeasurements	operation_Name
> 10/24/2019, 8:19:47.033 PM	SaveCartRequest OnExecuting	1	trace	{"TerminalID":"HOUSTON-17"}		
> 10/24/2019, 8:19:54.573 PM	SaveCartRequest OnExecuted	1	trace	{"TerminalID":"HOUSTON-17"}		
> 10/24/2019, 8:17:42.193 PM	SaveCartRequest OnExecuting	1	trace	{"TerminalID":"HOUSTON-17"}		
> 10/24/2019, 8:17:46.136 PM	SaveCartRequest OnExecuted	1	trace	{"TerminalID":"HOUSTON-17"}		

## Build the deployable package

For detailed information about how to build deployable packages, see [Create deployable packages](#).

1. Copy the **Contoso.Diagnostic** and **Microsoft.ApplicationInsights** assemblies to the **\RetailSDK\References** folder.
2. Update the **BuildTools\Customization.settings** file, and add the following entries in the **<ItemGroup>** section.

```
<ISV_CommerceRuntime_CustomizableFile Include="$(SdkReferencesPath)\Contoso.Diagnostic.dll" />
<ISV_CommerceRuntime_CustomizableFile
  Include="$(SdkReferencesPath)\Microsoft.ApplicationInsights.dll" />;
```

3. Open an MSBuild **Command Prompt** window for Microsoft Visual Studio 2015, and run the **build** command in the root of your Retail SDK folder.
4. Enter the following command to generate the deployable package.

```
msbuild /t:rebuild
```

5. In the **RetailSDK\Packages\RetailDeployablePackage** folder, find the deployable package. Go to the **content.folder** folder, and make sure that your three files are in the package (**Packages\RetailDeployablePackage\content.folder\RetailServer\Code\bin\ext**).
6. Upload the deployable package to your Shared asset library in Microsoft Dynamics Lifecycle Services (LCS).
7. In LCS, open your environment's main page, and select **Environment Features > Retail and Commerce > Manage**.
8. Select **Apply Extension**, and select the extension from your library.
9. After the extension has been successfully deployed, open an instance of Modern POS (MPOS) or POS (CPOS) that has been activated against the Commerce Scale Unit.
10. Run the extension scenario that that uses custom Application Insights logging.
11. Refresh the query in Application Insights to verify that the traces from the extension are logged correctly.

## Log events to Application Insights in the POS extension projects

1. In the **RetailSDK\POS\Extensions** folder, create a new folder named **Libraries**.
2. Open a command prompt and navigate to the **Libraries** folder.
3. Install **npm**. The **npm** package can be downloaded and installed from [OpenJS](#).
4. Run this command to install the **npm** package for the JavaScript Application Insights package.

```
npm i --save @microsoft/applicationinsights-web
```

After the package is installed, the **POS/Extensions/Libraries** folder should contain the **node\_modules** folder. The **node\_modules** folder contains the Application Insights library files.

5. Check that the file **POS/Extensions/Libraries/node\_modules/@microsoft/applicationinsights-web/dist/applicationinsights-web.js** exists in the library.

The file name might change in future versions of the Application Insights library. If the path changes, update the library path in steps 8 and 10 to a path that points to the main Application Insights library.

6. Open **ModernPOS.sln** or **CloudPos.sln** from **RetailSDK\POS**.
7. Open the **tsconfig.json** file from the **POS.Extensions** project. Under the **exclude** section, add an entry to the **Libraries** folder.

```
"exclude": [  
  "Libraries"  
],
```

8. Open the **tsconfig.json** file from the **POS.Extensions** project. Under the **compilerOptions** section, add the following properties.

```
"baseUrl": "./",  
"paths": {  
  "applicationinsights-web": [ "Libraries/node_modules/@microsoft/applicationinsights-  
web/dist/applicationinsights-web" ]  
}
```

9. Edit the **Pos.Extensions.csproj** file in the **CopyPosExtensionsFiles** section. Add the following targets to copy the Application Insights library to the POS application, so that the targets can be consumed by the extension code.

```
<JavaScriptFileList Include="Libraries\\**\\*.js">  
  <InProject>false</InProject>  
  <Visible>false</Visible>  
</JavaScriptFileList>
```

10. Include the following node in the **manifest.json** file of the POS extension folder (package) that is consuming the Application Insights library.

```
{  
  "dependencies": [  
    {  
      "alias": "applicationinsights-web",  
      "format": "amd",  
      "modulePath": "../Libraries/node_modules/@microsoft/applicationinsights-  
web/dist/applicationinsights-web"  
    }  
  ]  
}
```

The Application Insights library is now ready to be consumed and used in POS.

## Consume the library and log events

1. Open the **ModernPOS.sln** or **CloudPos.sln** solution from **RetailSDK\POS**.
2. Create a new TypeScript file inside the POS extension folder (package) and name it **AppInsights.ts**.
3. Copy the following code to the file. The code is used by the extensions to track events using Application Insights. Use the instrumentation key created in Azure App Insights.

```
import { ApplicationInsights } from "applicationinsights-web";

/**
 * Example implementation of an Application Insights singleton that can be used to log events and
 * metrics on Application Insights.
 */
export class AppInsights {
    private static _instance: AppInsights = null;
    private _applicationInsights: ApplicationInsights = null;

    /**
     * Gets a global reference to an Application Insights reference that can be used by other
     * extension code.
     * @returns {ApplicationInsights} The ApplicationInsights instance that can be used to log
     * events.
     */
    public static get instance(): ApplicationInsights {
        if (AppInsights._instance === null) {
            AppInsights._instance = new AppInsights();
        }

        return AppInsights._instance._applicationInsights;
    }

    /**
     * Initializes a new instance of AppInsights.
     */
    constructor() {
        this._applicationInsights = new ApplicationInsights({
            config: {
                instrumentationKey: 'YOUR_INSTRUMENTATION_KEY_GOES_HERE'
                /* ...Other Configuration Options... */
            }
        });
        this._applicationInsights.loadAppInsights();
    }
}
```

4. In the extension code, log the events by calling the AppInsights class as shown in the following code example.

```
AppInsights.instance.trackEvent({
    name: "extensionTest",
    properties: {
        "property1": "value1",
        "property2": "value2",
    },
    measurements: {
        "measurement1": 1,
        "measurement2": 2,
    },
});
```

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create asynchronous Commerce (CRT) APIs in your business logic

2/18/2021 • 9 minutes to read • [Edit Online](#)

## NOTE

This topic applies to Microsoft Dynamics 365 Commerce version 10.0.10 and later.

This topic explains how to create new business logic (application programming interfaces, or APIs) for the Commerce Runtime (CRT) by using the new asynchronous framework. The Commerce API framework now supports an asynchronous programming model for extensions and out-of-box Commerce handlers.

Before this framework enhancement was added, requests could be run only synchronously. Long operations, like input/output (I/O) operations, database queries, or network requests, blocked the execution thread. Now that support for the asynchronous model has been added to the Commerce runtime (CRT), you can use asynchronous versions of these operations. Asynchronous requests unblock the execution thread.

The Commerce API framework now supports the **Task** and **Task<T>** classes that are supported by the **async** and **await** keywords for the extension CRT request handlers. You should use the asynchronous Commerce API framework for all new extension APIs and the out-of-box asynchronous Commerce API in extensions.

## Async classes/interface added in the Commerce API framework:

The following asynchronous classes and interfaces were added in the Commerce API framework.

CLASS/INTERFACE	DESCRIPTION
SingleAsyncRequestHandler	The base class for asynchronous handlers that support only one request.
IRequestHandlerAsync	The interface for the asynchronous request handler.
IRequestTriggerAsync	The interface for the request trigger.
CommerceControllerAsync	The base class for asynchronous extension Retail server controller class.
DatabaseContext	The base class for asynchronous database execution methods.

The following asynchronous methods were added in the Commerce API framework.

CLASS/INTERFACE	METHOD	DESCRIPTION
SingleAsyncRequestHandler	Task<TResponse> Process	The execute method that will be overridden by each derived class.
	Task<Response> Execute	The method that represents the entry point of the request handler.

CLASS/INTERFACE	METHOD	DESCRIPTION
IRequestHandlerAsync	Task<Response> Execute (Request request)	The interface for the asynchronous request handler.
IRequestTriggerAsync	Task OnExecuting(Request request)	The method that is invoked before the request has been processed by <b>IRequestHandler</b> .
IRequestTriggerAsync	Task OnExecuted(Request request, Response response)	The method that is invoked after the request has been processed by <b>IRequestHandler</b> .
DatabaseContext	async Task<Tuple<int, PagedResult>> ExecuteStoredProcedureAsync(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings)	The method that executes the stored procedure using the specified parameters.
DatabaseContext	async Task<ReadOnlyCollection> ExecuteNonPagedStoredProcedureAsync(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings) where T : CommerceEntity, new()	The method that executes the stored procedure using the specified parameters. The execution is non-paginated.
DatabaseContext	async Task<Tuple<PagedResult, ReadOnlyCollection>> ExecuteStoredProcedureAsync<T1, T2>(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new()	The method that executes the specified stored procedure.
DatabaseContext	async Task<Tuple<int, Tuple<PagedResult, ReadOnlyCollection>>> ExecuteStoredProcedureAsync<T1, T2>(string procedureName, ParameterSet parameters, ParameterSet outputParameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new()	The method that executes the specified stored procedure.
DatabaseContext	async Task<Tuple<PagedResult, ReadOnlyCollection, ReadOnlyCollection>> ExecuteStoredProcedureAsync<T1, T2, T3>(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new() where T3 : CommerceEntity, new()	The method that executes the specified stored procedure.

CLASS/INTERFACE	METHOD	DESCRIPTION
DatabaseContext	<pre> async Task&lt;Tuple&lt;int, Tuple&lt;PagedResult, ReadOnlyCollection, ReadOnlyCollection&gt;&gt;&gt; ExecuteStoredProcedureAsync&lt;T1, T2, T3&gt;(string procedureName, ParameterSet parameters, ParameterSet outputParameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new() where T3 : CommerceEntity, new() </pre>	The method that executes the specified stored procedure.
DatabaseContext	<pre> async Task&lt;Tuple&lt;PagedResult, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection&gt;&gt; ExecuteStoredProcedureAsync&lt;T1, T2, T3, T4&gt;(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new() where T3 : CommerceEntity, new() where T4 : CommerceEntity, new() </pre>	The method that executes the specified stored procedure.
DatabaseContext	<pre> async Task&lt;Tuple&lt;int, Tuple&lt;PagedResult, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection&gt;&gt;&gt; ExecuteStoredProcedureAsync&lt;T1, T2, T3, T4&gt;(string procedureName, ParameterSet parameters, ParameterSet outputParameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new() where T3 : CommerceEntity, new() where T4 : CommerceEntity, new() </pre>	The method that executes the specified stored procedure.

CLASS/INTERFACE	METHOD	DESCRIPTION
DatabaseContext	async Task<Tuple<PagedResult, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection, Tuple<ReadOnlyCollection>>>> ExecuteStoredProcedureAsync<T1, T2, T3, T4, T5, T6, T7, T8>(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new() where T3 : CommerceEntity, new() where T4 : CommerceEntity, new() where T5 : CommerceEntity, new() where T6 : CommerceEntity, new() where T7 : CommerceEntity, new() where T8 : CommerceEntity, new()	The method that executes the specified stored procedure.
DatabaseContext	Task<Tuple<int, Tuple<PagedResult, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection, ReadOnlyCollection, Tuple<ReadOnlyCollection>>>>> ExecuteStoredProcedureAsync<T1, T2, T3, T4, T5, T6, T7, T8>(string procedureName, ParameterSet parameters, ParameterSet outputParameters, QueryResultSettings resultSettings) where T1 : CommerceEntity, new() where T2 : CommerceEntity, new() where T3 : CommerceEntity, new() where T4 : CommerceEntity, new() where T5 : CommerceEntity, new() where T6 : CommerceEntity, new() where T7 : CommerceEntity, new() where T8 : CommerceEntity, new()	The method that executes the specified stored procedure.
DatabaseContext	Task<ReadOnlyCollection> ExecuteStoredProcedureScalarCollectionAsync(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings)	Executes a query that returns a collection of single results.
DatabaseContext	Task ExecuteStoredProcedureNonQueryAsync(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings)	The method that executes the specified stored procedure with the specified parameters.



CLASS/INTERFACE	METHOD	DESCRIPTION
DatabaseContext	Task ExecuteStoredProcedureScalarAsync(string procedureName, ParameterSet parameters, QueryResultSettings resultSettings)	The method that executes the stored procedure using the specified parameters and returns the return value.
DatabaseContext	Task OnExecuting(Request request)	Executes the stored procedure using the specified parameters and returns the return value.
DatabaseContext	Task ExecuteStoredProcedureScalarAsync(string procedureName, ParameterSet parameters, ParameterSet outputParameters, QueryResultSettings resultSettings)	The method that executes the stored procedure using the specified parameters and returns the return value.
DatabaseContext	Task<PagedResult> ReadEntityAsync(IDatabaseQuery query) where T : CommerceEntity, new()	The method that reads an entity from the database.
DatabaseContext	Task ExecuteNonQueryAsync(IDatabaseQuery query)	The method that executes a query that has no output.
DatabaseContext	Task ExecuteScalarAsync(IDatabaseQuery query)	The method that executes a query that has no output.
DatabaseContext	Task<ReadOnlyCollection> ExecuteScalarCollectionAsync(IDatabaseQuery query)	The method that executes a query that returns a collection of single results.

Asynchronous execution is supported for these scenarios:

- New Commerce APIs
  - Asynchronous Commerce Runtime API
  - Asynchronous Retail server controller
- Overrides of the Commerce API handler that is running
- Pre-triggers and post-triggers

## Create a new asynchronous Commerce Runtime API

To create a new asynchronous Commerce API, you must create three classes:

- Create the request class by implementing the **Request** class.
- Create the asynchronous response class by implementing the **Response** class.
- Create the asynchronous handler class.

To create the classes, follow these steps.

1. Create the request class.

```

namespace Contoso
{
    namespace Commerce.Runtime.AsyncRequestSample.Messages
    {
        using System.Runtime.Serialization;
        using System;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;

        /// <summary>
        /// Sample request for AsyncRequestSampleRequest
        /// </summary>
        [DataContract]
        public sealed class AsyncRequestSampleRequest : Request
        {
            /// <summary>
            /// Initializes a new instance of the <see cref="AsyncRequestSampleRequest"/> class.
            /// </summary>
            public AsyncRequestSampleRequest()
            {
            }

        }
    }
}

```

## 2. Create the response class.

```

namespace Contoso
{
    namespace Commerce.Runtime.AsyncSample.Messages
    {
        using System.Runtime.Serialization;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;

        /// <summary>
        /// Defines a simple response class .
        /// </summary>
        [DataContract]
        public sealed class AsyncSampleResponse : Response
        {
            /// <summary>
            /// Initializes a new instance of the <see cref="AsyncSampleResponse"/> class.
            /// </summary>
            /// <param name="CustomerRec">Customer.</param>
            public AsyncSampleResponse(Customer customer)
            {
                this.CustomerRec = customer;
            }

            /// <summary>
            /// Gets the customer related to the request.
            /// </summary>
            [DataMember]
            public Customer CustomerRec { get; private set; }

        }
    }
}

```

## 3. Create the asynchronous handler.

```

using Contoso.Commerce.Runtime.AsyncSample.Messages;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.Dynamics.Commerce.Runtime;
using Microsoft.Dynamics.Commerce.Runtime.DataModel;
using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;

namespace Commerce.Runtime.AsyncSample.Messages
{
    public class AsyncSampleRequestHandler : SingleAsyncRequestHandler<AsyncSampleRequest,
    AsyncSampleResponse>
    {
        protected override async Task<AsyncSampleResponse> Process(AsyncSampleRequest request)
        {
            var customers = await AsyncSampleMethodGetCustomer(request.RequestContext);

            return new AsyncSampleResponse(customers);
        }

        private async Task<Customer> AsyncSampleMethodGetCustomer(RequestContext context)
        {
            //do custom logic here. Execute any custom logic or request asynchronously.
            var getCustomerRequest = new GetCustomersServiceRequest(QueryResultSettings.SingleRecord,
            "2001", SearchLocation.Local);
            var getCustomerResponse = await context.ExecuteAsync<GetCustomersServiceResponse>
            (getCustomerRequest);
            return getCustomerResponse.Customers.SingleOrDefault();
        }
    }
}

```

## Create a new asynchronous Retail server controller

To create an asynchronous Retail server controller extension, extend the controller class from the **CommerceControllerAsync** class, as shown in the following example.

```

namespace Microsoft.Dynamics.Retail.RetailServerLibrary.ODataControllers
{
    using System.Runtime.InteropServices;
    using System.Threading.Tasks;
    using System.Web.Http.Controllers;
    using System.Web.OData;
    using Commerce.Runtime;
    using Commerce.Runtime.Client;
    using Commerce.Runtime.DataModel;

    /// <summary>
    /// The catalogs controller.
    /// </summary>
    [ComVisible(false)]
    public class CustomController : CommerceControllerAsync<MyEntity, string>
    {

        /// <summary>
        /// Gets the controller name.
        /// </summary>
        public override string ControllerName
        {
            get { return "MyEntity"; }
        }

        /// <summary>
        /// Gets the <c>MyEntity</c> entity by key.
        /// </summary>
        /// <param name="key">
        /// The key.
        /// </param>
        /// <returns>
        /// The <see cref="ScanResult"/>.
        /// </returns>
        [CommerceAuthorization(CommerceRoles.Employee)]
        protected override Task<MyEntity> GetEntityByKey(string key)
        {
            return this.GetMyEntityAsync(key);
        }

        /// <summary>
        /// Retrieves the entity information based on <see cref="key"/> input.
        /// </summary>
        /// <param name="key">key input.</param>
        /// <returns>Instance of <see cref="MyEntity"/> that contains entities found by input text.
    </returns>
        public async Task<MyEntity> GetMyEntityAsync(string key)
        {
            ThrowIf.Null(key, nameof(key));

            var request = new MyCustomRequest(key);
            var response = await this.runtime.ExecuteAsync<MyCustomResponse>(request).ConfigureAwait(false);
            return response.Result;
        }
    }
}

```

## Override an out-of-box request handler asynchronously

To override the supported out-of-box request handler, follow this pattern.

For example, to override the out-of-box `GetScanResultRequestHandler` handler, override the handler, and return the response by using `Task` and `await`.

```

namespace Contoso
{
    namespace Commerce.Runtime.Workflow
    {
        using System.Linq;
        using System.Threading.Tasks;
        using Microsoft.Dynamics.Commerce.Runtime;
        using Microsoft.Dynamics.Commerce.Runtime.DataModel;
        using Microsoft.Dynamics.Commerce.Runtime.Messages;
        using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;

        /// <summary>
        /// Workflow that retrieves information based on scan input.
        /// </summary>
        public class GetScanResultRequestHandler : SingleAsyncRequestHandler<GetScanResultRequest,
GetScanResultResponse>
        {
            /// <summary>
            /// Executes the workflow to get scan result.
            /// </summary>
            /// <param name="request">Instance of <see cref="GetScanResultRequest"/>.</param>
            /// <returns>Instance of <see cref="GetScanResultResponse"/>.</returns>
            protected override async Task<GetScanResultResponse> Process(GetScanResultRequest request)
            {
                ThrowIf.Null(request, nameof(request));

                var getBarcodeRequest = new GetBarcodeRequest(request.ScanInfo);
                var getBarcodeResponse = request.RequestContext.Execute<GetBarcodeResponse>
(getBarcodeRequest);
                Barcode barcode = getBarcodeResponse.Barcode;
                BarcodeMaskType maskType = barcode == null ? BarcodeMaskType.None : barcode.Mask.MaskType;
                ScanResult result = new ScanResult(request.ScanInfo.ScannedText) { Barcode = barcode,
MaskType = maskType };

                //For this sample, we skipped lot of logic inside this method. This should be used only for
reference on how to override OOB request. The logic written inside this sample is different from the actual
implementation of this handler.

                result.Product = await this.GetSingleProductByItemId(request.RequestContext,
barcode.ItemBarcode.ItemId, barcode.ItemBarcode.InventoryDimensionId).ConfigureAwait(false);

                return new GetScanResultResponse(result);
            }

            private async Task<SimpleProduct> GetSingleProductByItemId(RequestContext context, string
itemId, string inventDimId)
            {
                //do custom logic here. Execute any custom logic or request asynchronously.

                if (string.IsNullOrEmpty(itemId))
                {
                    return null;
                }

                var lookupClause = new ProductLookupClause { ItemId = itemId, InventDimensionId =
inventDimId };

                var getProductRequest = new GetProductsServiceRequest(
                    context.GetPrincipal().ChannelId,
                    new[] { lookupClause },
                    QueryResultSettings.AllRecords);
                getProductRequest.SearchLocation = SearchLocation.Local; // Scanned products must be found
locally.

                var products = (await context.ExecuteAsync<GetProductsServiceResponse>
(getProductRequest).ConfigureAwait(false)).Products;

                if (products.Results.IsNullOrEmpty() || products.Results.HasMultiple())
                    ,

```

```

        {
            // if product is not found or multiple products founds (not exact match) return 'null'
            return null;
        }

        return products.Results.Single();
    }
}
}
}

```

## Create a trigger asynchronously

To run the logic in the trigger asynchronously, extend the **IRequestTriggerAsync** interface, and add the logic.

```

namespace Contoso
{
    using System;
    using System.Collections.Generic;
    using System.Composition;
    using System.Threading.Tasks;
    using Microsoft.Dynamics.Commerce.Runtime.Messages;
    using Microsoft.Dynamics.Commerce.Runtime.DataServices.Messages;

    /// <summary>
    /// Test implementation of request trigger for <see cref="TestRequest"/>.
    /// </summary>
    public class SampleRequestTriggerAsync : IRequestTriggerAsync
    {
        private readonly Func<Task> onExecutingCallback;
        private readonly Func<Task> onExecutedCallback;

        public TestRequestTriggerAsync()
        {
        }

        [ImportingConstructor]
        public TestRequestTriggerAsync([Import("OnExecutingCallback")]Func<Task> onExecutingCallback,
[Import("OnExecutedCallback")]Func<Task> onExecutedCallback)
        {
            this.onExecutingCallback = onExecutingCallback;
            this.onExecutedCallback = onExecutedCallback;
        }

        /// <summary>
        /// Gets the collection of request types supported by this trigger.
        /// </summary>
        public IEnumerable<Type> SupportedRequestTypes
        {
            get
            {
                return new[]
                {
                    // Add the request for pre or post trigger execution.
                    typeof(GetCustomerDataRequest)
                };
            }
        }

        /// <summary>
        /// Invoked before request has been processed by <see cref="IRequestHandler"/>.
        /// </summary>
        /// <param name="request">The incoming request message.</param>
        /// <returns>The empty Task.</returns>
        public Task OnExecuting(Request request)
    }
}

```

```
public Task OnExecuting(Request request)
{
    if (this.onExecutingCallback != null)
    {
        return this.onExecutingCallback();
    }

    return Task.CompletedTask;
}

/// <summary>
/// Invoked after request has been processed by <see cref="IRequestHandler"/>.
/// </summary>
/// <param name="request">The request message processed by handler.</param>
/// <param name="response">The response message generated by handler.</param>
/// <returns>The empty Task.</returns>
public Task OnExecuted(Request request, Response response)
{
    if (this.onExecutedCallback != null)
    {
        return this.onExecutedCallback();
    }

    return Task.CompletedTask;
}
}
```

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Pre-extended columns in the channel database

2/18/2021 • 4 minutes to read • [Edit Online](#)

Some columns in the channel database are *pre-extended*. In other words, the column length in the channel database exceeds the column length in Microsoft Dynamics 365 Commerce Headquarters. For example, the length of the **INVENTSERIALID** field is 20 characters in the Commerce Headquarters database but 50 characters in the channel database.

Although fields in the channel database are often extended, column lengths for those fields aren't extensible. Therefore, out-of-box column lengths have been increased to support extension scenarios.

## NOTE

If you must extend a field that isn't already pre-extended, you must file an extension request in Lifecycle Services (LCS). For more information, see [Extensibility requests](#).

Although the fields are extended in the channel database, your extension must also extend the field in the Commerce Headquarters database by using the extended data type (EDT) extension model. Additionally, you must extend the corresponding point of sale (POS) or Commerce runtime (CRT) user interface (UI).

If the Commerce Headquarters database isn't extended, either synchronization between the channel database and the Commerce Headquarters database will fail during the P-job or the extra characters will be truncated. Likewise, if the corresponding point of sale (POS) user interface (UI) or the Commerce runtime (CRT) isn't extended, validation will prevent more than the default character length from being entered. The default length is determined by the base column length in the Commerce Headquarters database.

Here are some examples:

- The length of the **INVENTSERIALID** field in the channel database is 50 characters, but the POS UI accepts serial numbers that are a maximum of only 20 characters long. In this case, you must extend the field in both the POS UI and the Commerce Headquarters database.
- The length of the **STREET** field in the channel database is extended to 400 characters, but validation in CRT prevents more than the default length in the Commerce Headquarters database from being accepted. In this case, you must extend both the CRT request handler (**ValidateAddressLengthServiceRequest**) and the Commerce Headquarters database, so that 400 characters are accepted for the **STREET** field.

Extension of the POS UI or CRT handlers isn't required for some fields, because they might be read-only fields in POS. For example, **ECORES** product-related fields are read-only. Because product creation isn't supported in POS, there is no write scenario for products in POS.

## Sample code to override the **ValidateAddressLengthServiceRequest** handler

The following example shows how to override the **ValidateAddressLengthServiceRequest** handler.

```
namespace Contoso
{
    namespace Commerce.Runtime.ReceiptsSample
    {
        using System;
        using System.Collections.Generic;
        using Microsoft.Dynamics.Commerce.Runtime;
```



```

using Microsoft.Dynamics.Commerce.Runtime.Messages;
using Microsoft.Dynamics.Commerce.Runtime.Services.Messages;

public class ValidateAddressLengthServiceRequestExt : IRequestHandler
{
    private static int maxDefaultFullAddressColumnLength = 250;
    private static int maxDefaultStreetColumnLength = 250;
    private static int maxDefaultCountyColumnLength = 10;

    /// <summary>
    /// Gets the collection of supported request types by this handler.
    /// </summary>
    public IEnumerable<Type> SupportedRequestTypes
    {
        get
        {
            return new[]
            {
                typeof(ValidateAddressLengthServiceRequest),
            };
        }
    }

    public Response Execute(Request request)
    {
        if (request == null)
        {
            return null;
        }

        ValidateAddressLengthServiceRequest validateAddressLengthServiceRequest =
(ValidateAddressLengthServiceRequest)request;

        var validationFailures = new List<DataValidationFailure>();

        // Add custom logic to check your desired length.

        if (!string.IsNullOrEmpty(validateAddressLengthServiceRequest?.Address?.FullAddress)
            && validateAddressLengthServiceRequest.Address.FullAddress.Length >
maxDefaultFullAddressColumnLength)
        {
            validationFailures.Add(new DataValidationFailure(
                DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_AddressLengthExceeded,
                string.Format("The full address exceeds the maximum number of {0} characters
allowed.",
                    maxDefaultFullAddressColumnLength))
            {
                LocalizedMessageParameters = new object[] { maxDefaultFullAddressColumnLength }
            });
        }

        // Add custom logic to check your desired length.

        if (!string.IsNullOrEmpty(validateAddressLengthServiceRequest?.Address?.Street)
            && validateAddressLengthServiceRequest.Address.Street.Length > maxDefaultStreetColumnLength)
        {
            validationFailures.Add(new DataValidationFailure(
                DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_StreetLengthExceeded,
                string.Format("The street exceeds the maximum number of {0} characters allowed.",
maxDefaultStreetColumnLength))
            {
                LocalizedMessageParameters = new object[] { maxDefaultStreetColumnLength }
            });
        }

        // Add custom logic to check your desired length.

        if (!string.IsNullOrEmpty(validateAddressLengthServiceRequest?.Address?.County)
            && validateAddressLengthServiceRequest.Address.County.Length > maxDefaultCountyColumnLength)

```

```

        {
            validationFailures.Add(new DataValidationFailure(
                DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_CountyLengthExceeded,
                string.Format("The county exceeds the maximum number of {0} characters allowed.",
                    maxDefaultCountyColumnLength))
                {
                    LocalizedMessageParameters = new object[] { maxDefaultCountyColumnLength }
                });
        }

        if (validationFailures.Count > 0)
        {
            throw new
            DataValidationException(DataValidationErrors.Microsoft_Dynamics_Commerce_Runtime_AggregateValidationError,
                validationFailures, "An error occurred when validating the address.");
        }

        return new NullResponse();
    }
}
}
}
}

```

## Pre-extended columns in the channel database

The following table lists the columns that are pre-extended.

TABLE	COLUMN	LENGTH	EXTENSION IN CRT	EXTENSION IN POS
INVENTSERIAL	INVENTSERIALID	nvarchar(50)		GetSerialNumberClientRequestHandler
LOGISTICSPOSTALADDRESS	ADDRESS	nvarchar(500)	ValidateAddressLength	
LOGISTICSPOSTALADDRESS	STREET	nvarchar(400)	ValidateAddressLength	
LOGISTICSPOSTALADDRESS	COUNTY	nvarchar(60)	ValidateAddressLength	
LOGISTICSADDRESSCITY	COUNTYID	nvarchar(60)		
LOGISTICSADDRESSCOUNTRY	COUNTYID	nvarchar(60)		
LOGISTICSADDRESSDISTRICT	COUNTYID_RU	nvarchar(60)		
LOGISTICSADDRESSZIPCODE	STREETNAME	nvarchar(400)		
LOGISTICSADDRESSZIPCODE	COUNTY	nvarchar(60)		
RETAILASYNCADDRESS	STREET	nvarchar(400)		

TABLE	COLUMN	LENGTH	EXTENSION IN CRT	EXTENSION IN POS
RETAILASYNCADDRESS	COUNTY	nvarchar(60)		
RETAILASYNCCUSTOMER	STREET	nvarchar(400)		
RETAILASYNCCUSTOMER	COUNTY	nvarchar(60)		
RETAILFISCALDOCUMENT_BR	FEADDRESSSTREET	nvarchar(400)		
RETAILFISCALDOCUMENT_BR	THIRDPARTYADDRESSSTREET	nvarchar(400)		
RETAILTAXFILTERS	COUNTYID	nvarchar(60)		
RETAILTRANSACTIONADDRESSTRANS	STREET	nvarchar(400)		
RETAILTRANSACTIONADDRESSTRANS	COUNTY	nvarchar(60)		
RETAILTRANSACTIONSALESTRANS	INVENTBATCHID	nvarchar(50)		
RETAILTRANSACTIONSALESTRANS	INVENTSERIALID	nvarchar(50)		
RETAILTRANSACTIONSALESTRANS	WAREHOUSELOCATION	nvarchar(60)		
INVENTDIM	INVENTBATCHID	nvarchar(50)		
INVENTDIM	INVENTSERIALID	nvarchar(50)		
INVENTDIM	CONFIGID	nvarchar(60)		
INVENTDIM	INVENTCOLORID	nvarchar(60)		
INVENTDIM	INVENTSIZEID	nvarchar(60)		
INVENTDIM	INVENTSTYLEID	nvarchar(60)		
INVENTDIM	WMSLOCATIONID	nvarchar(60)		
ECORESOLOR	NAME	nvarchar(60)		
ECORESSTYLE	NAME	nvarchar(60)		
ECORESCONFIGURATION	NAME	nvarchar(60)		

TABLE	COLUMN	LENGTH	EXTENSION IN CRT	EXTENSION IN POS
ECORESSIZE	NAME	nvarchar(60)		
RETAILTABLEFIELDID	TABLENAME	nvarchar(81)		
	FIELDNAME	nvarchar(81)		
RETAILTRANSACTION TAXTRANSGTE	TAXCOMPONENT	nvarchar(60)		
TAXCOMPONENTTAB LE_IN	COMPONENT	nvarchar(60)		
WMSLOCATION	INPUTLOCATION	nvarchar(60)		
	WMSLOCATIONID	nvarchar(60)		
INVENTLOCATION	RBODEFAULTWMSLO CATIONID	nvarchar(60)		
	WMSLOCATIONIDDE FAULTISSUE	nvarchar(60)		
	WMSLOCATIONIDDE FAULTRECEIPT	nvarchar(60)		
PRICEDISCTABLE	CONSTRAINT I_137462222_19048 21809: { "action": "replace-line", "value": "CONSTRAINT [I_PRICEDISCTABLE_R ECID] PRIMARY KEY CLUSTERED ( [RECID] ) "CONSTRAINT I_PRICEDISCTABLE_R ECID": { "action": "remove-line" }			
OMOPERATINGUNIT	OMOPERATINGUNIT NUMBER	nvarchar(30)		
RETAILPARAMETERS	USEADVANCEDAUTO CHARGE INT NULL DEFAULT(0)			
	GIFTCARDINQUIRYP RINTHISTORY INT NULL DEFAULT(0)			

TABLE	COLUMN	LENGTH	EXTENSION IN CRT	EXTENSION IN POS
RETAILINFOCODETABLE	PRINTINPUTONFISCALRECEIPT [int] NULL DEFAULT (0) PRINTTEXTONFISCALRECEIPT [nvarchar] (50) NULL DEFAULT("")			
RETAILINFORMATI NSUBCODETABLE	PRINTTEXTONFISCALRECEIPT [nvarchar] (50) NULL DEFAULT("")			

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Extend Commerce Data Exchange - Real-time Service

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how you can extend Commerce Data Exchange (CDX) - Real-time service by adding extension methods to the `RetailTransactionServiceEx` class. Real-time Service enables clients to interact with Commerce functionality in real time. Finance and Operation databases and classes can't be accessed directly from Retail server. You should access them through the CDX class extension using the Finance and Operations and Commerce Runtime extension.

To extend Commerce Data Exchange - Real-time Service, you create a new method in the `RetailTransactionServiceEx` class. This method must meet the following criteria:

- The method must be a public static method.
- The return value must be a container that has a length of 2 or more. The first element must be a Boolean value that indicates whether the method call was successful, and a string value that you can use for a comment or error message. The other items in the container can be of any type, and they can even be nested containers.
- The method parameters must be one of the following primitive types:
  - Boolean
  - date
  - int
  - int64
  - str
  - guid
  - Real

## Create and call a new extension method

1. Start Microsoft Visual Studio.
2. On the **Dynamics 365** menu, click **Model management > Create model**.
3. In the **Create model** dialog box, enter the following details.
  - **Model name** - Contoso
  - **Model publisher** - Contoso
  - **Layer** - USR (Select the relevant layer)
  - **Version** - 1.0.0.0
  - **Model display name** - Contoso
4. Click **Next**.
5. In the dialog box, select **Select existing package**, and then select **Application Suite** in the list.
6. Click **Next**.
7. Click **Finish**.
8. In the **New project** dialog box, enter **ContosoRetailTransactionServiceEx** as the project name.

- Click **OK**.
- Right-click the project and select **Add > New item**. In the **Add New Item** window, select **Class** and enter the name of the class as **ContosoRetailTransactionServiceSample**.

To consume the CDX method in Commerce runtime (CRT) you must add the `ExtensionOf` attribute to your class, such as `ExtensionOf(classStr(RetailTransactionServiceEx))`. This means that the class is extending from the `RetailTransactionServiceEx`.

- In the code editor, add the following code.

```
[ExtensionOf(classStr(RetailTransactionServiceEx))]  
final class ContosoRetailTransactionServiceSample  
{  
}
```

- Inside the class, add a new method to do your custom logic. This is the method that you will call from CRT to do the custom logic.

```
[ExtensionOf(classStr(RetailTransactionServiceEx))]  
final class ContosoRetailTransactionServiceSample  
{  
    public static container SerialCheck(str _serialNum)  
    {  
        boolean success = false;  
        str errorMessage;  
        int fromLine;  
  
        ttsbegin;  
  
        try  
        {  
            if (_serialNum)  
            {  
                // check whether the serial number exists  
  
                // Add your custom logic  
  
                errorMessage = "Serial number found";  
            }  
            else  
            {  
                // Add your custom logic  
                success = false;  
                errorMessage = "Serial number not found";  
            }  
        }  
        catch (Exception::Error)  
        {  
            error = RetailTransactionServiceUtilities::getInfologMessages(fromLine);  
        }  
  
        ttscommit;  
  
        // Return sanitized error code.  
        errorMessage = RetailTransactionServiceUtilities::getErrorCode(errorMessage);  
  
        return [success, errorMessage, "Custom values"];  
    }  
}
```

- In Solution Explorer, right-click the project, and then click **Build**.

After you've finished building your new extension methods, the project will be deployed.

## Call the new method from the CRT

1. In your commerce runtime (CRT) extension, include the `Microsoft.Dynamics.Commerce.Runtime.RealtimeServices.Messages` nuget package, if it hasn't already been added.
2. Use the following sample code to call the new method.

```
InvokeExtensionMethodRealtimeRequest extensionRequest = new
InvokeExtensionMethodRealtimeRequest("SerialCheck", "123");
InvokeExtensionMethodRealtimeResponse response = await
request.RequestContext.ExecuteAsync<InvokeExtensionMethodRealtimeResponse>
(extensionRequest).ConfigureAwait(false);
ReadOnlyCollection<object> results = response.Result;

string resValue = (string)results[0];
```

3. From the results object, you can read the response values from Real-time Service.
4. The CRT framework code will check the success/failure state and provide an error message based on the values returned from the CDX methods. If required, the extension code can catch this and provide additional logic.

### NOTE

The `InvokeExtensionMethodRealtimeRequest` method takes two parameters. One parameter is the Real-time Service method name, and the other is the list of parameters that should be used. The method name that is passed should be the same as the method name that you created in the `ContosoRetailTransactionServiceSample` class.

```
public InvokeExtensionMethodRealtimeRequest(string methodName, params object[] parameters)
    : base(methodName, parameters)
{
}
```

## CDX offline

When there is no connectivity to the HQ, client/Retail Server will not be able to call the CDX method. In this case, the extension code should follow the best practice mentioned below:

- Check before calling the CDX method to determine if CRT is connected to the online (Retail server) or the offline (local) database. This can be done both in POS and CRT.

### How to check the connection status

#### POS

Use the `GetConnectionStatusClientRequest` POS API.

#### CRT

```
if(request.RequestContext.Runtime.Configuration.IsMasterDatabaseConnectionString)
{ }
```



- If the connection to the CDX method failed, an error message might display saying that the operation cannot be performed if there is no connectivity to HQ or that you need to have mitigation logic if this operation needs to work if there is no connectivity to the CDX method.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create an end-to-end payment integration for a payment terminal

2/18/2021 • 22 minutes to read • [Edit Online](#)

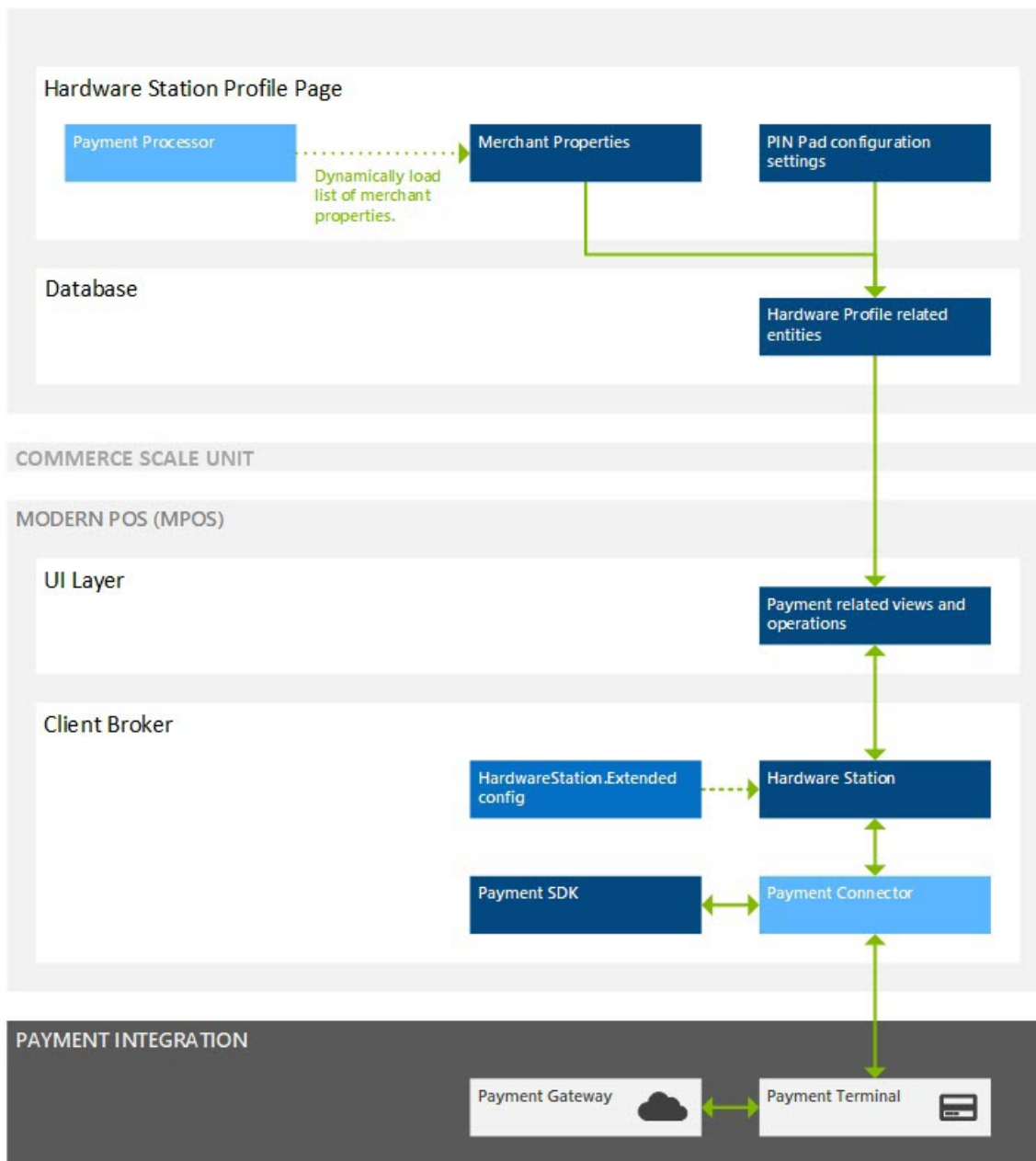
This topic describes how to write a payment integration for Microsoft Dynamics 365 Retail Modern POS and Cloud POS (the POS) for a payment terminal that can directly communicate with the payment gateway.

## Key terms

TERM	DESCRIPTION
Payment connector	An extension library that is written to integrate the POS with a payment terminal.
Payment processor	An extension library that is written to retrieve merchant properties that the payment connector uses.

## Overview

The following illustration shows a high-level overview of the payment terminal integration through the POS. Although this illustration assumes that a local Hardware Station is used to communicate with the payment terminal, the same patterns apply to a shared Hardware Station.



This topic describes the following steps that are required to create an end-to-end payment integration for a payment terminal:

- **Write a payment connector:** The payment connector is the main integration point between the POS and the payment terminal. The section for this step describes how to implement and configure a new payment connector that can relay payment requests (for example, authorize, refund, and void requests) to the payment terminal.
- **Write a payment processor:** The payment processor is used to define the merchant properties that are used as part of the payment integration. The section for this step describes how to implement a new payment processor. It includes information about the interfaces that you should implement and patterns that you should follow.

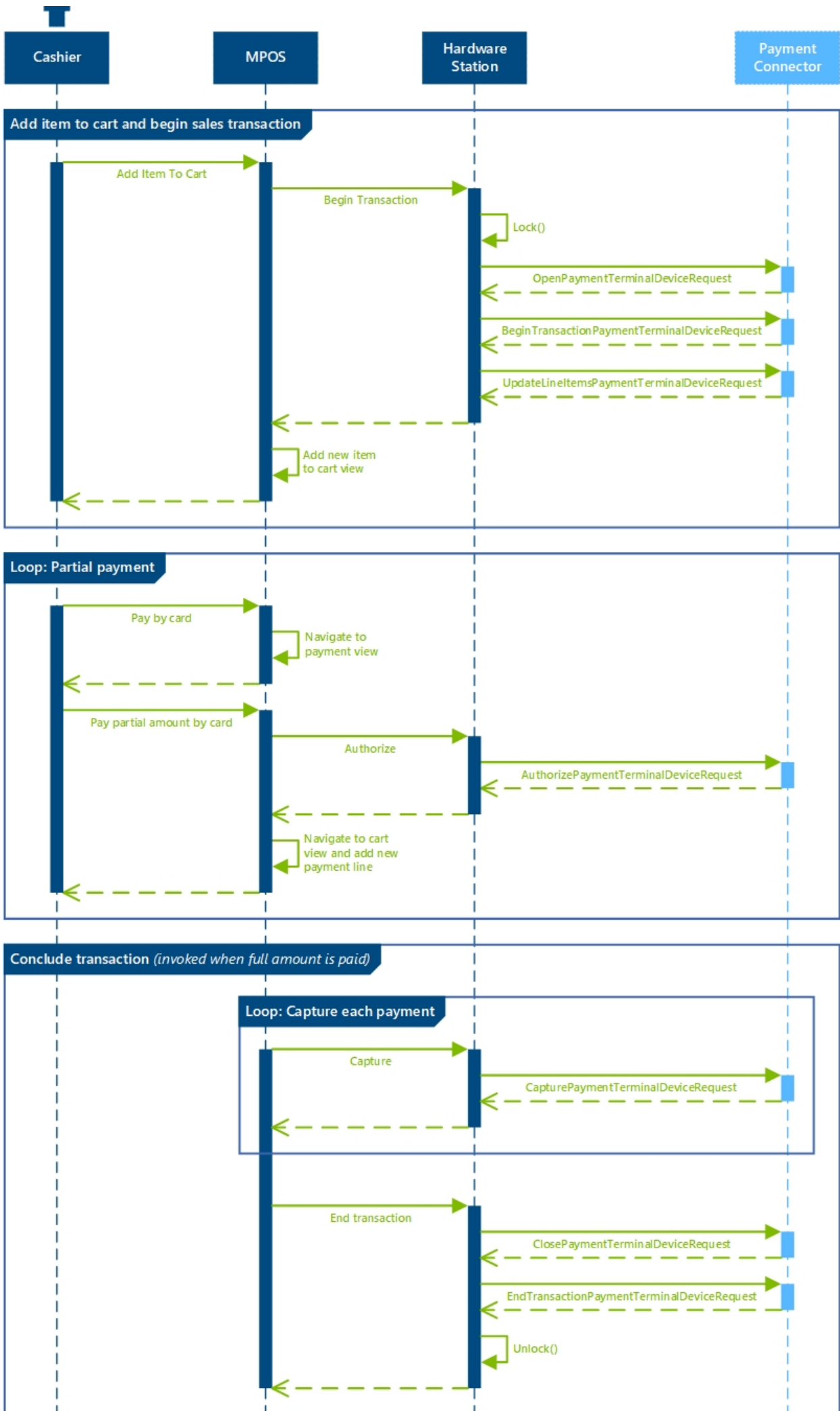
## Write a payment connector

This section describes how to write a new payment connector.

### Understanding the payment flows

The following illustration shows a high-level overview of several payment flows (Begin Transaction, Update Cart Lines, Authorize, Capture, and End Transaction) across the POS, Hardware Station, and payment connector.





## Implement a payment connector

This section below describes how to implement a new payment connector. The examples that are shown here can be found in the **PaymentDeviceSample** class that is located under the **SampleExtensions\HardwareStation\Extension.PaymentSample** folder in the Retail software development kit (SDK).

### Implement the **INamedRequestHandler** interface

All POS payment-related flows are handled through request/response patterns in the Hardware Station. The first step in the process of writing a new payment connector is to create a class that implements the **INamedRequestHandler** interface that is defined in the **Microsoft.Dynamics.Commerce.Runtime.Framework** library.

```
namespace Contoso.Commerce.HardwareStation.PaymentSample
{
    public class PaymentDeviceSample : INamedRequestHandler
    {
        private const string PaymentTerminalDevice = "MOCKPAYMENTTERMINAL";

        /// <summary>
        /// Gets the specify the name of the request handler.
        /// </summary>
        public string HandlerName
        {
            get
            {
                return PaymentDeviceSample.PaymentTerminalDevice;
            }
        }
    }
}
```

The **HandlerName** string is used to configure the payment connector that is used on a given POS register through the client (see the information later in this topic).

### Implement supported payment requests

To process payment-related flows, the payment connector must define the supported request types that it can handle. Additionally, the **Execute** method must be implemented to route each request that the connector supports to a given method. The following example shows the complete list of supported request types and an example of a specific request (that is, an authorize request).

```
namespace Contoso.Commerce.HardwareStation.PaymentSample
{
    /// <summary>
    /// <c>Simulator</c> manager payment device class.
    /// </summary>
    public class PaymentDeviceSample : INamedRequestHandler
    {
        /// <summary>
        /// Gets the collection of supported request types by this handler.
        /// </summary>
        public IEnumerable<Type> SupportedRequestTypes
        {
            get
            {
                return new[]
                {
                    typeof(LockPaymentTerminalDeviceRequest),
                    typeof(OpenPaymentTerminalDeviceRequest),
                    typeof(ClosePaymentTerminalDeviceRequest),
                    typeof(BeginTransactionPaymentTerminalDeviceRequest),
                    typeof(EndTransactionPaymentTerminalDeviceRequest),
                }
            }
        }
    }
}
```

```

        typeof(UpdateLineItemsPaymentTerminalDeviceRequest),
        typeof(AuthorizePaymentTerminalDeviceRequest),
        typeof(CapturePaymentTerminalDeviceRequest),
        typeof(VoidPaymentTerminalDeviceRequest),
        typeof(RefundPaymentTerminalDeviceRequest),
        typeof(FetchTokenPaymentTerminalDeviceRequest),
        typeof(ExecuteTaskPaymentTerminalDeviceRequest),
        typeof(ActivateGiftCardPaymentTerminalRequest),
        typeof(AddBalanceToGiftCardPaymentTerminalRequest),
        typeof(GetGiftCardBalancePaymentTerminalRequest),
        typeof(GetPrivateTenderPaymentTerminalDeviceRequest),
        typeof(CancelOperationPaymentTerminalDeviceRequest),
        typeof(GetTransactionReferencePaymentTerminalDeviceRequest),
        typeof(GetTransactionByTransactionReferencePaymentTerminalDeviceRequest),
        typeof(CashoutGiftCardPaymentTerminalRequest)
    };
}
}

/// <summary>
/// Executes the payment device simulator operation based on the incoming request type.
/// </summary>
/// <param name="request">The payment terminal device simulator request message.</param>
/// <returns>Returns the payment terminal device simulator response.</returns>
public Response Execute(Microsoft.Dynamics.Commerce.Runtime.Messages.Request request)
{
    ThrowIf.Null(request, "request");

    Type requestType = request.GetType();

    if (requestType == typeof(AuthorizePaymentTerminalDeviceRequest))
    {
        return this.AuthorizePayment((AuthorizePaymentTerminalDeviceRequest)request);
    }
    else if (...)
    {
        ...
    }

    return new NullResponse();
}

/// <summary>
/// Authorize payment.
/// </summary>
/// <param name="request">The authorize payment request.</param>
/// <returns>The authorize payment response.</returns>
public AuthorizePaymentTerminalDeviceResponse AuthorizePayment(AuthorizePaymentTerminalDeviceRequest
request)
{
    ThrowIf.Null(request, "request");

    PaymentInfo paymentInfo = Utilities.WaitAsyncTask(() =>
this.AuthorizePaymentAsync(request.Amount, request.Currency, request.VoiceAuthorization,
request.IsManualEntry, request.ExtensionTransactionProperties));

    return new AuthorizePaymentTerminalDeviceResponse(paymentInfo);
}
}
}

```

#### Full list of supported request types

The following table describes all supported requests types that a payment connector can implement.

REQUEST CLASS	PAYMENT FLOW DESCRIPTION
OpenPaymentTerminalDeviceRequest	This request is called before a sales transaction is initiated. It is used to establish a connection to the payment terminal.
BeginTransactionPaymentTerminalDeviceRequest	This request is called when a new sales transaction is initiated. It is used to handle any initialization on the payment terminal (for example, by initializing the transaction screen).
LockPaymentTerminalDeviceRequest	This request is called when a payment terminal is locked for a transaction.
UpdateLineItemsPaymentTerminalDeviceRequest	This request is called when line items in the cart are updated.
AuthorizePaymentTerminalDeviceRequest	This request is called when a payment is initiated in the POS payment view.
CancelOperationPaymentTerminalDeviceRequest	This request is called when a user selects the <b>Cancel</b> button in the payment view dialog box after the payment is initiated but before the payment is completed on the payment terminal.
CapturePaymentTerminalDeviceRequest	This request is called for each payment line when the whole amount in the cart is paid but before the sales transaction is concluded.
VoidPaymentTerminalDeviceRequest	This request is called when a payment line is voided in the cart.
RefundPaymentTerminalDeviceRequest	This request is called when a refund is issued.
FetchTokenPaymentTerminalDeviceRequest	This request is called to fetch a payment token to support deferred payments for customer orders.
EndTransactionPaymentTerminalDeviceRequest	This request is called when the sales transaction is concluded and all payments have been captured.
ClosePaymentTerminalDeviceRequest	This request is called after the sales transaction is concluded. It is used to close the connection to the payment terminal.
ActivateGiftCardPaymentTerminalRequest	This request is called when an external gift card is being activated through the POS.
AddBalanceToGiftCardPaymentTerminalRequest	This request is called when a balance is being added to an external gift card.
GetGiftCardBalancePaymentTerminalRequest	This request is called when the balance on the gift card is being retrieved.
GetPrivateTenderPaymentTerminalDeviceRequest	This request is called when gift card numbers are retrieved from the payment terminal for gift card flows (for example, Issue gift card, Pay by gift card, or Add to gift card).

REQUEST CLASS	PAYMENT FLOW DESCRIPTION
ExecuteTaskPaymentTerminalDeviceRequest	This extension request can be invoked from the POS through customizations. It is used to enable additional payment-related flows.
GetTransactionReferencePaymentTerminalDeviceRequest	This request is called to check the correlation ID. It is used for duplicate payment protection.
GetTransactionByTransactionReferencePaymentTerminalDeviceRequest	This request is used to obtain the previous transaction by correlation ID.
CashoutGiftCardPaymentTerminalRequest	This request is called when a the cash out gift card operation is executed from the POS.

#### OpenPaymentTerminalDeviceRequest

Signature

```
public OpenPaymentTerminalDeviceRequest(string token, string deviceName, SettingsInfo terminalSettings,
PeripheralConfiguration deviceConfig, ExtensionTransaction extensionTransactionProperties);
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
deviceName	The name of the device, as defined on the <b>POS hardware profile</b> page in the client.
terminalSettings	The set of payment terminal-specific configuration properties that are defined in the client, such as the minimum amount for signature capture and the debit cash-back limit.
deviceConfig	The set of payment terminal-specific configuration properties in the form of name/value pairs, such as the IP address and port in the case of network devices.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

#### BeginTransactionPaymentTerminalDeviceRequest

Signature

```
public BeginTransactionPaymentTerminalDeviceRequest(string token, string paymentConnectorName, string
merchantInformation, string invoiceNumber, bool isTestMode, ExtensionTransaction
extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.



VARIABLE	DESCRIPTION
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if you plan to integrate with payment flows that use the <b>IPaymentProcessor</b> interface.
merchantInformation	The merchant information that is defined on the <b>POS hardware profile</b> page in the client.
invoiceNumber	The unique invoice number that the POS generates to track the sales transaction.
isTestMode	A value that indicates whether the payment connector is being used in testing mode.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**LockPaymentTerminalDeviceRequest**  
Signature

```
public LockPaymentTerminalDeviceRequest(string clientDeviceNumber, string deviceType, string deviceName,
bool isExclusive, bool isOverride)
```

Variables

VARIABLE	DESCRIPTION
clientDeviceNumber	The unique POS device number that is acquiring the lock.
deviceType	The device type that the lock is acquired for as configured in the POS hardware profile (such as "Windows").
deviceName	The device type that the lock is acquired for as configured in the POS hardware profile (such as "MOCKPAYMENTTERMINAL").
isExclusive	Determines whether the lock that is acquired is exclusive.
isOverride	Determines whether this request will override any existing lock.

**UpdateLineItemsPaymentTerminalDeviceRequest**  
Signature

```
public UpdateLineItemsPaymentTerminalDeviceRequest(string token, string totalAmount, string taxAmount,
string discountAmount, string subTotalAmount, IEnumerable<ItemInfo> items, ExtensionTransaction
extensionTransactionProperties = null)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
totalAmount	The total amount on the current sales transaction.

VARIABLE	DESCRIPTION
taxAmount	The tax amount on the current sales transaction.
discountAmount	The discount amount on the current sales transaction.
subTotalAmount	The subtotal amount on the current sales transaction.
items	The list of line items to show.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**AuthorizePaymentTerminalDeviceRequest**  
Signature

```
public AuthorizePaymentTerminalDeviceRequest(string token, string paymentConnectorName, decimal amount,
string currency, TenderInfo tenderInfo, string voiceAuthorization, bool isManualEntry,
Retail.PaymentSDK.Portable.PaymentTransactionReferenceData transactionReferencedata, bool isTippingEnabled,
ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if there is an integration with payment flows that use the <b>IPaymentProcessor</b> interface.
amount	The amount to authorize.
currency	The currency for the amount to authorize.
tenderInfo	The card information that is sent from the POS that is retrieved from an external source (if an external source is present).
voiceAuthorization	The voice approval code that is sent from the POS if voice authorization is required.
isManualEntry	A value that defines whether the card number was entered manually.
transactionReferenceData	Merchant's transaction reference that is sent to the processor.
isTippingEnabled	Indicates if tipping is supported by the payment connector. Optional. The default value is <b>false</b> .
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs. Optional. The default value is <b>null</b> .

Response

The **AuthorizePaymentCardPaymentResponse** response object must be returned when the

**AuthorizePaymentTerminalDeviceRequest** request is handled. The response must contain an instance of the **PaymentInfo** object that has the following required properties.

PROPERTY	DESCRIPTION
ApprovedAmount	The amount that was approved for the transaction. Includes tip amount if tipping is enabled.
CardNumberMasked	The masked credit card number. The value must contain at least the first digit of the credit card to support bin range lookup in the POS. (Most devices return the first six digits and the last four digits.)
CardType	The type of card that was used for the payment (for example, <b>Credit</b> or <b>Debit</b> ) by using the <b>Microsoft.Dynamics.Commerce.HardwareStation.CardPayment.CardType</b> entity.
CashbackAmount	For debit transactions, the cash-back amount that was defined on the payment terminal.
Errors	The list of errors that occurred during the authorize call.
IsApproved	A flag that indicates whether the payment was approved.
PaymentSdkData	The response data that is used to support state between the authorize/refund and capture/void calls or cross-channel payment operations.
TipAmount	The tip amount that was selected by the customer on the device.

The **PaymentSdkData** property must contain the following data.

NAMESPACE	NAME	DESCRIPTION	SAMPLE VALUE
Connector	ConnectorName	The name of the <b>IPaymentProcessor</b> interface that is used for the transactions, as described in the "Write a payment processor" section later in this topic.	
AuthorizationResponse	Properties	The list of authorization responses.	See the next table.

The **Properties** field of the **PaymentSdkData** property must contain the following fields.

NAMESPACE	NAME	DESCRIPTION	SAMPLE VALUE
AuthorizationResponse	ApprovedAmount	The amount that was approved for the transaction.	28.08m
AuthorizationResponse	AvailableBalance	The available balance on the card.	100.00m

NAMESPACE	NAME	DESCRIPTION	SAMPLE VALUE
AuthorizationResponse	ApprovalCode	The approval code for the transaction.	Z123456
AuthorizationResponse	ProviderTransactionId	The transaction identifier of the payment provider.	123456789
AuthorizationResponse	AuthorizationResult	The result of the authorization call.	AuthorizationResult.Success.ToString()
AuthorizationResponse	ExternalReceipt	The external receipt data from the payment provider.	<ReceiptData> ... </ReceiptData>
AuthorizationResponse	TerminalId	The unique identifier of the terminal that handled the payment.	000001

The following example shows how to construct the **PaymentSdkData** object.

```
List<PaymentProperty> paymentSdkProperties = new List<PaymentProperty>();
paymentSdkProperties.Add(new PaymentProperty(GenericNamespace.Connector, ConnectorProperties.ConnectorName,
"TestConnector"));

List<PaymentProperty> paymentSdkAuthorizationProperties = new List<PaymentProperty>();
paymentSdkAuthorizationProperties.Add(new PaymentProperty(GenericNamespace.AuthorizationResponse,
AuthorizationResponseProperties.ApprovedAmount, 28.08m));
paymentSdkAuthorizationProperties.Add(new PaymentProperty(GenericNamespace.AuthorizationResponse,
AuthorizationResponseProperties.AvailableBalance, 100.00m));
paymentSdkAuthorizationProperties.Add(new PaymentProperty(GenericNamespace.AuthorizationResponse,
AuthorizationResponseProperties.ApprovalCode, "Z123456"));
paymentSdkAuthorizationProperties.Add(new PaymentProperty(GenericNamespace.AuthorizationResponse,
AuthorizationResponseProperties.ProviderTransactionId, "123456789"));
paymentSdkAuthorizationProperties.Add(new PaymentProperty(GenericNamespace.AuthorizationResponse,
AuthorizationResponseProperties.AuthorizationResult, AuthorizationResult.Success.ToString()));
paymentSdkAuthorizationProperties.Add(new PaymentProperty(GenericNamespace.AuthorizationResponse,
TransactionDataProperties.TerminalId, "000001"));

paymentSdkProperties.Add(new PaymentProperty(GenericNamespace.AuthorizationResponse,
AuthorizationResponseProperties.Properties, paymentSdkAuthorizationProperties.ToArray()));

string paymentSdkData = PaymentProperty.ConvertPropertyArrayToXML(paymentSdkProperties.ToArray());
```

If the payment terminal returns a receipt, you can print it through the POS by setting the following data on the **ExternalReceipt** object that was described earlier.

```
<ReceiptData>
  <Receipt Type='Customer'>
    <Line>Line 1 of receipt.</Line>
    <Line>Line 2 of receipt.</Line>
  </Receipt>
  <Receipt Type='Merchant'>
    <Line>Line 1 of receipt.</Line>
    <Line>Line 2 of receipt.</Line>
  </Receipt>
</ReceiptData>
```

#### Other considerations

If the payment terminal handles the authorize and capture requests in a single call (that is, if *immediate capture* occurs), and the cashier wants to void the transaction, the payment terminal must support reversal of an

immediate capture. When an immediate capture is voided, if the void request fails, the cashier will be asked whether they want to locally void the payment. If the cashier selects **Yes**, the tender is voided only in the POS. No call is made to the payment terminal to void the payment. Basically, this behavior lets the cashier unblock the POS if it can no longer void the payment on the payment terminal. However, this behavior can cause issues, because a lock lasts for three to five days, until the bank reverses it, but the payment is made for immediate capture. Therefore, duplicate payments can occur.

**CancelOperationPaymentTerminalDeviceRequest**  
Signature

```
public CancelOperationPaymentTerminalDeviceRequest(string token)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.

**CapturePaymentTerminalDeviceRequest**  
Signature

```
public CapturePaymentTerminalDeviceRequest(string token, decimal amount, string currency, string paymentPropertiesXml, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
amount	The amount to capture.
currency	The currency for the amount to capture.
paymentPropertiesXml	The content of the <b>PaymentSdkData</b> object that is returned by the <b>AuthorizePaymentTerminalDeviceRequest</b> or <b>RefundPaymentTerminalDeviceRequest</b> request, and that is used to support stateful properties between the requests.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

Other considerations

If the payment terminal handles the authorize and capture requests in a single call, the **CapturePaymentTerminalDeviceRequest** request should be a no-op and should immediately return.

If the payment terminal requires state from the authorize requests to handle the capture call, the properties should be stored in the **PaymentSdkData** object of the **AuthorizePaymentTerminalDeviceResponse** request that is described earlier, and passed through the **paymentPropertiesXml** variable of the **CapturePaymentTerminalDeviceRequest** request.

**VoidPaymentTerminalDeviceRequest**  
Signature

```
public VoidPaymentTerminalDeviceRequest(string token, string paymentConnectorName, decimal amount, string
currency, TenderInfo tenderInfo, string paymentPropertiesXml, ExtensionTransaction
extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if there is an integration with payment flows that use the <b>IPaymentProcessor</b> interface.
amount	The amount for the payment to void.
currency	The currency for the payment to void.
tenderInfo	The card information that is sent from the POS that is retrieved from an external source (if an external source is present).
paymentPropertiesXml	The content of the <b>PaymentSdkData</b> object that is returned by the <b>AuthorizePaymentTerminalDeviceRequest</b> or <b>RefundPaymentTerminalDeviceRequest</b> request, and that is used to support stateful properties between the requests.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**RefundPaymentTerminalDeviceRequest**  
Signature

```
public RefundPaymentTerminalDeviceRequest(string token, string paymentConnectorName, TenderInfo tenderInfo,
decimal amount, string currency, bool isManualEntry, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if there is an integration with payment flows that use the <b>IPaymentProcessor</b> interface.
tenderInfo	The card information that is sent from the POS that is retrieved from an external source (if an external source is present).
amount	The amount to refund.

VARIABLE	DESCRIPTION
currency	The currency for the amount to refund.
isManualEntry	A value that defines whether the card number was entered manually.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**FetchTokenPaymentTerminalDeviceRequest**  
Signature

```
public FetchTokenPaymentTerminalDeviceRequest(string token, bool isManualEntry, ExtensionTransaction
extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
isManualEntry	A value that defines whether the card number was entered manually.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**EndTransactionPaymentTerminalDeviceRequest**  
Signature

```
public EndTransactionPaymentTerminalDeviceRequest(string token, ExtensionTransaction
extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**ClosePaymentTerminalDeviceRequest**  
Signature

```
public ClosePaymentTerminalDeviceRequest(string token, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**ActivateGiftCardPaymentTerminalRequest**  
Signature

```
public ActivateGiftCardPaymentTerminalRequest(string token, string paymentConnectorName, decimal amount, string currencyCode, TenderInfo tenderInfo, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if there is an integration with payment flows that use the <b>IPaymentProcessor</b> interface.
amount	The initial amount to add to the gift card during activation.
currency	The currency for the initial amount to add to the gift card during activation.
tenderInfo	The card information that is sent from the POS that is retrieved from an external source (if an external source is present).
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**AddBalanceToGiftCardPaymentTerminalRequest**  
Signature

```
public AddBalanceToGiftCardPaymentTerminalRequest(string token, string paymentConnectorName, decimal amount, string currencyCode, TenderInfo tenderInfo, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if there is an integration with payment flows that use the <b>IPaymentProcessor</b> interface.
amount	The amount to add to the gift card.
currency	The currency for the amount to add to the gift card balance.
tenderInfo	The card information that is sent from the POS that is retrieved from an external source (if an external source present).
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**GetGiftCardBalancePaymentTerminalRequest**



Signature

```
public GetGiftCardBalancePaymentTerminalRequest(string token, string paymentConnectorName, string currencyCode, TenderInfo tenderInfo, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if there is an integration with payment flows that use the <b>IPaymentProcessor</b> interface.
currency	The currency to retrieve the gift card balance in.
tenderInfo	The card information that is sent from the POS that is retrieved from an external source (if an external source present).
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**GetPrivateTenderPaymentTerminalDeviceRequest**

Signature

```
public GetPrivateTenderPaymentTerminalDeviceRequest(string token, decimal amount, bool declined, bool isSwipe, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
amount	The amount that is set on the POS. (Typically, this variable is used to show the amount on the payment terminal when the card number is retrieved.)
declined	This variable is obsolete.
isSwipe	A value that determines whether the card number should be retrieved through a swipe or manual entry on the payment terminal.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**ExecuteTaskPaymentTerminalDeviceRequest**

Signature

```
public ExecuteTaskPaymentTerminalDeviceRequest(string token, string task, ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
token	The unique token value that is generated when the payment terminal is initially locked for the transaction.
task	The unique identifier for the task that is being run.
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

**GetTransactionReferencePaymentTerminalDeviceRequest**  
Signature

```
public GetTransactionReferencePaymentTerminalDeviceRequest(string lockToken, string posTerminalId, string eftTerminalId)
```

Variables

VARIABLE	DESCRIPTION
locktoken	Gets the unique lock token that was generated when the payment terminal was initially locked for the transaction.
posTerminalId	Gets the POS terminal ID associated with the lock token.
extensionTransactionProperties	Gets the EFT terminal ID associated with the transaction and lock token.

**GetTransactionByTransactionReferencePaymentTerminalDeviceRequest**  
Signature

```
public GetTransactionByTransactionReferencePaymentTerminalDeviceRequest(string lockToken, Retail.PaymentSDK.Portable.PaymentTransactionReferenceData transactionReferenceData)
```

Variables

VARIABLE	DESCRIPTION
locktoken	Gets the unique lock token that was generated when the payment terminal was initially locked for the transaction.
Retail.PaymentSDK.Portable.PaymentTransactionReferenceData transactionReferenceData	Gets reference data for the for payment transactions in case the correlation ID is out of sync.

**CashoutGiftCardPaymentTerminalRequest**  
Signature

```
public CashoutGiftCardPaymentTerminalRequest(
    string paymentConnectorName,
    decimal amount,
    string currencyCode,
    TenderInfo tenderInfo,
    ExtensionTransaction extensionTransactionProperties)
```

Variables

VARIABLE	DESCRIPTION
----------	-------------

VARIABLE	DESCRIPTION
paymentConnectorName	The name of the payment connector that is used as part of the payment flow. This variable is used if there is an integration with payment flows that use the IPaymentProcessor interface.
amount	The amount gift card cash out request.
currencyCode	The currency for the gift card cash out request.
tenderinfo	The card information that is sent from the POS that is retrieved from an external source (if an external source is present).
extensionTransactionProperties	The set of extension configuration properties in the form of name/value pairs.

### State in the payment connector

The payment connector can be hosted as part of the dllhost.exe process when it's hosted through the in-process Hardware Station inside the POS. Alternatively, the payment connector can be hosted as a w3wp.exe process when it's hosted in the Hardware Station that is based on Microsoft Internet Information Services (IIS). In some circumstances, both processes can be terminated or stop responding between or during payment flows. Therefore, we recommend that payment connectors not have state dependencies, and that they be able to recover if they are terminated at any point during the payment flow–related requests that are described earlier.

### Configure the payment connector in the Hardware Station config

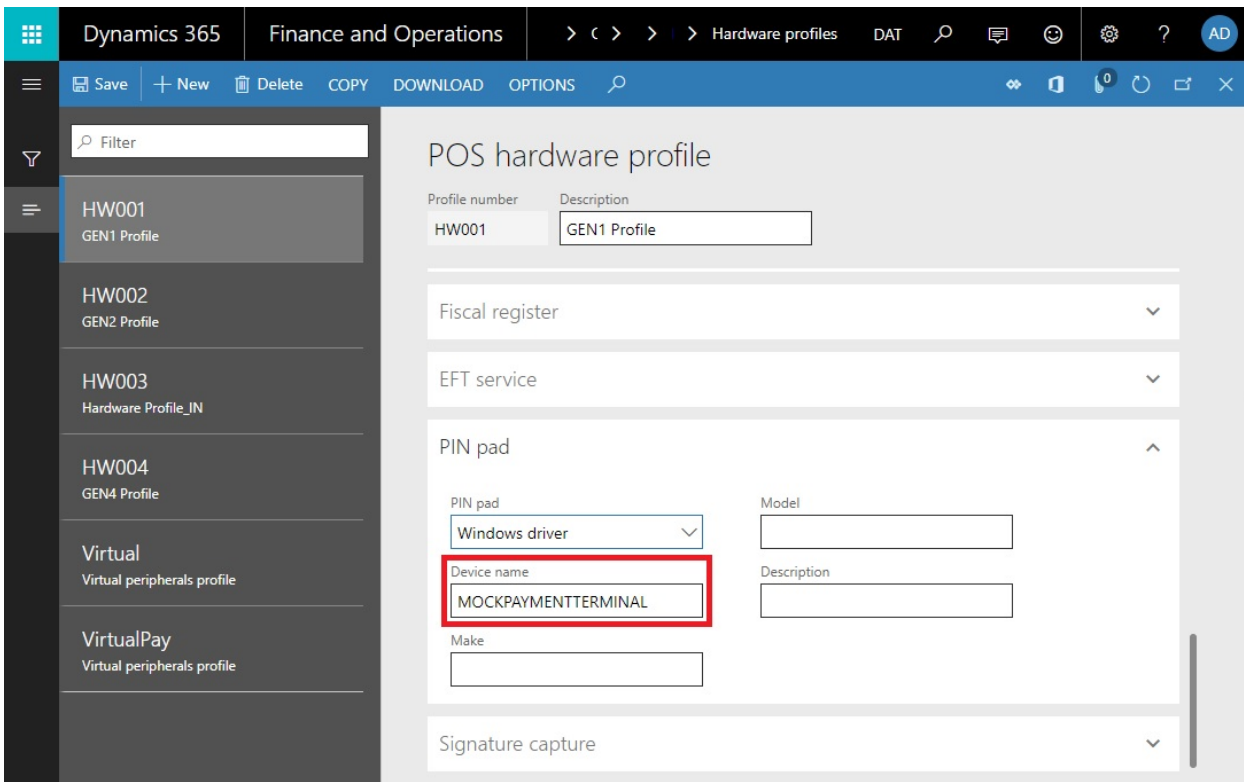
To help guarantee that the Hardware Station loads the payment connector, you must set the corresponding assembly reference in the **HardwareStation.Extension.config** file that is located in the **Assets** folder in the Retail SDK.

```
<?xml version="1.0" encoding="utf-8"?>
<hardwareStationExtension>
  <composition>
    <!--
      Register your own assemblies or types here. The following example registers NewPeripheralDevice
      (and all its request handlers). Any other services are not being overridden:

      <add source="type"
        value="Contoso.Commerce.HardwareStation.NewPeripheralDevice,
Contoso.Commerce.HardwareStation.NewPeripheralDevice" />
      <add source="assembly"
        value="Contoso.Commerce.HardwareStation.NewPeripheralDevice" />
    -->
    <add source="assembly" value="Contoso.Commerce.HardwareStation.PaymentSample" />
  </composition>
</hardwareStationExtension>
```

### Configure the payment connector on the POS hardware profile page in the client

To determine the correct payment connector that should be loaded on the POS, you must set the value of the **PaymentTerminalDevice** property in the **Device name** field on the **PIN pad** FastTab of the **POS hardware profile** page in the client, as shown in the following illustration.



## Write a payment processor

Payment processes are usually used only if a direct connection to a payment gateway is established. This scenario most often occurs in card-not-present sales transactions or more complex card-present scenarios. Additionally, the payment processor is used to process the merchant properties that are configured through the POS hardware profile page in the client.

### NOTE

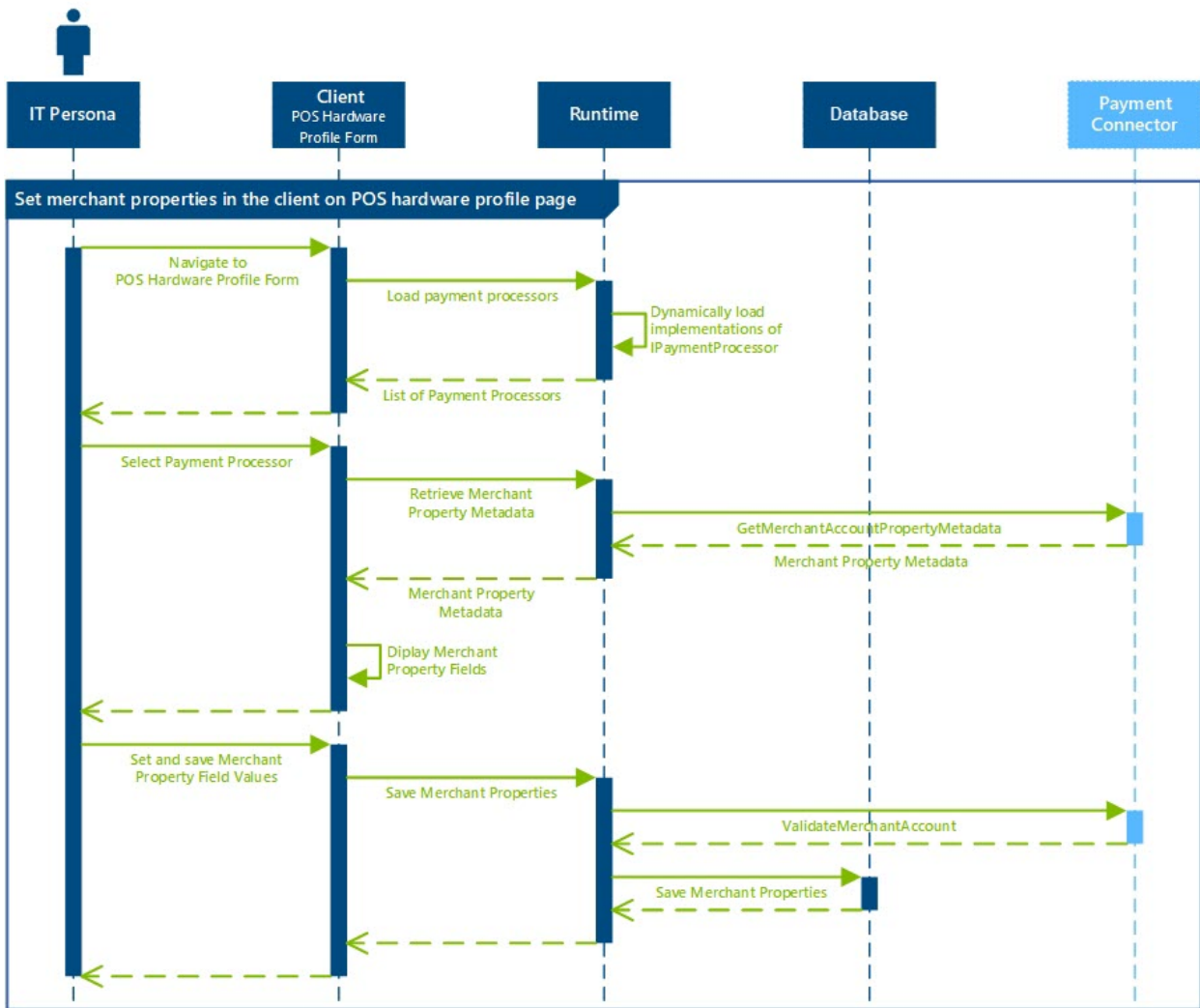
The payment processor is currently required, even if all payment requests are handled directly through the payment terminal and no merchant properties must be set through the POS. For more information about implementing the `IPaymentProcessor` interface, read the [Implementing a payment connector and payment device](#) white paper.

### Understanding the merchant properties flows

The following sections describe how the merchant properties are set on the POS hardware profile page in the client, and how they are passed to the payment connector during payment flows on the POS.

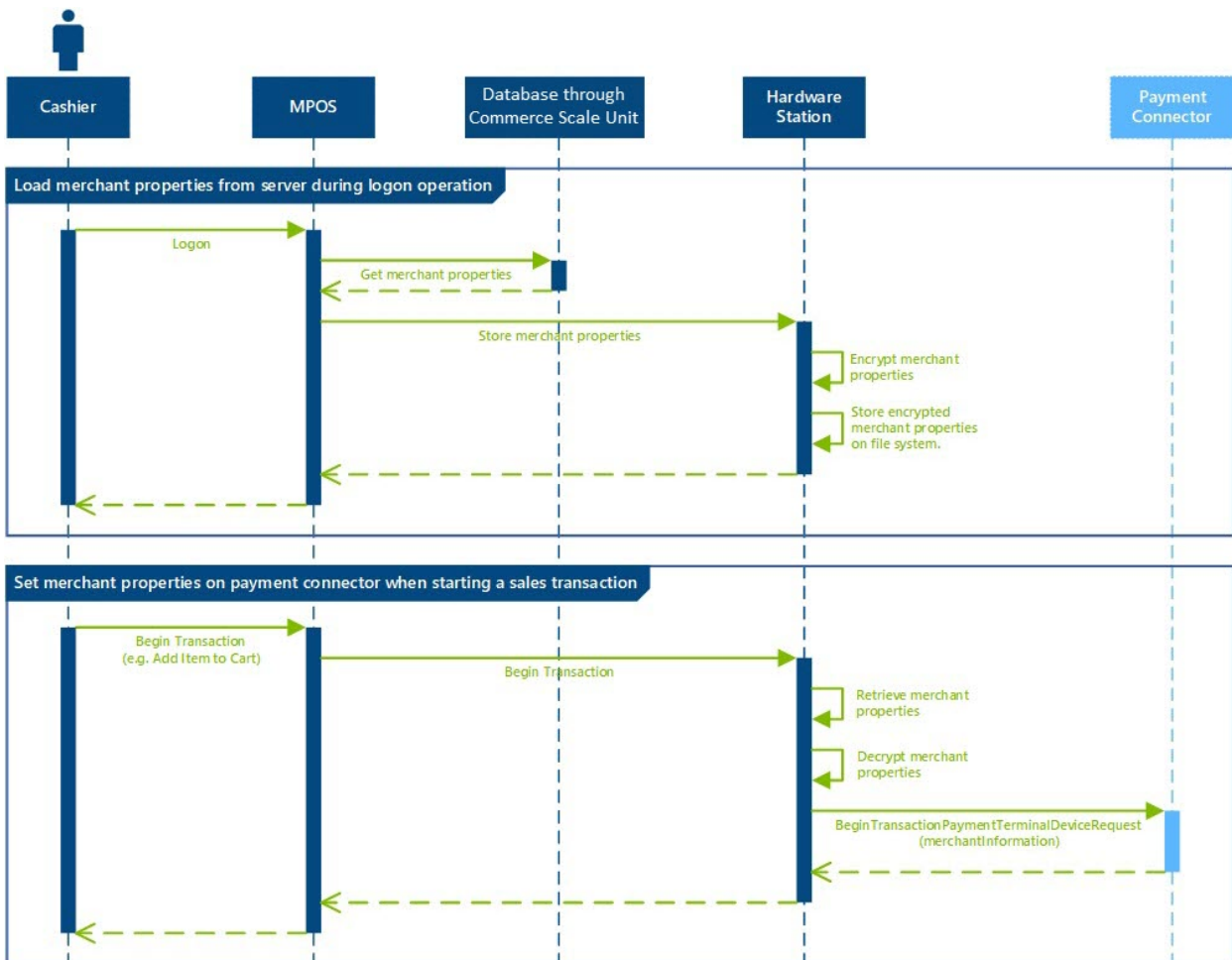
#### Set merchant properties on the POS hardware profile page in the client

The following illustration shows how the merchant properties are set through the POS hardware profile page in the client. To enable the merchant properties to be set, the `IPaymentProcessor` interface that is defined in the `Microsoft.Dynamics.Retail.PaymentSDK` library must be implemented. Two interface methods are required: `GetMerchantAccountPropertyMetadata` and `ValidateMerchantAccount`.



**Set merchant properties on payment connector during POS sales transaction**

The following illustration shows how the merchant properties are retrieved from the database through the Commerce Scale Unit and passed to the payment connector during the `BeginTransactionPaymentTerminalDeviceRequest` request.



### Implement the IPaymentProcessor interface

To handle merchant properties that are related to payment flows, the `IPaymentProcessor` interface that is defined in the `Microsoft.Dynamics.Retail.PaymentsSDK` library must be implemented. The following example shows how to implement the two required interface methods, `GetMerchantAccountPropertyMetadata` and `ValidateMerchantAccount`. Other interface methods can be left blank (for example, they can return `FeatureNotSupportedException`).

```

/// <summary>
/// SampleConnector class (Portable Class Library version).
/// </summary>
public class SampleConnector : IPaymentProcessor
{
    /// <summary>
    /// GetMerchantAccountPropertyMetadata returns the merchant account properties need by the payment
    provider.
    /// </summary>
    /// <param name="request">Request object.</param>
    /// <returns>
    /// Response object.
    /// </returns>
    public Response GetMerchantAccountPropertyMetadata(Request request)
    {
        string methodName = "GetMerchantAccountPropertyMetadata";

        // Check null request
        List<PaymentError> errors = new List<PaymentError>();
        if (request == null)
        {
            errors.Add(new PaymentError(ErrorCode.InvalidRequest, "Request is null."));
            return PaymentUtilities.CreateAndLogResponseForReturn(methodName, this.Name, Platform, locale:
            null, properties: null, errors: errors);
        }
    }
}

```

```

// Prepare response
List<PaymentProperty> properties = new List<PaymentProperty>();
PaymentProperty property;
property = new PaymentProperty(
    GenericNamespace.MerchantAccount,
    MerchantAccountProperties.AssemblyName,
    this.GetAssemblyName());
property.SetMetadata("Assembly Name:", "The assembly name of the test provider", false, true, 0);
properties.Add(property);

Response response = new Response();
response.Locale = request.Locale;
response.Properties = properties.ToArray();
if (errors.Count > 0)
{
    response.Errors = errors.ToArray();
}

PaymentUtilities.LogResponseBeforeReturn(methodName, this.Name, Platform, response);
return response;
}

/// <summary>
/// ValidateMerchantAccount the passed merchant account properties with the payment provider.
/// </summary>
/// <param name="request">Request object to validate.</param>
/// <returns>
/// Response object.
/// </returns>
public Response ValidateMerchantAccount(Request request)
{
    string methodName = "ValidateMerchantAccount";

    // Convert request
    ValidateMerchantAccountRequest validateRequest = null;
    try
    {
        validateRequest = ValidateMerchantAccountRequest.ConvertFrom(request);
    }
    catch (SampleException ex)
    {
        return PaymentUtilities.CreateAndLogResponseForReturn(methodName, this.Name, Platform, locale:
request == null ? null : request.Locale, properties: null, errors: ex.Errors);
    }

    // Validate merchant account
    List<PaymentError> errors = new List<PaymentError>();
    ValidateMerchantProperties(validateRequest, errors);
    if (errors.Count > 0)
    {
        return PaymentUtilities.CreateAndLogResponseForReturn(methodName, this.Name, Platform,
validateRequest.Locale, errors);
    }

    // Create response
    var validateResponse = new ValidateMerchantAccountResponse(validateRequest.Locale,
validateRequest.ServiceAccountId, this.Name);

    // Convert response and return
    Response response = ValidateMerchantAccountResponse.ConvertTo(validateResponse);
    PaymentUtilities.LogResponseBeforeReturn(methodName, this.Name, Platform, response);
    return response;
}
}

```

### Required merchant property fields

The following table shows the required merchant property fields that must be set as part of the

## GetMerchantAccountPropertyMetadata method.

NAMESPACE	NAME	SAMPLE VALUE*
MerchantAccount	PortableAssemblyName	Contoso.Microsoft.PaymentsSample
MerchantAccount	ServiceAccountId	f35989c8-e571-4de1-862a-996c82a2e6b6
MerchantAccount	SupportedCurrencies	AUD;BRL;CAD;CHF;CNY;CZK;DKK;EUR;GBP;HKD;HUF;INR;JPY;KPW;KRW;MXN;NOK;NZD;PLN;SEK;SGD;TWD;USD;ZAR
MerchantAccount	SupportedTenderTypes	Visa;MasterCard;Amex;Discover;Debit

\* You **must** replace the sample values in this column with unique values for your own payment processor.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).



# Deploy payment connectors

2/18/2021 • 5 minutes to read • [Edit Online](#)

## Overview

This topic guides IT professionals or value-added resellers (VARs) through the process of deploying a payment connector to the appropriate components. We assume that the payment connector has been implemented and tested by the payment provider or the payment independent software vendor (ISV), and that it's ready for validation and subsequent production deployment in a customer environment.

This topic doesn't include information about how to package a payment connector by using the Retail software development kit (SDK). For information about how to download the SDK, see [Retail software development kit \(SDK\) architecture](#). For guidelines about how to package a payment connector, see the Retail SDK packaging document in the downloaded SDK.

This topic also doesn't include information about how to deploy the payment web application, payment front-end processor, or payment back-end processor, because those applications are managed by payment providers or payment ISVs. If you're using Microsoft Dynamics AX 7.0 (February 2016), you must apply KB 3183058 before you create the Commerce deployable package.

## Payment packaging folder

A payment provider or a payment ISV creates a payment connector. The payment connector will include some or all of the following folders:

### NOTE

You can find these folders in `\RetailSDK\PaymentExternals`.

- **IPaymentProcessor Assemblies** – This folder contains the assembly that implements the IPaymentProcessor interface, and its dependent assemblies.
- **Payment Web Files** – This folder contains the callback HTML, JavaScript, or CSS files that are required in order to enable the payment accepting page. Payment connector developers will provide these web files if their payment accepting page requires them.
- **IPaymentDevice Assemblies** – This folder contains the assembly that implements the IPaymentDevice interface and payment request handlers, and the interface's dependent assemblies. These assemblies are used in Hardware station and Modern Point of Sale (Modern POS) to communicate with payment terminal devices, such as VeriFone MX925. If you don't have a payment terminal device, you don't need these files.

To package the payment connector files, copy the payment assemblies to the folder in `\RetailSDK\PaymentExternals`. After the payment assemblies are copied, use **msbuild** from the root of the Retail SDK folder to generate the deployable packages. After the **msbuild** operation is completed, you can find the deployable package in `\RetailSDK\Packages\RetailDeployablePackage`.

- **Retail deployable package** – This package includes payments plus all the other channel extension components. This is this combined package for all extension components. RetailDeployablePackage includes the payment connectors for the following components:
  - Commerce Scale Unit (CSU)
  - Self-service installer, which enables installation of the following:

- Hardware station
- Modern POS
- Commerce scale unit (self-hosted)

#### NOTE

In releases earlier than 10.0.10, the RetailDeployablePackage can be deployed to both AOS and CSU. For later releases, the RetailDeployablePackage can be deployed only to CSU. To deploy the payment connector to AOS application version 10.0.10 and later, follow the information in [Create payment packaging for Application Explorer for self-service deployment](#).

### Upload and deploy deployable packages

1. Open your LCS project page.
2. In the **More tools** section, click **Asset library**.
3. Select **Software package**, and then click the plus sign (+).
4. Enter a name and description, and select the correct package type, as described in the following table.

DEPLOYABLE PACKAGE	DEPLOYABLE PACKAGE TYPE IN LCS
RetailDeployablePackage.zip	Combined commerce package

5. Click **Upload**.
6. Select the zipped package, upload it, and then click **Confirm**.

After you've uploaded your deployable packages to the LCS asset library, you can deploy them to your environments through the LCS portal. After you've validated your deployment in your sandbox environment, you can create a service request to deploy it to your production environment. For more information, see [Apply a deployable package](#).

#### Download and run installers on client computers

The self-service package contains the installers for both Hardware station and Modern POS. After your deployable packages have been applied to your environment, you can download the updated Hardware station and Modern POS installers. For information about how to download Hardware station and Modern POS, and install them on client computers, see [Configure, install, and activate Modern POS \(MPOS\)](#).

## Manual deployment

This section describes how to manually deploy a payment connector. You can use a manual deployment to test locally in a developer environment. This developer environment can be either cloud-hosted or on a downloadable virtual hard disk (VHD). These steps are not applicable for sandbox or production environments.

#### Put the payment connector assemblies and files in the correct locations

Payment connectors are pluggable. In a development environment, you can put the correct files in the correct locations on the server. The following table shows which files should go in which location.

CONNECTOR	CONTENTS OF THE IPAYMENTPROCESSOR ASSEMBLIES FOLDER	CONTENTS OF THE PAYMENT WEB FILES FOLDER	CONTENTS OF THE IPAYMENTDEVICE ASSEMBLIES FOLDER
Application Object Server (AOS)	<Aos.PackageDirectory>/bin/Connectors/ <Aos.WebRoot>/bin/	<Aos.WebRoot>/Connectors/	Not applicable

CONNECTOR	CONTENTS OF THE IPAYMENTPROCESSOR ASSEMBLIES FOLDER	CONTENTS OF THE PAYMENT WEB FILES FOLDER	CONTENTS OF THE IPAYMENTDEVICE ASSEMBLIES FOLDER
Commerce Scale Unit	<RS.WebRoot>/bin/	Not applicable	Not applicable
Cloud POS	Not applicable	<CPOS.WebRoot>/Connectors/	Not applicable
Remote Hardware Station (Internet Information Services [IIS])	<HWS.WebRoot>/bin/	Not applicable	<HWS.WebRoot>/bin/
Modern POS Local Hard Station (Information Protection and Control [IPC])	<MPOS.AppRoot>/ClientBroker/	Not applicable	<MPOS.AppRoot>/ClientBroker/
E-commerce	Not applicable	<ECOM.WebRoot>/Connectors/	Not applicable

Here is a key to the preceding table:

- <Aos.PackageDirectory> is the package directory of AOS. You can find the path from the web.config file for AOS (key = Aos.PackageDirectory).
- <Aos.WebRoot> is the web application root of AOS.
- <RS.WebRoot> is the web application root of Commerce Scale Unit.
- <HWS.WebRoot> is the web application root of a remote Hardware Station.
- <MPOS.AppRoot> is the app installation folder of Modern POS (for example, C:\Program Files (x86)\Microsoft Dynamics AX70\Retail Modern POS).
- <ECOM.WebRoot> is the web application root of an e-commerce website.

The Payment Web Files folder usually contains a subfolder. Be sure to copy the whole subfolder to the target locations.

## Use a payment connector with an e-commerce site

E-commerce sites aren't deployed in LCS-managed environments. You should work with your partner to decide how to host an e-commerce site. If the payment connector requires payment web files, you must deploy those web files to your e-commerce site. If your payment connector doesn't require payment web files, no additional steps are required. For information about how to deploy payment web files to an e-commerce site, see the [Put the payment connector assemblies and files in the correct locations](#) section earlier in this topic.

## Additional resources

[Guide to implementing a payment connector and a payment device](#)

[Create deployable packages](#)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Incremental capture for back-office invoicing

2/18/2021 • 4 minutes to read • [Edit Online](#)

## IMPORTANT

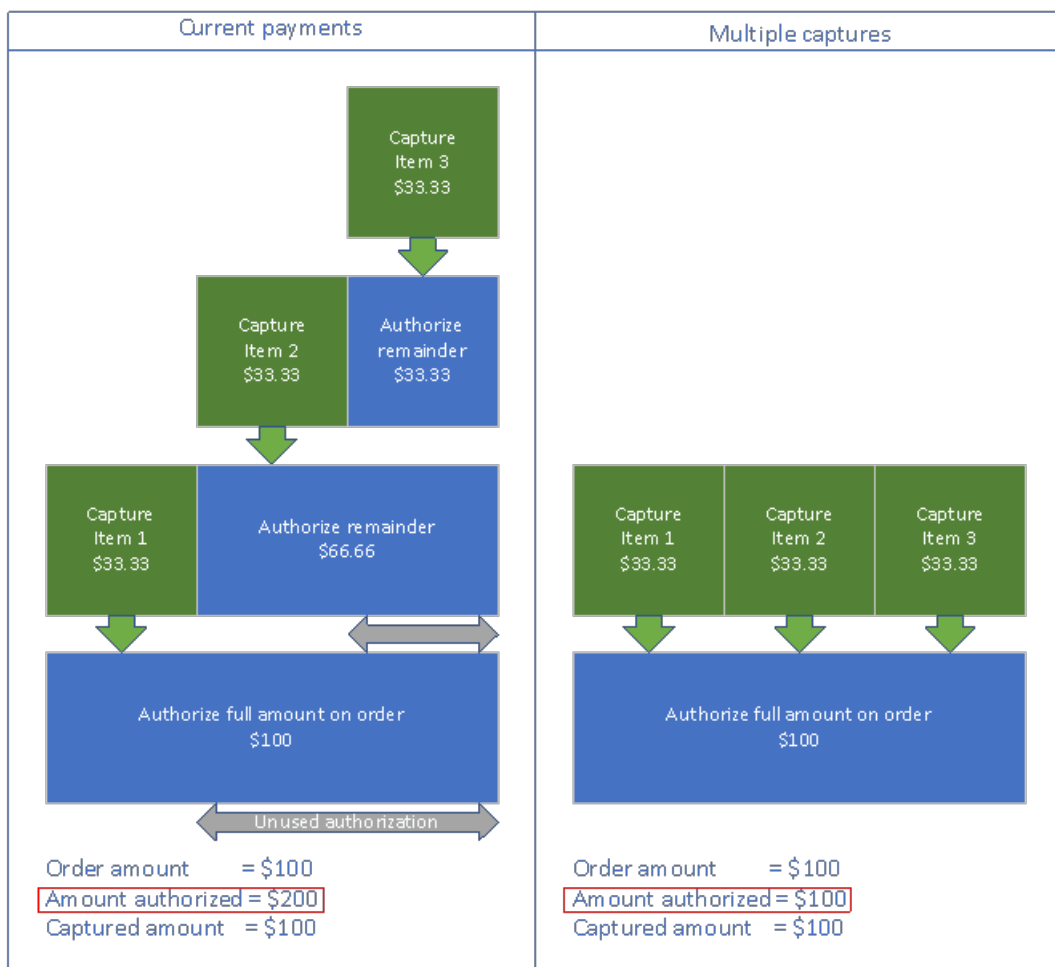
Some or all of the functionality noted in this topic is available as part of a preview release. The content and the functionality are subject to change. For more information about preview releases, see [One version service updates FAQ](#).

This topic describes how the payments software development kit (SDK) supports incremental capture as part of order invoicing. In incremental capture, payment processing supports fulfillment of an order that has multiple invoices. Specifically, as invoices are processed against the order, payment capture requests reference the original payment authorization instead of getting a new payment authorization when previous payment captures have caused the balance due to change.

## Overview

Many merchants fulfill orders in multiple shipments. In Microsoft Dynamics 365 Commerce version 10.0.12 and earlier, the out-of-box process for handling credit card payments for orders that are fulfilled over multiple shipments is to capture payments as those shipments are invoiced and then get a new payment authorization for the balance due for the remaining items that must be shipped. This process ensures that payment capture can be consistently supported across payment processors. However, it also has some downsides. Specifically, when an authorization is partially captured, and then a new authorization is created for the balance due, the old authorization and new authorization might overlap. In this case, open authorizations that exceed the order total might appear against customer payment cards. Therefore, authorizations might exceed open balances that are available for credit cards, or card issuers might block cards from being processed because of suspicion of fraud.

To address these issues, the payments SDK in Commerce version 10.0.13 and later includes incremental capture support for back-office payment captures. Therefore, if a processor supports incremental capture, payment connectors can be updated to capture against a single authorization multiple times over the course of order fulfillment. The following illustration shows the difference between the different payment capture frameworks when multiple captures are done against a single authorization.



In Commerce version 10.0.13, incremental capture is supported as part of back-office invoicing (that is, any invoicing that occurs as part of order fulfillment in the back office). In a later release, support for incremental capture from the point of sale (POS) will be added to the out-of-box Adyen payment connector.

## Turn on incremental capture for back-office invoicing

In the **Feature management** workspace, select the **All** filter to show all available features, and then search for **Extensibility to support incremental credit card capture**. Select the feature, and then select **Enable now**.

## Uptake incremental capture for back-office invoicing

To uptake support for incremental capture as part of back-office invoicing, you must upgrade to the payments SDK that is provided in Commerce version 10.0.13.

## IPaymentReferenceProvider

This version of the payment SDK adds the **IPaymentReferenceProvider** interface. This interface supports using a **PaymentTrackingID** value for each request and response. The **PaymentTrackingID** value can be used to track payment requests. It also ensures that, between back-office invoicing requests and the processor, duplicate requests can be caught before they are re-sent. In this way, it helps prevent duplicate payments.

Here is a sample implementation from the SampleConnector.cs file in the payments SDK.

```

#region ITrackingSupport
/// <summary>
/// Get the payment provider reference to safeguard against duplicate requests.
/// </summary>
/// <param name="command">The payment operation that will use the tracking ID.</param>
/// <param name="amount">The payment transaction amount.</param>
/// <returns>Returns the PaymentTransactionReferenceData.</returns>
/// <remarks>List of supported commands can be seen in the constants defined in <see
cref="Microsoft.Dynamics.Retail.PaymentSDK.Portable.Constants.SupportedCorrelationCommands"/></remarks>
public PaymentTransactionReferenceData GetPaymentReferenceData(string command, decimal amount)
{
    PaymentTransactionReferenceData paymentTransactionReferenceData = new PaymentTransactionReferenceData();
    paymentTransactionReferenceData.Amount = amount;
    paymentTransactionReferenceData.Command = command;
    paymentTransactionReferenceData.IdFromConnector = Guid.NewGuid().ToString();
    paymentTransactionReferenceData.InitiatedDate = DateTime.UtcNow;
    return paymentTransactionReferenceData;
}
#endregion

```

The following sample from the `AuthorizeRequest.cs` file in the payments SDK uses a `PaymentTrackingID` value.

```

authorizeRequest.PaymentTrackingId = PaymentUtilities.GetPropertyStringValue(
    hashtable,
    GenericNamespace.TransactionData,
    TransactionDataProperties.PaymentTrackingId);

```

## Support for multiple captures

If a payment processor supports multiple captures, the `SupportsMultipleCaptures` property for authorization responses from the connector should be set to **True**. If the property is set to **False**, or if it isn't provided, the authorization won't be eligible for incremental capture. In this case, if there is a new balance due after invoicing, a new authorization will be obtained.

Here is a sample from the `AuthorizationResponseProperties.cs` file in the payments SDK.

```

/// <summary>
/// Gets the SupportsMultipleCaptures property.
/// </summary>
public static string SupportsMultipleCaptures
{
    get { return "SupportsMultipleCaptures"; }
}

```

As back-office invoicing processes captures against authorizations that are enabled for multiple capture, the total captured amount is tracked. Capture requests continue to reference the original authorization until it's fully captured.

Authorizations that are expired according to Accounts receivable parameters aren't eligible for incremental capture.

The `SupportsMultipleCaptures` property isn't global. It's specific to the authorization. An environment might have both connectors that support incremental capture and connectors that don't support it.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create custom localized error messages for payment terminal extensions

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic explains how to create custom localized error messages for payment terminal extensions. These custom error messages are most often used so that the payment terminal can give the cashier who is using the point of sale (POS) terminal relevant information about why a specific payment was unsuccessful. For example, the external payment terminal or gateway might return unique identifiers (such as reference numbers or transaction identifiers) that are relevant for troubleshooting with the payment provider.

## Key terms

TERM	DESCRIPTION
Payment connector	An extension library that is written to integrate the POS with a payment terminal.

## Overview

The remaining sections in this topic describe the following steps for creating custom localized error messages for payment terminal extensions:

- [Create custom error messages](#) – This section explains how to create custom error messages in the payment connector that can be returned and shown in the POS.
- [Create localized error messages](#) – This section explains how to localize the error messages in the payment connector that are returned and shown in the POS.

## Create custom error messages

To trigger a custom error message in the POS, you must set the appropriate error in the **Errors** property of the **paymentInfo** object that is passed to the **AuthorizePaymentTerminalDeviceResponse** object. Specifically, you must set the **isLocalized** parameter on the constructor of the **PaymentError** object to **true** to force the POS to use the custom error message instead of the built-in error message for a declined payment.

```
namespace Contoso.Commerce.HardwareStation.PaymentSample
{
    /// <summary>
    /// <c>Simulator</c> manager payment device class.
    /// </summary>
    public class PaymentDeviceSample : INamedRequestHandler
    {
        /// <summary>
        /// Gets the collection of supported request types by this handler.
        /// </summary>
        public IEnumerable<Type> SupportedRequestTypes
        {
            get
            {
                return new[]
                {
                    typeof(AuthorizePaymentTerminalDeviceRequest),
                    ...
                }
            }
        }
    }
}
```



```

    };
}
}

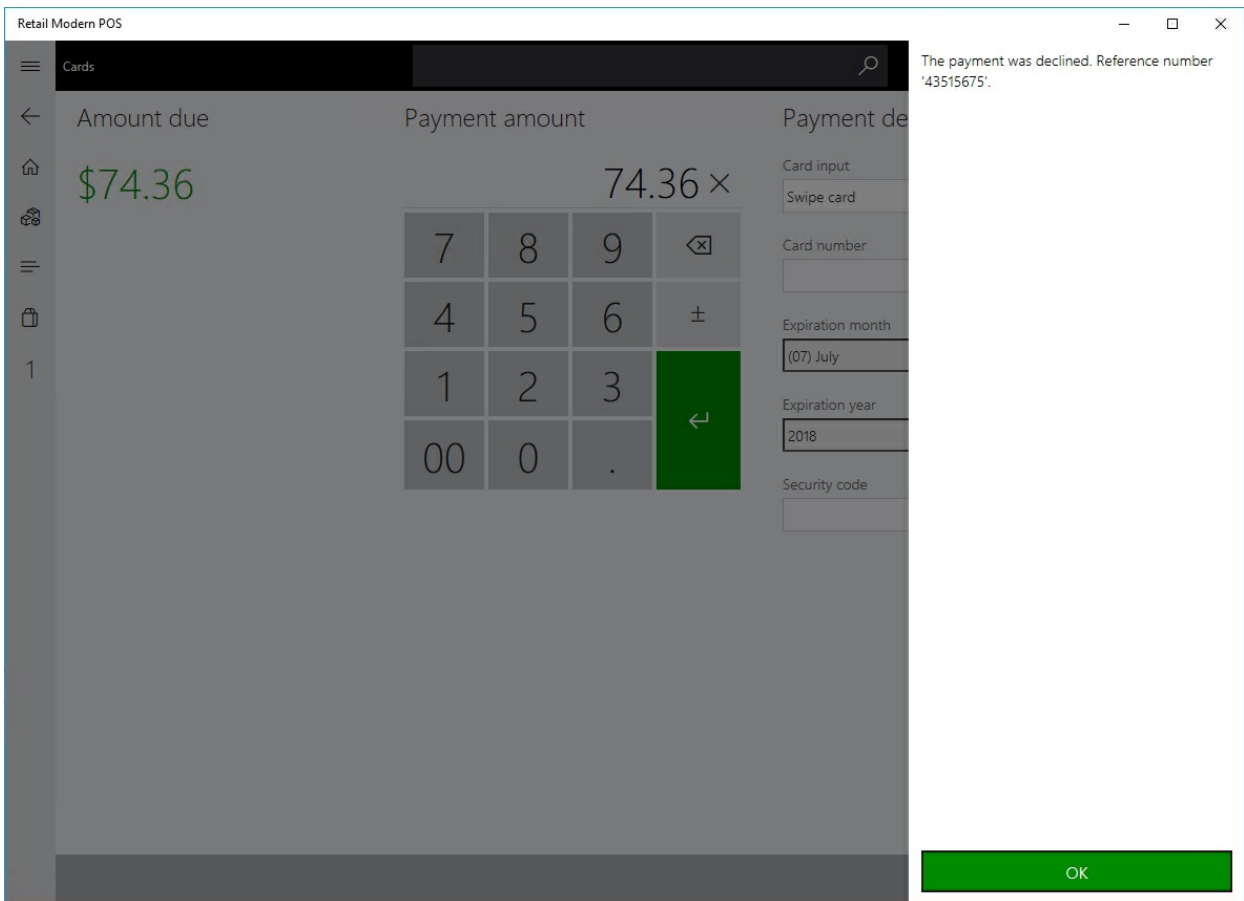
/// <summary>
/// Executes the payment device simulator operation based on the incoming request type.
/// </summary>
/// <param name="request">The payment terminal device simulator request message.</param>
/// <returns>Returns the payment terminal device simulator response.</returns>
public Response Execute(Microsoft.Dynamics.Commerce.Runtime.Messages.Request request)
{
    ThrowIf.Null(request, nameof(request));
    Type requestType = request.GetType();
    Response response;
    if (requestType == typeof(AuthorizePaymentTerminalDeviceRequest))
    {
        response = this.AuthorizePayment((AuthorizePaymentTerminalDeviceRequest)request);
    }
    else if (...)
    {
        ...
    }
    else
    {
        throw new NotSupportedException(string.Format(CultureInfo.InvariantCulture, "Request '{0}'
is not supported.", request));
    }
    return response;
}

/// <summary>
/// Authorize payment.
/// </summary>
/// <param name="request">The authorize payment request.</param>
/// <returns>The authorize payment response.</returns>
public AuthorizePaymentTerminalDeviceResponse AuthorizePayment(AuthorizePaymentTerminalDeviceRequest
request)
{
    ThrowIf.Null(request, "request");
    ...

    // Assuming the external payment terminal/gateway returned a decline and a reference number.
    // Construct the custom error message and set the payment error on the 'paymentInfo' object set
    // on the response.
    PaymentInfo paymentInfo = new PaymentInfo();
    bool isLocalized = true;
    string errorMessage = string.Format("The payment was declined. Reference number '{0}'.",
referenceNumber);
    PaymentError paymentError = new PaymentError(ErrorCode.Decline, errorMessage, isLocalized);
    paymentInfo.Errors = new PaymentError[] { paymentError };
    return new AuthorizePaymentTerminalDeviceResponse(paymentInfo);
}
}
}

```

The following illustration shows how the custom error message appears in the POS.

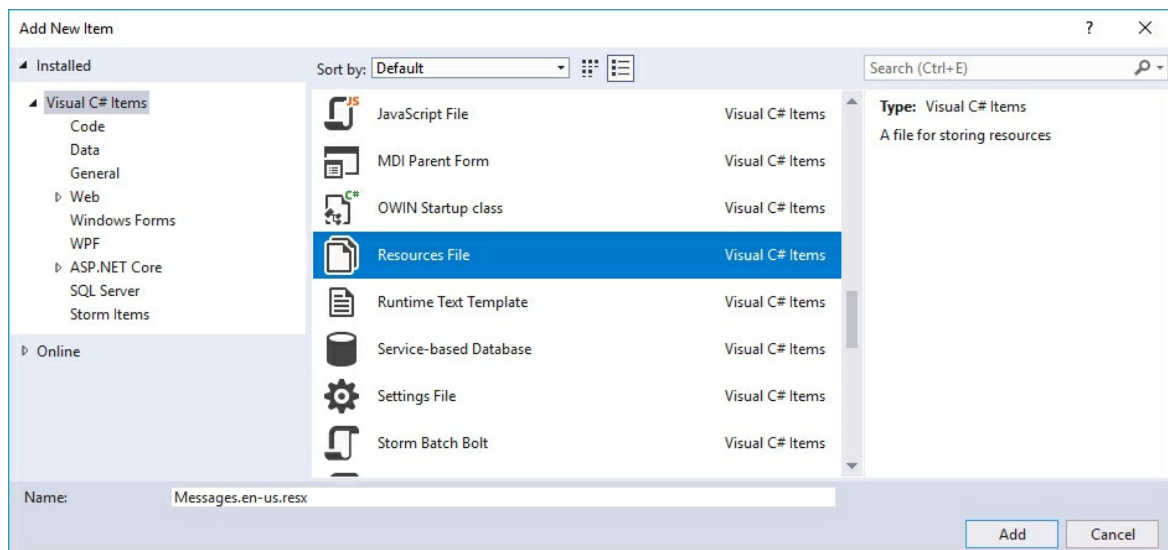


## Create localized error messages

### Create resource files for each locale

To return localized error messages from the payment connector to the POS, you must create localized resource files for each locale that you plan to support. To create a resource file, follow these steps.

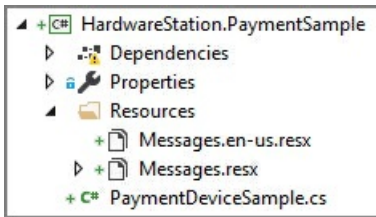
1. In Microsoft Visual Studio, right-click the connector project (or a subfolder, as required), and then select **Add > New Item**.
2. In the new **Add New Item** dialog box, select **Visual C# Items** in the left pane and **Resource File** in the center pane.



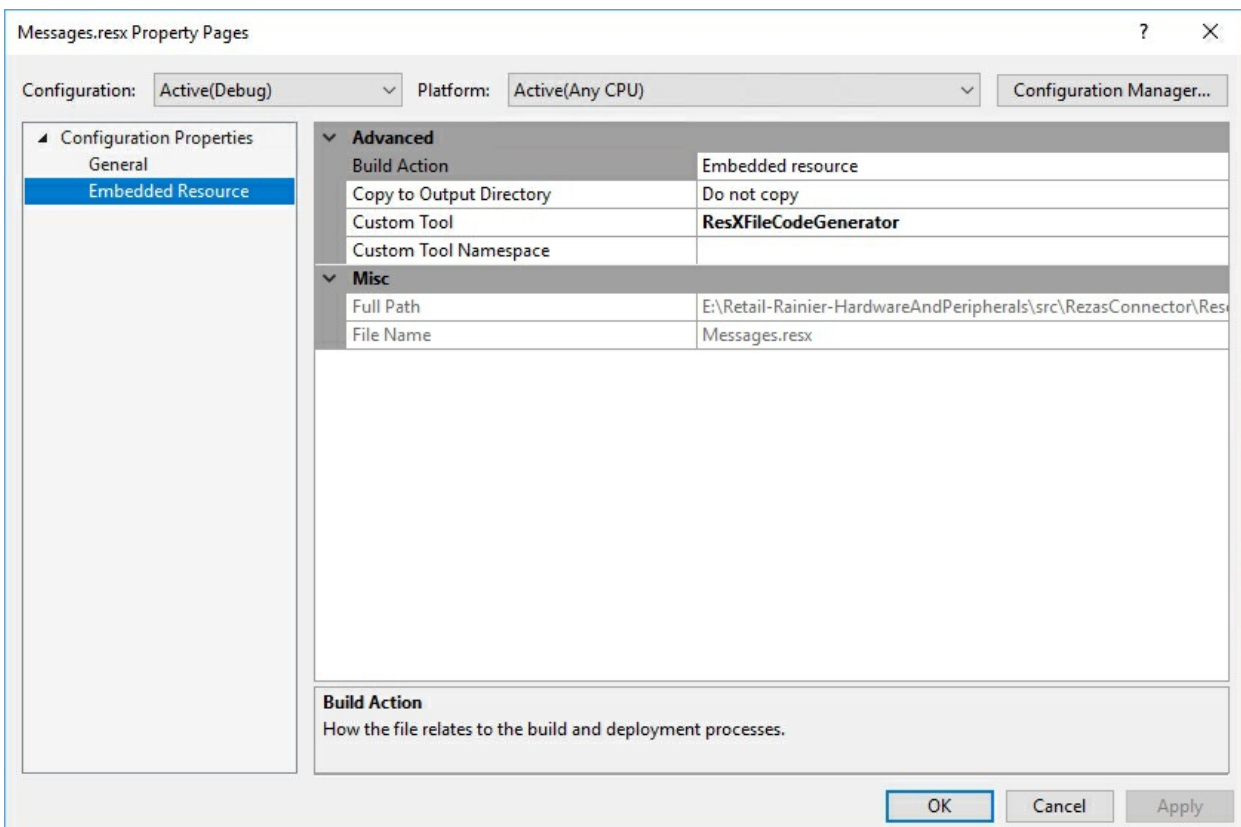
Note that a culture-specific postfix (for example, **en-us**) is required in the file name of every resource file that you create, so that localized satellite assemblies can be generated.

When you've finished, the following resource files should be present in your project. Although the following illustration shows only one extra locale (**en-us**), you can add support for as many locales as you require.

Make sure that a culture-neutral resources file (**Messages.resx** in this example) is defined. This file is used as a fallback if the file for a specific culture is missing.

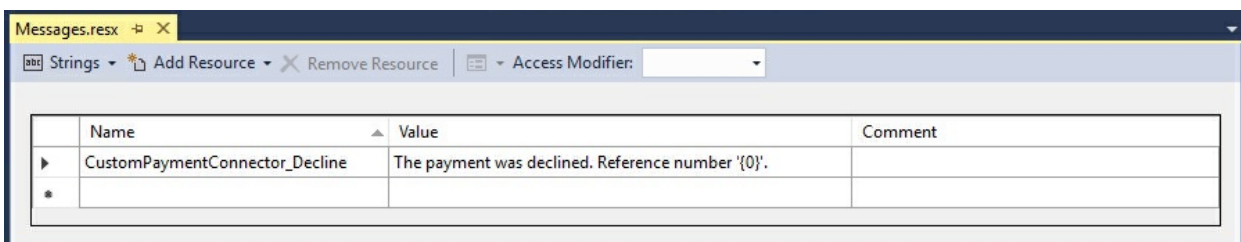


You must also make sure that the correct properties are set for the resource files in Visual Studio, as shown in the following illustration.



### Create custom localized error messages

Every resource file must contain every error message that you want to customize and localize. The following illustration shows an example of a resource file. Notice that the **CustomPaymentConnector\_Decline** entry is referenced in the code to retrieve the appropriate message for a specific locale. Every resource file for every locale should have an identical set of localized messages.



### Load the localized message in the connector code

The following example shows how you can use the resource files that you created earlier in your payment connector code to load a localized message. The process consists of two steps:

1. Make sure that `terminalSettings` is retrieved during the `OpenPaymentTerminalDeviceRequest` request, to access the locale for the request.
2. During the `AuthorizePaymentTerminalDeviceRequest` call (or equivalent calls), use the `Locale` property on `terminalSettings` to retrieve the correct resource file for the localized message.

#### NOTE

The following example has been significantly simplified to show the mechanics of loading localized messages during the runtime of your payment connector code. However, we recommend that you introduce a new set of classes to manage loading of the appropriate resource file.

```
namespace Contoso.Commerce.HardwareStation.PaymentSample
{
    /// <summary>
    /// <c>Simulator</c> manager payment device class.
    /// </summary>
    public class PaymentDeviceSample : INamedRequestHandler
    {
        // Cached version of the terminal settings retrieved during the OpenPaymentTerminalDeviceRequest
        // call.
        private SettingsInfo terminalSettings;

        // Resource manager to retrieve localized messages.
        private ResourceManager messagesResourceManager;

        /// <summary>
        /// Initializes a new instance of the <see cref="PaymentDeviceSample"/> class.
        /// </summary>
        public PaymentDeviceSample()
        {
            this.messagesResourceManager = new
            ResourceManager("Contoso.Commerce.HardwareStation.PaymentSample.PaymentDeviceSample.Resources.Messages",
            typeof(PaymentDeviceSample).GetTypeInfo().Assembly);
        }

        /// <summary>
        /// Gets the collection of supported request types by this handler.
        /// </summary>
        public IEnumerable<Type> SupportedRequestTypes
        {
            get
            {
                return new[]
                {
                    typeof(OpenPaymentTerminalDeviceRequest),
                    typeof(AuthorizePaymentTerminalDeviceRequest),
                    ...
                };
            }
        }

        /// <summary>
        /// Executes the payment device simulator operation based on the incoming request type.
        /// </summary>
        /// <param name="request">The payment terminal device simulator request message.</param>
        /// <returns>Returns the payment terminal device simulator response.</returns>
        public Response Execute(Microsoft.Dynamics.Commerce.Runtime.Messages.Request request)
        {
            ThrowIf.Null(request, nameof(request));
            Type requestType = request.GetType();
            Response response;
            if (requestType == typeof(OpenPaymentTerminalDeviceRequest))
            {
                response = this.Open((OpenPaymentTerminalDeviceRequest)request);
            }
        }
    }
}
```

```

    }
    else if (requestType == typeof(AuthorizePaymentTerminalDeviceRequest))
    {
        response = this.AuthorizePayment((AuthorizePaymentTerminalDeviceRequest)request);
    }
    else if (...)
    {
        ...
    }
    else
    {
        throw new NotSupportedException(string.Format(CultureInfo.InvariantCulture, "Request '{0}'
is not supported.", request));
    }
    return response;
}

/// <summary>
/// Open the payment terminal.
/// </summary>
/// <param name="request">The open request.</param>
/// <returns>The open response.</returns>
private Response Open(OpenPaymentTerminalDeviceRequest request)
{
    this.terminalSettings = request.TerminalSettings;
    ...
}

/// <summary>
/// Authorize payment.
/// </summary>
/// <param name="request">The authorize payment request.</param>
/// <returns>The authorize payment response.</returns>
public AuthorizePaymentTerminalDeviceResponse AuthorizePayment(AuthorizePaymentTerminalDeviceRequest
request)
{
    ...

    // Assuming the external payment terminal/gateway returned a decline and a reference number.
Construct
    // the custom error message and set the payment error on the 'paymentInfo' object set on the
response.
    PaymentInfo paymentInfo = new PaymentInfo();
    CultureInfo cultureInfo = new CultureInfo(this.terminalSettings.Locales);
    string localizedString =
this.messagesResourceManager.GetString("CustomPaymentConnector_Decline", cultureInfo);
    string errorMessage = string.Format(localizedString, referenceNumber);
    bool isLocalized = true;
    PaymentError paymentError = new PaymentError(ErrorCode.Decline, errorMessage, isLocalized);
    paymentInfo.Errors = new PaymentError[] { paymentError };
    return new AuthorizePaymentTerminalDeviceResponse(paymentInfo);
}
}
}
}

```

#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create Windows installers for payment connectors

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes how to create a Windows installer for a payment connector. This topic is targeted at developers working for or with a payment connector provider, for example, MasterCard or Visa, to describe how to package a payment connector, which can then be shared with implementation partners working for specific customers.

After you've implemented and tested your payment connector in the development environment, you must create an installer to transfer the payment connector to a retailer IT professional or a value-added reseller (VAR) for production deployment. For more information, see [Create an end-to-end payment integration for a payment terminal](#).

## Windows Installer

Microsoft Windows Installer (MSI) is the application and configuration service for Microsoft Windows. You must create an installer that contains all the files that are required in order to deploy a payment connector. The installer itself doesn't deploy the payment connector. It just unzips and copies the connector files to a designated folder. The next section of this topic defines the contents of that folder. You can configure the installer to require that customers accept a user agreement before they can continue with the installation. You can also choose your preferred format for the installer. For example, you can have an .exe file.

## Installer contents

The installer must install the required files in the following structure:

### NOTE

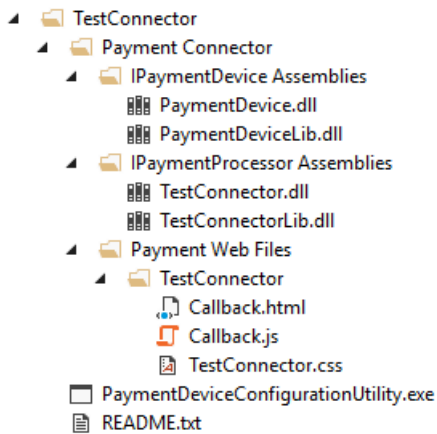
You can find these folders in ...\`RetailSDK\PaymentExternals`.

- **PaymentExternals** – This folder contains up to three subfolders:
  - **IPaymentDevice Assemblies** – This folder contains the assembly that implements the IPaymentDevice interface and its dependent assemblies. These assemblies are used in Hardware station and Retail Modern POS to communicate with payment terminal devices, such as VeriFone MX925. If you don't support a payment terminal device, you can omit this folder.
  - **IPaymentProcessor Assemblies** – This folder contains the assembly that implements the IPaymentProcessor interface and its dependent assemblies.
  - **Payment Web Files** – This folder contains the callback HTML/JavaScript/CSS files that are required in order to enable the payment accepting page. If the payment accepting web app follows the Microsoft implementation guide, these payment web files aren't required, and you can omit this folder. In some cases, the payment accepting page requires that a callback page be deployed to the host application server. In that case, you must include a folder for the callback page:
    - **<Payment\_Connector\_Name>** – To avoid file conflicts from different payment connectors, the payment web files must be placed inside subfolders. We recommend that you use the connector name as the name of the subfolder.
- **Other utility tools** – If you have other utility tools, such as a configuration utility for a payment terminal device, you can include them here. If you don't support any payment terminal devices, you can omit these files. The utility tool should be designed to interact only with the payment terminal devices. It should not

interact with anything in Hardware station or Modern POS. Hardware station and Modern POS will be packaged together with files from the Payment Connector folder.

- **README.txt** – This file describes the contents of the folder and how to use any utilities that are included. The file must also include the payment page URLs that are required in the Cloud POS web.config and Modern POS package.appxmanifest files. If you support any payment terminal devices, the file must also mention the name of the assembly that implements the IPaymentDevice interface. The assembly will be registered in the Hardware Station .config file.

The following illustration shows the file structure for a connector that is named TestConnector.



#### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Dynamics 365 payment data use

2/18/2021 • 13 minutes to read • [Edit Online](#)

This topic provides an overview of the data that is managed by the payment connectors for Microsoft Dynamics 365.

## Key terms

TERM	DESCRIPTION
Card present	The term <i>card present</i> describes the use of a payment connector to process transactions where a physical card is present, such as at a point of sale (POS) register.
Card not present	The term <i>card not present</i> describes the use of a payment connector to process transactions where a physical card isn't present, such as in call center or e-Commerce scenarios.

## Overview

This topic provides specific details about the following areas with respect to data that is managed by the payment connectors:

- [Data used in card-present scenarios](#) – This section provides a list and descriptions of data fields that are passed to the payment connector for card-present scenarios.
- [Data used in card-not-present scenarios](#) – This section provides a list and descriptions of data fields that are passed to the payment connector for card-not-present scenarios.

## Data used in card-present scenarios

This section describes all data points that are sent to the payment connector for card-present scenarios. The payment connect might not use the data.

### Payment connector request-specific data

#### BeginTransactionPaymentTerminalDeviceRequest

FIELD	DESCRIPTION
merchantInformation	The merchant information that is defined on the <b>POS hardware profile</b> page in the Finance and Operations client.
invoiceNumber	The unique invoice number that the POS generates to track the sales transaction.

#### UpdateLineItemsPaymentTerminalDeviceRequest

FIELD	DESCRIPTION
totalAmount	The total amount on the current sales transaction.



FIELD	DESCRIPTION
taxAmount	The tax amount on the current sales transaction.
discountAmount	The discount amount on the current sales transaction.
subTotalAmount	The subtotal amount on the current sales transaction.
items	The list of product-specific details, such as product names, quantities, or units of measure.

#### AuthorizePaymentTerminalDeviceRequest

FIELD	DESCRIPTION
amount	The amount to authorize.
currency	The currency for the amount to authorize.
voiceAuthorization	The voice approval code that is sent from the POS if voice authorization is required.

#### CapturePaymentTerminalDeviceRequest

FIELD	DESCRIPTION
amount	The amount to capture.
currency	The currency for the amount to capture.
paymentPropertiesXml	The content of the <b>PaymentSdkData</b> object that is returned by the <b>AuthorizePaymentTerminalDeviceRequest</b> or <b>RefundPaymentTerminalDeviceRequest</b> request, and that is used to support stateful properties between the requests. For more details, see the <a href="#">Payment SDK data</a> section later in this topic.

#### VoidPaymentTerminalDeviceRequest

FIELD	DESCRIPTION
amount	The amount of the payment to void.
currency	The currency for the payment to void.
paymentPropertiesXml	The content of the <b>PaymentSdkData</b> object that is returned by the <b>AuthorizePaymentTerminalDeviceRequest</b> or <b>RefundPaymentTerminalDeviceRequest</b> request, and that is used to support stateful properties between the requests. For more details, see the <a href="#">Payment SDK data</a> section later in this topic.

#### RefundPaymentTerminalDeviceRequest

FIELD	DESCRIPTION
amount	The amount to refund.
currency	The currency for the amount to refund.

#### ActivateGiftCardPaymentTerminalRequest

FIELD	DESCRIPTION
amount	The initial amount to add to the gift card during activation.
currency	The currency for the initial amount to add to the gift card during activation.

#### AddBalanceToGiftCardPaymentTerminalRequest

FIELD	DESCRIPTION
amount	The amount to add to the gift card balance.
currency	The currency for the amount to add to the gift card balance.

#### GetGiftCardBalancePaymentTerminalRequest

FIELD	DESCRIPTION
currency	The currency to retrieve the gift card balance in.

#### GetPrivateTenderPaymentTerminalDeviceRequest

FIELD	DESCRIPTION
amount	The amount that is set on the POS. (Typically, this variable is used to show the amount on the payment terminal when the card number is retrieved.)

### Shared data

#### Payment SDK data

FIELD	DESCRIPTION
ApprovedAmount	The amount that was approved for the transaction.
AvailableBalance	The available balance on the card.
ApprovalCode	The approval code for the transaction.
ProviderTransactionId	The transaction identifier of the payment provider.
AuthorizationResult	The result of the authorization call.
ExternalReceipt	The external receipt data from the payment provider.

FIELD	DESCRIPTION
TerminalId	The unique identifier of the terminal that handled the payment.

## Data used in card-not-present scenarios

This section describes data that is sent to the payment connector for card-not-present scenarios. The specific data that each connector processes varies, and a given connector might not use all the data that is provided.

### Payment connector method-specific data

#### Authorization

NAMESPACE	FIELD	DESCRIPTION
MerchantAccount	MerchantId	The merchant information that is defined on the <b>POS hardware profile</b> page in the Finance and Operations client.
PaymentCard	Last4Digits	The last four digits of the card that is used for the payment.
PaymentCard	UniqueCardId	The unique randomized identifier of the card that is used for the payment.
PaymentCard	ExpirationYear	The expiration year of the card that is used for the payment.
PaymentCard	ExpirationMonth	The expiration month of the card that is used for the payment.
PaymentCard	StreetAddress	The street of the billing address that is associated with the card that is used for the payment.
PaymentCard	City	The city of the billing address that is associated with the card that is used for the payment.
PaymentCard	State	The state or province of the billing address that is associated with the card that is used for the payment.
PaymentCard	PostalCode	The postal code of the billing address that is associated with the card that is used for the payment.
TransactionData	IndustryType	The type of channel where the payment occurred (for example, <b>Retail</b> , <b>Direct Marketing</b> , or <b>E-Commerce</b> ).
TransactionData	AllowPartialAuthorization	A value that indicates whether partial authorization is supported.
TransactionData	Amount	The total amount of the transaction.

NAMESPACE	FIELD	DESCRIPTION
TransactionData	CurrencyCode	The currency code for the transaction.
TransactionData	TerminalId	The unique identifier of the terminal where the transaction occurred.
PurchaseLevelData	L2Data	The list of "Level 2" data. For more details, see the <a href="#">L2 data</a> section later in this topic.
PurchaseLevelData	L3Data	The list of "Level 3" data. For more details, see the <a href="#">L3 data</a> section later in this topic.

#### Capture

NAMESPACE	FIELD	DESCRIPTION
MerchantAccount	MerchantId	The merchant information that is defined on the <b>POS hardware profile</b> page in the Finance and Operations client.
TransactionData	Amount	The total amount of the transaction.
TransactionData	CurrencyCode	The currency code for the transaction.
PurchaseLevelData	L2Data	The list of "Level 2" data. For more details, see the <a href="#">L2 data</a> section later in this topic.
PurchaseLevelData	L3Data	The list of "Level 3" data. For more details, see the <a href="#">L3 data</a> section later in this topic.

#### Void

NAMESPACE	FIELD	DESCRIPTION
MerchantAccount	MerchantId	The merchant information that is defined on the <b>POS hardware profile</b> page in the Finance and Operations client.
TransactionData	Amount	The total amount of the transaction.
TransactionData	CurrencyCode	The currency code for the transaction.

#### Refund

NAMESPACE	FIELD	DESCRIPTION
MerchantAccount	MerchantId	The merchant information that is defined on the <b>POS hardware profile</b> page in the Finance and Operations client.

NAMESPACE	FIELD	DESCRIPTION
PaymentCard	Last4Digits	The last four digits of the card that is used for the payment.
PaymentCard	UniqueCardId	The unique randomized identifier of the card that is used for the payment.
PaymentCard	ExpirationYear	The expiration year of the card that is used for the payment.
PaymentCard	ExpirationMonth	The expiration month of the card that is used for the payment.
PaymentCard	StreetAddress	The street of the billing address that is associated with the card that is used for the payment.
PaymentCard	City	The city of the billing address that is associated with the card that is used for the payment.
PaymentCard	State	The state or province of the billing address that is associated with the card that is used for the payment.
PaymentCard	PostalCode	The postal code of the billing address that is associated with the card that is used for the payment.
TransactionData	IndustryType	The type of channel where the payment occurred (for example, <b>Retail</b> , <b>Direct Marketing</b> , or <b>E-Commerce</b> ).
TransactionData	AllowPartialAuthorization	A value that indicates whether partial authorization is supported.
TransactionData	Amount	The total amount of the transaction.
TransactionData	CurrencyCode	The currency code for the transaction.
TransactionData	TerminalId	The unique identifier of the terminal where the transaction occurred.
PurchaseLevelData	L2Data	The list of "Level 2" data. For more details, see the <a href="#">L2 data</a> section later in this topic.
PurchaseLevelData	L3Data	The list of "Level 3" data. For more details, see the <a href="#">L3 data</a> section later in this topic.

#### GetPaymentAcceptPoint

NAMESPACE	FIELD	DESCRIPTION
MerchantAccount	MerchantId	The merchant information that is defined on the <b>POS hardware profile</b> page in the Finance and Operations client.
PaymentCard	Name	The name of the cardholder.
PaymentCard	StreetAddress	The street of the billing address that is associated with the card that is used for the payment.
PaymentCard	City	The city of the billing address that is associated with the card that is used for the payment.
PaymentCard	State	The state or province of the billing address that is associated with the card that is used for the payment.
PaymentCard	PostalCode	The postal code of the billing address that is associated with the card that is used for the payment.
PaymentCard	Country	The country or region of the billing address that is associated with the card that is used for the payment.
PaymentCard	ShowSameAsShippingAddress	A value that identifies whether the billing address is the same as the shipping address.
TransactionData	IndustryType	The type of channel where the payment occurred (for example, <b>Retail</b> , <b>Direct Marketing</b> , or <b>E-Commerce</b> ).
TransactionData	AllowPartialAuthorization	A value that indicates whether partial authorization is supported.
TransactionData	CurrencyCode	The currency code for the transaction.
TransactionData	TerminalId	The unique identifier of the terminal where the transaction occurred.

## Shared data

### L2 data

#### NOTE

L2 data is sent to the connector only if this behavior is explicitly configured through the corresponding connector configuration in the Commerce client.

NAMESPACE	FIELD	DESCRIPTION
L2Data	OrderDateTime	The date and time when the order occurred.
L2Data	OrderNumber	The order number that is associated with the order.
L2Data	InvoiceDateTime	The date and time when the order was invoiced.
L2Data	InvoiceNumber	The invoice number for the order.
L2Data	OrderDescription	The description of the order.
L2Data	SummaryCommodityCode	The commodity code that is associated with the product.
L2Data	MerchantContact	The contact information for the merchant.
L2Data	MerchantTaxId	The unique tax identifier of the merchant.
L2Data	MerchantType	The unique merchant identifier that is maintained by the payment processor.
L2Data	PurchaserId	The unique identifier of the purchaser.
L2Data	PurchaserTaxId	The unique tax identifier of the purchaser.
L2Data	ShipToCity	The city of the shipping address.
L2Data	ShipToCounty	The county of the shipping address.
L2Data	ShipToState_ProvinceCode	The state or province code of the shipping address.
L2Data	ShipToPostalCode	The postal code of the shipping address.
L2Data	ShipToCountryCode	The country or region code of the shipping address.
L2Data	ShipFromCity	The city of the address that the order is shipped from.
L2Data	ShipFromCounty	The county of the address that the order is shipped from.
L2Data	ShipFromState_ProvinceCode	The state or province code of the address that the order is shipped from.

NAMESPACE	FIELD	DESCRIPTION
L2Data	ShipFromPostalCode	The postal code of the address that the order is shipped from.
L2Data	ShipFromCountryCode	The country or region code of the address that the order is shipped from.
L2Data	DiscountAmount	The discount amount that is applied to the specific line item part of the order.
L2Data	MiscCharge	The miscellaneous charges that are applied to the specific line item part of the order.
L2Data	DutyAmount	The duty amount that is applied to the specific line item part of the order.
L2Data	FreightAmount	The freight amount that is applied to the specific line item part of the order.
L2Data	HandlingCharge	The handling charge that is applied to the specific line item part of the order.
L2Data	IsTaxable	A value that identifies whether the specific line item part of the order is taxable.
L2Data	TotalTaxAmount	The total tax amount that is applied to the specific line item part of the order.
L2Data	TotalTaxRate	The total tax rate that is applied to the specific line item part of the order.
L2Data	MerchantName	The name of the merchant.
L2Data	MerchantCity	The city of the address of the merchant.
L2Data	MerchantState	The state or province of the address of the merchant.
L2Data	MerchantCounty	The county of the address of the merchant.
L2Data	MerchantCountryCode	The country or region code of the address of the merchant.
L2Data	MerchantZip	The postal code of the address of the merchant.
L2Data	TaxRate	The tax rate that is applied to the specific line item part of the order.
L2Data	TaxAmount	The tax amount that is applied to the specific line item part of the order.



NAMESPACE	FIELD	DESCRIPTION
L2Data	TaxDescription	The description of the taxes that are applied to the specific line item part of the order.
L2Data	TaxTypeIdentifier	The type identifier of the taxes that are applied to the specific line item part of the order.
L2Data	RequesterName	The name of the requester.
L2Data	TotalAmount	The total amount of the specific line item part of the order.
L2Data	PurchaseCardType	The card type of the purchaser.
L2Data	AmexLegacyDescription1	Legacy American Express description field 1.
L2Data	AmexLegacyDescription2	Legacy American Express description field 2.
L2Data	AmexLegacyDescription3	Legacy American Express description field 3.
L2Data	AmexLegacyDescription4	Legacy American Express description field 4.
L2Data	TaxDetails[].TaxRate	The list of individual tax rates that are applied to the specific line item part of the order.
L2Data	TaxDetails[].TaxDescription	The list of individual descriptions of the taxes that are applied to the specific line item part of the order.
L2Data	TaxDetails[].TaxAmount	The list of individual tax amounts that are applied to the specific line item part of the order.
L2Data	TaxDetails[].TaxTypeIdentifier	The list of type identifiers of the taxes that are applied to the specific line item part of the order.
L2Data	MiscellaneousCharges[].ChargeType	The list of charge types that are applied to the specific line item part of the order.
L2Data	MiscellaneousCharges[].ChargeAmount	The list of charge amounts that are applied to the specific line item part of the order.

### L3 data

**NOTE**

L3 data is sent to the connector only if this behavior is explicitly configured through the corresponding connector configuration in the Commerce client.

NAMESPACE	FIELD	DESCRIPTION
L3Data	SequenceNumber	The sequence number of the item for the order.
L3Data	CommodityCode	The commodity code that is associated with the product.
L3Data	ProductCode	The unique code of the product.
L3Data	ProductName	The name of the product.
L3Data	ProductSKU	The stock keeping unit (SKU) of the product.
L3Data	Descriptor	The description of the product.
L3Data	UnitOfMeasure	The unit of measure of the product.
L3Data	UnitPrice	The unit price of the product.
L3Data	Discount	The discount that is applied to the product.
L3Data	DiscountRate	The discount rate that is applied to the product.
L3Data	Quantity	The quantity of the product.
L3Data	MiscCharge	The miscellaneous charge of the product.
L3Data	NetTotal	The net total amount of the product.
L3Data	TaxAmount	The tax amount of the product.
L3Data	TaxRate	The tax rate of the product.
L3Data	TotalAmount	The total amount of the product.
L3Data	CostCenter	The cost center of the product.
L3Data	FreightAmount	The freight amount of the product.
L3Data	HandlingAmount	The handling amount of the product.
L3Data	CarrierTrackingNumber	The carrier tracking number of the product that is being shipped.

NAMESPACE	FIELD	DESCRIPTION
L3Data	MerchantTaxID	The unique tax identifier of the merchant.
L3Data	MerchantCatalogNumber	The catalog number of the merchant.
L3Data	TaxCategoryApplied	The tax category that is applied to the product.
L3Data	PickupAddress	The street of the pickup address.
L3Data	PickupCity	The city of the pickup address.
L3Data	PickupState	The state or province of the pickup address.
L3Data	PickupCounty	The county of the pickup address.
L3Data	PickupZip	The postal code of the pickup address.
L3Data	PickupCountry	The country or region of the pickup address.
L3Data	PickupDateTime	The date and time of the pickup.
L3Data	PickupRecordNumber	The record number of the pickup.
L3Data	CarrierShipmentNumber	The shipment number of the carrier.
L3Data	UNSPSCCode	The United Nations Standard Products and Services Code (UNSPSC).
L2Data	TaxDetails[].TaxRate	The list of individual tax rates that are applied to the specific line item part of the order.
L2Data	TaxDetails[].TaxDescription	The list of individual descriptions of the taxes that are applied to the specific line item part of the order.
L2Data	TaxDetails[].TaxAmount	The list of individual tax amounts that are applied to the specific line item part of the order.
L3Data	TaxDetails[].TaxTypeIdentifier	The list of type identifiers of the taxes that are applied to the specific line item part of the order.
L3Data	MiscellaneousCharges[].ChargeType	The list of charge types that are applied to the specific line item part of the order.

NAMESPACE	FIELD	DESCRIPTION
L3Data	MiscellaneousCharges[].ChargeAmount	The list of charge amounts that are applied to the specific line item part of the order.

## Related topics

- [Create an end-to-end payment integration for a payment terminal](#) – This topic describes how to create a custom payment connector.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Enable duplicate payment protection for payment connector

2/18/2021 • 5 minutes to read • [Edit Online](#)

This topic describes how to enable duplicate payment protection functionality in a payment connector that manages the integration with a payment terminal. A *payment connector* is an extension library that is written to integrate the POS with a payment terminal.

## Overview

- [Required reading](#) - List of topics that you should be read before starting the implementation of the duplicate payment protection functionality in a payment connector.
- [Prerequisites](#) - List of prerequisites to enable duplicate payment protection in a payment connector implementation.
- [Understanding duplicate payment protection flows](#) - Describes the various flows where the duplicate payment protection is invoked in the POS.
- [Implement duplicate payment requests](#) - Describes the various payment-related requests that need to be implemented to support the duplicate payment protection feature.

## Required reading

Be sure to read the following topic before enabling duplicate payment protection for a given payment connector.

- [Create an end-to-end payment integration for a payment terminal](#) - The duplicate payment protection feature builds on the payment integration for a payment terminal described in this topic.

## Prerequisites

The following prerequisites must be met before duplicate payment protection can be enabled for a payment connector implementation.

### **Support for unique transaction scope in payment terminal or payment gateway/processor**

In order to enable support for duplicate payment protection, the corresponding payment terminal or payment gateway/processor must provide support for unique transaction scopes. This support is usually handled through a unique payment reference identifier that can be generated by the payment terminal or payment gateway/processor before the payment is being processed. Without support for this unique identifier, the connector will not be able to uniquely match a previously initiated transaction with a successful payment authorization, which is at the core of the duplicate payment protection feature.

## Understanding duplicate payment protection flows

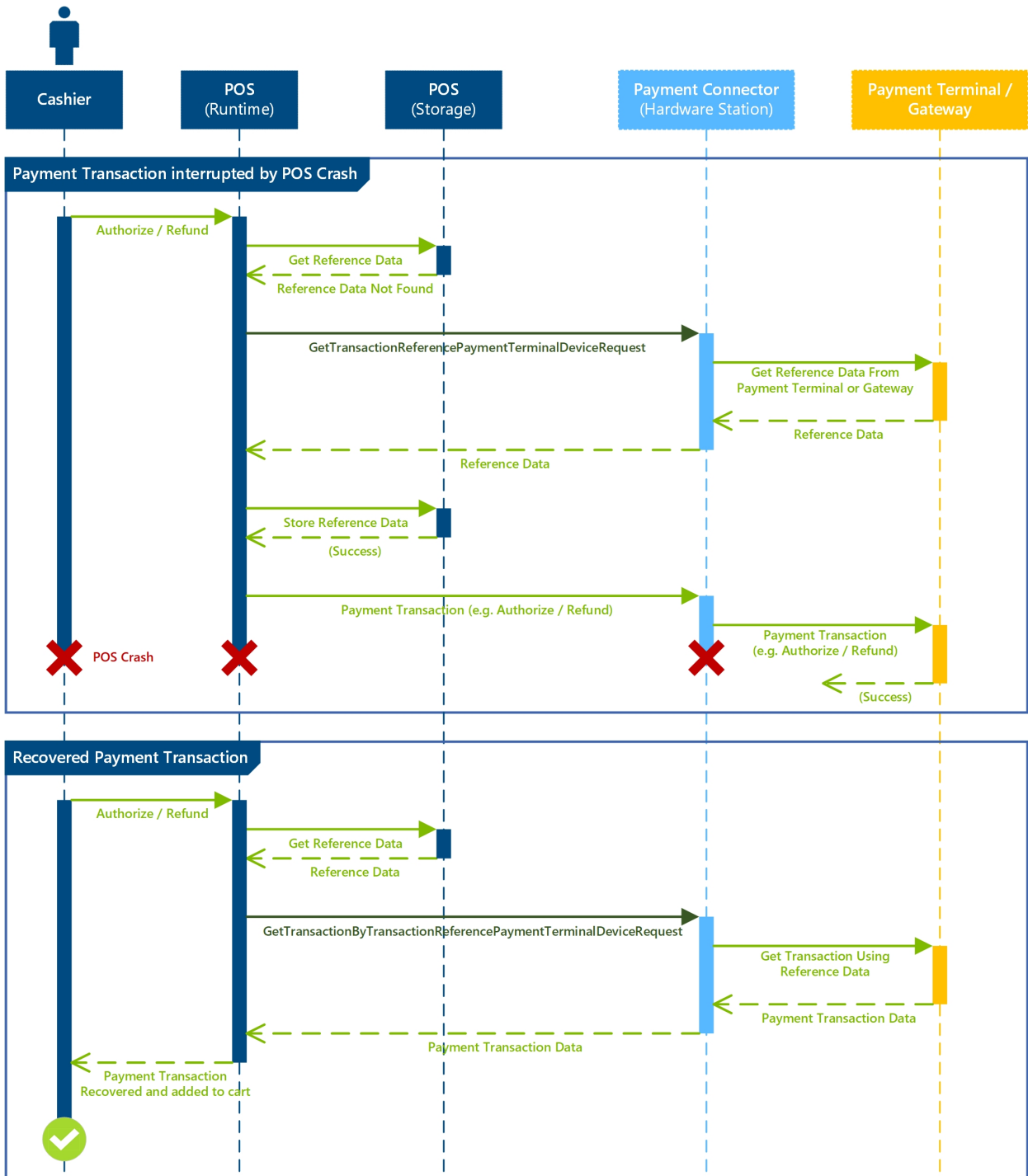
The Retail POS has been extended to invoke the new requests,

`GetTransactionReferencePaymentTerminalDeviceRequest` and

`GetTransactionByTransactionReferencePaymentTerminalDeviceRequest`

in various scenarios across the POS, such as immediately before an **Authorize** request is issued to the payment connector. The purpose of these new requests is to detect and recover successfully processed payment through the payment connector before a new payment request is issued. The following diagram illustrates a simple scenario where a payment request is successfully processed through the payment connector but the POS has crashed before it can receive the response. Subsequently, the POS is able to recover the previously processed payment through the duplicate

payment protection feature.



### Supported POS flows

The following list describes all of the POS flows where the

`GetTransactionByTransactionReferencePaymentTerminalDeviceRequest` is invoked to recover an existing payment.

These are the most commonly executed flows when the POS crashes or loses connectivity to the payment terminal or payment gateway during the processing of a payment transaction.

- Cashier invokes payment for any amount using a card payment
- Cashier invokes payment for any amount using a cash payment
- Cashier attempts to void a line on the cart
- Cashier attempts to void the transaction
- Cashier attempts to suspend the transaction

### Implement duplicate payment requests

The following sample illustrates the requests on the `INamedRequestHandler` payment connector implementation that are required to fully enable the duplicate payment protection feature.

```
namespace Contoso.Commerce.HardwareStation.PaymentSample
{
    /// <summary>
    /// <c>Simulator</c> manager payment device class.
    /// </summary>
    public class PaymentDeviceSample : INamedRequestHandler
    {
        /// <summary>
        /// Gets the collection of supported request types by this handler.
        /// </summary>
        public IEnumerable<Type> SupportedRequestTypes
        {
            get
            {
                return new[]
                {
                    // New request types specific to the duplicate payment protection feature.
                    typeof(GetTransactionReferencePaymentTerminalDeviceRequest),
                    typeof(GetTransactionByTransactionReferencePaymentTerminalDeviceRequest),

                    // Extended with new functionality.
                    typeof(AuthorizePaymentTerminalDeviceRequest)
                };
            }
        }

        /// <summary>
        /// Executes the payment device simulator operation based on the incoming request type.
        /// </summary>
        /// <param name="request">The payment terminal device simulator request message.</param>
        /// <returns>Returns the payment terminal device simulator response.</returns>
        public Response Execute(Microsoft.Dynamics.Commerce.Runtime.Messages.Request request)
        {
            ThrowIf.Null(request, nameof(request));

            Type requestType = request.GetType();

            if (requestType == typeof(GetTransactionReferencePaymentTerminalDeviceRequest))
            {
                return
                this.GetTransactionReference((GetTransactionReferencePaymentTerminalDeviceRequest)request);
            }
            else if (requestType ==
                typeof(GetTransactionByTransactionReferencePaymentTerminalDeviceRequest))
            {
                return
                this.GetTransactionByTransactionReference((GetTransactionByTransactionReferencePaymentTerminalDeviceRequest)
                request);
            }
            else if (requestType == typeof(AuthorizePaymentTerminalDeviceRequest))
            {
                return this.AuthorizePayment((AuthorizePaymentTerminalDeviceRequest)request);
            }
            ...

            return new NullResponse();
        }
    }
}
```

## **GetTransactionReferencePaymentTerminalDeviceRequest / GetTransactionReferencePaymentTerminalDeviceResponse**

### **Description**

The `GetTransactionReferencePaymentTerminalDeviceRequest` is invoked by the Retail POS at the beginning of a payment transaction and sets the scope of the payment. The scope of this request ends once the payment line is successfully added to the cart or an error, indicating that a card cannot be used and returned from the payment connector. The corresponding `GetTransactionReferencePaymentTerminalDeviceResponse` response will contain the corresponding unique identifier generated by the payment connector to look up a payment transaction. Note that the payment connector implementation must not cache the generated ID because the ID must survive the application's restarts, typically the ID should be generated by the Payment Gateway.

#### Request signature

```
public GetTransactionReferencePaymentTerminalDeviceRequest(string lockToken, string posTerminalId, string eftTerminalId);
```

#### Request variables

VARIABLE	DESCRIPTION
lockToken	Unique token value that is generated when the payment terminal is initially locked for the transaction.
posTerminalId	Unique name of the POS register.
eftTerminalId	EFT POS terminal number configured either on the POS register or the shared Hardware Station.

#### Response signature

```
public GetTransactionReferencePaymentTerminalDeviceResponse(string id);
```

#### Response variables

VARIABLE	DESCRIPTION
id	Unique identifier used for the scope of the payment transaction.

#### PaymentTransactionReferenceData

After the `GetTransactionReferencePaymentTerminalDeviceRequest` is executed, the Retail POS will generate a new instance of the `PaymentTransactionReferenceData` class to carry the required contextual data that is needed for the POS to maintain the duplicate payment protection scope. This variable will be stored and maintained in the POS and then used to check for existing transactions during key payment operations.

#### Properties

VARIABLE	DESCRIPTION
Command	Payment related command that was invoked. Possible values are <code>Sale</code> , <code>Refund</code> , <code>Activate</code> , or <code>Load</code> .
IdFromConnector	Unique identifier generated through the <code>GetTransactionReferencePaymentTerminalDeviceRequest</code> request.
InitiatedDate	Date and time when the original payment-related transaction was initiated.



VARIABLE	DESCRIPTION
UniqueTransactionId	Unique identifier for the cart transaction (not specific to a payment itself).
Amount	Amount of the payment transaction being handled.

## AuthorizePaymentTerminalDeviceRequest

### Description

`AuthorizePaymentTerminalDeviceRequest` parameter has now been extended to provide a new constructor that supports the new request, `PaymentTransactionReferenceData`. The purpose of this parameter is to maintain the contextual reference data and stamp the authorization request to the payment terminal or payment gateway/processor with the unique identifier generated by the payment connector and used by the POS to later query for and identify duplicate payments.

## GetTransactionByTransactionReferencePaymentTerminalDeviceRequest / GetTransactionByTransactionReferencePaymentTerminalDeviceResponse

`GetTransactionByTransactionReferencePaymentTerminalDeviceRequest` is invoked by the Retail POS immediately before certain payment-related operations to identify whether an existing payment transaction was already invoked and handled by the payment terminal or payment gateway/processor. If an existing transaction is recovered by the payment connector, it will be returned and short circuit subsequent calls to the connector to trigger a new transaction.

### Request signature

```
public GetTransactionByTransactionReferencePaymentTerminalDeviceRequest(string lockToken,
    PaymentTransactionReferenceData transactionReferenceData);
```

### Request variables

VARIABLE	DESCRIPTION
lockToken	Unique token value that is generated when the payment terminal is initially locked for the transaction.
transactionReferenceData	Property bag containing various properties used to uniquely identify a payment transaction. For more information, see the section <a href="#">PaymentTransactionReferenceData</a> in this topic.

### Response signature

```
public GetTransactionByTransactionReferencePaymentTerminalDeviceResponse(PaymentInfo paymentInfo);
```

### Response variables

VARIABLE	DESCRIPTION
paymentInfo	The recovered payment transaction. This is identical to the payment response returned for any other payment request, such as <b>Authorize</b> or <b>Refund</b> .

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Create Commerce payment packaging for Finance and Operations deployment

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic explains how to package a payment connector for Finance and Operations deployment in Microsoft Dynamics 365 Commerce.

In releases that are earlier than 10.0.10, you use the Commerce software development kit (SDK) to create a payment connector package. (The Commerce SDK was previously known as the Retail SDK.) In the 10.0.10 release and later, you can use only Visual Studio to create an Application Object Server (AOS) payment connector package. Packages that you create by using this approach can be deployed for both earlier deployments and self-service deployments by using [all-in-one packages](#).

## NOTE

In releases that are earlier than 10.0.10, you can create a single payment package and use it both for Application Explorer and for the commerce channel and cloud components (Commerce Scale unit). In the 10.0.10 release, you must create two packages. One package is for Application Explorer, and you create it by using the Dynamics 365 packaging model. The other package is for the commerce channel and cloud components, and you create it by using the Commerce SDK. The previous approach, where the Commerce SDK is used to create Application Explorer payment packaging, is obsolete (deprecated) as of the 10.0.10 release.

To create a payment package that you can deploy, follow the steps in the next section.

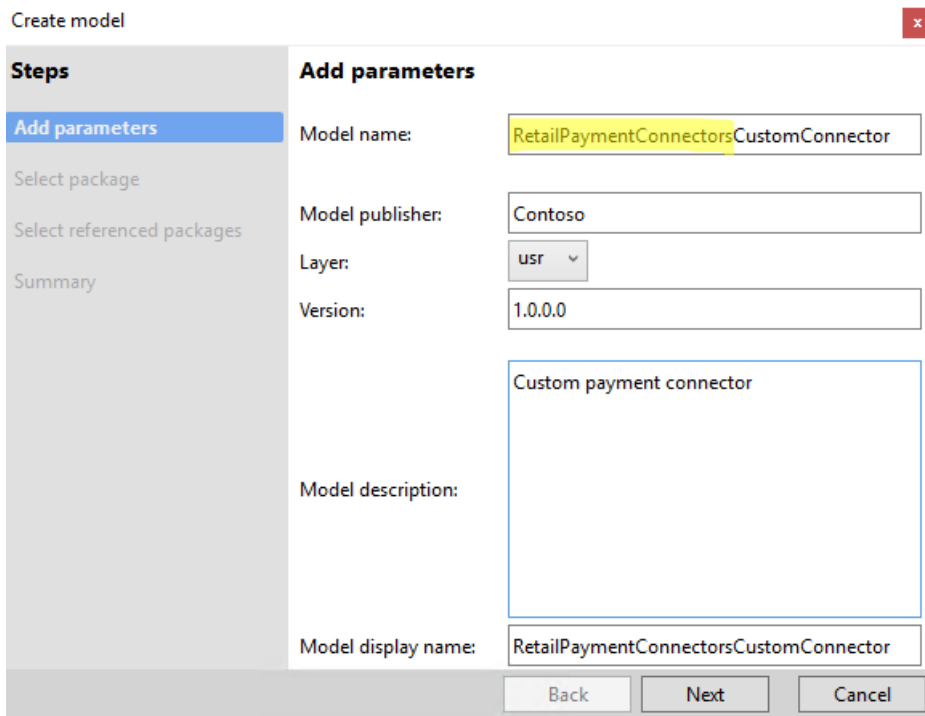
## NOTE

The steps for using the Commerce SDK to create the package for the commerce channel and cloud components haven't changed. For more information, see [Create and deploy connector](#).

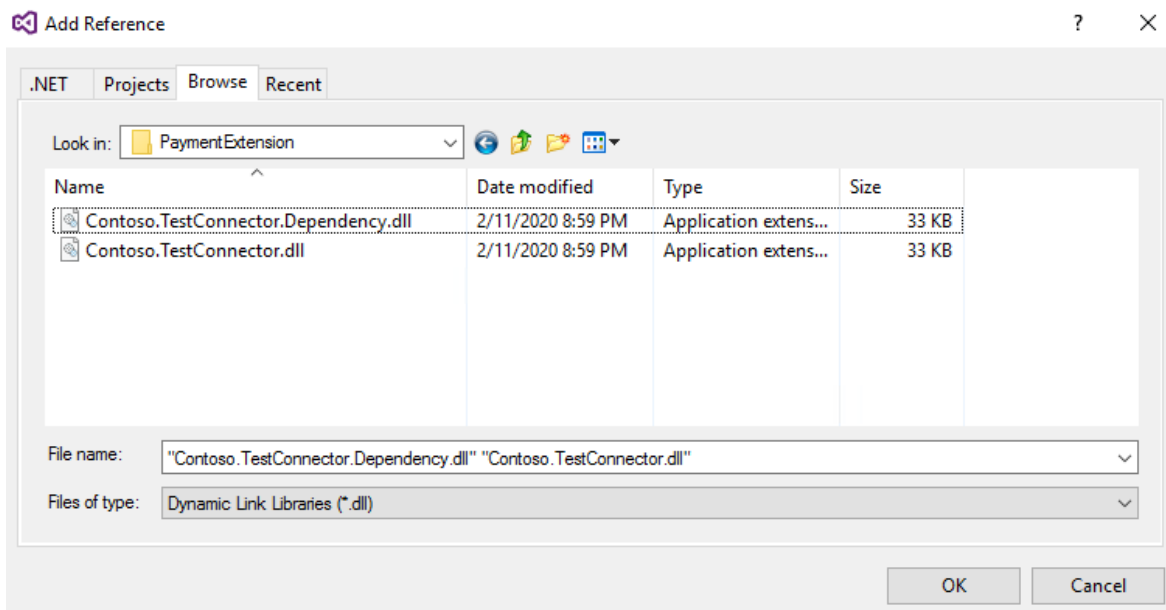
## Create an AOS payment package in the 10.0.10 release

1. In Visual Studio, on the **Dynamics 365** menu, select **Model Management > Create model**.
2. Enter the model name, the model publisher, and other required details. Then select **Next**.

The model name must be prefixed with (that is, start with) **RetailPaymentConnectors**. After this prefix, add information about the custom model name. For example, the model that you create might be named **RetailPaymentConnectorsCustomConnector**. Only model names that begin with the **RetailPaymentConnectors** prefix will be loaded in the Commerce payment connector options.



3. Select the **Create new package** option, and then select **Next**.
4. Select the required referenced package, and then select **Next**.
5. Select **Finish** to finish creating the model.
6. In Solution Explorer, select the project, right-click **References**, and then select **Add Reference**.
7. Add all the payment connector assemblies and their dependencies to the project as references.



8. If your extension needs an HTML and CSS file for the implementation, then add them as a resource file to your project. During deployment, the HTML files will be copied to the AosService\WebRoot\Resources\Html folder. The CSS files will be copied to the AosService\WebRoot\Resources\Styles folder, then accessed with the following URL format.

```
https://AOSUr1/resources/html/Myhtml1.html
https://AOSUr1/resources/styles/Mycss.css
```

#### NOTE

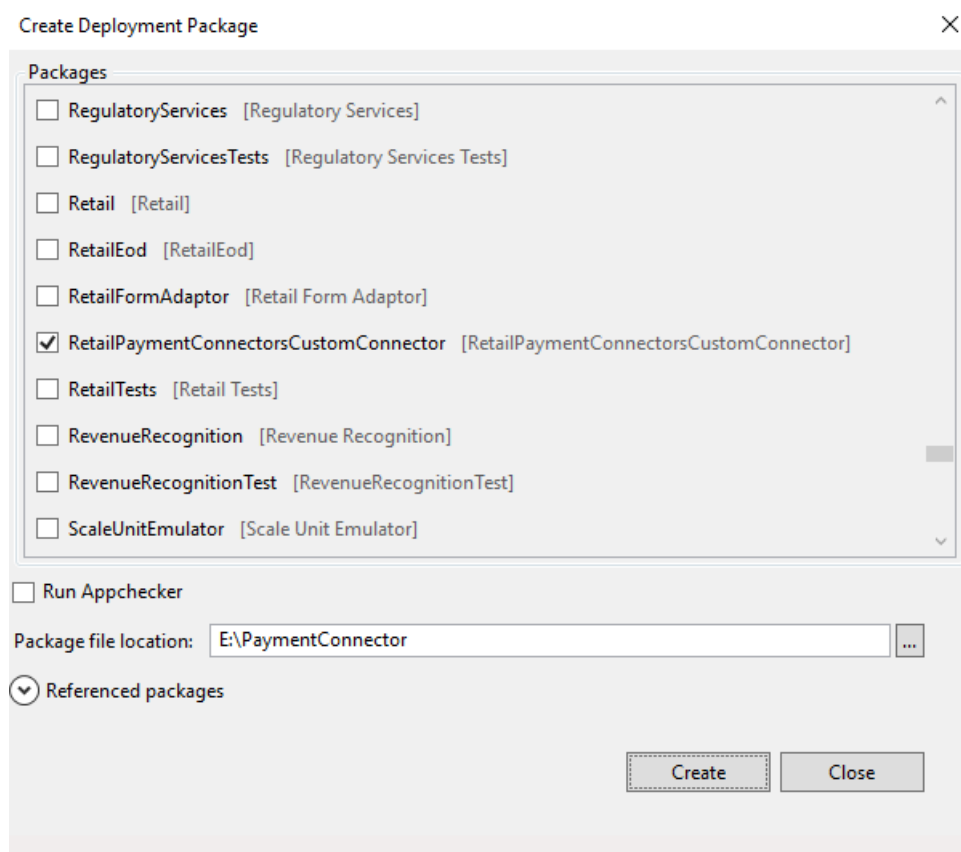
Only HTML and CSS file formats added as Resources to the project will be copied to the AOSService\WebRoot, other file formats added as Resources will not be copied to AOSService\WebRoot. If you need the file in AOSService\WebRoot\ folder then migrate it to HTML file format or host the non-supported file formats externally. IF hosted externally, hosting has to be managed by the customer or partner.

9. If you don't have any other payment X++ extensions that are related to the payment connector, build the solution.

#### NOTE

If there are no other extensions package, then continue with these steps. If you have additional extensions packages, then combine all of them into all-in-one deployable packages. If you do not do this, this package will override other packages. For more information, see [All-in-one deployable packages](#).

10. To create the deployable package, on the Dynamics 365 menu, select **Deploy > Create Deployment Package**.
11. Select the model that you created earlier, specify the location of the package file, and then select **Create**.



Visual Studio builds the model and creates the deployable package.

12. After the deployable package has been created, sign in to Microsoft Dynamics Lifecycle Services (LCS), and then, in your LCS project, select the **Asset Library** tile.
13. Upload the deployable package that you created.

## Apply a deployable package

For information about how to apply a deployable package to an environment, see [Apply updates to cloud](#)

environments.

## Remove a deployable package

For information about how to uninstall or remove a deployable package from an environment, see [Uninstall a package](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Configure online stores

2/18/2021 • 2 minutes to read • [Edit Online](#)

This article provides links to topics that will help you centrally configure and manage an online store.

The topics listed in the following table help you configure Commerce components and the online store in the client.

## Configure an online store

TASK	DETAILS	TOPICS
Configure components.	Set up and maintain information for commerce operations. This information includes stores, taxes, products, gift cards, promotions, and discounts.	<a href="#">Setting up and maintaining Retail</a> (TechNet content for Microsoft Dynamics AX 2012)
Configure a channel navigation hierarchy.	Create a channel navigation category hierarchy to set up a category structure for products that you offer through an online store. You define the category hierarchy, and assign products, product attribute groups, and attribute values to the categories. You then assign the category hierarchy to an online store.	<a href="#">Set up a retail hierarchy</a> (TechNet content for AX 2012) <a href="#">Set up attributes and attribute types</a> (TechNet content for AX 2012) <a href="#">Set up retail attribute groups</a> (TechNet content for AX 2012)
Add the online store to the organization hierarchy.	Before you can assign product assortments or fulfill orders for the online store that you created, or generate reports that include information from that online store, you must assign the store to one or more organization hierarchies. At a minimum, you must assign the online store to an organization hierarchy that includes product assortments.	<a href="#">Set up an online store</a> (TechNet content for AX 2012)
Add modes of delivery to the online store.	Select the delivery methods that the online store offers.	<a href="#">Set up an online store</a> (TechNet content for Microsoft AX 2012)
Map attributes, and add metadata.	Select the options that indicate how the attributes for each category or channel product should behave in the online store on the Microsoft SharePoint site.	<a href="#">Set up an online store</a> (TechNet content for AX 2012)

## Configure online store products

TASK	DETAILS	TOPICS
------	---------	--------

TASK	DETAILS	TOPICS
Add assortments to the online store.	Add the assortments that include the products that you offer in an online store.	<a href="#">Set up an online store</a> (TechNet content for AX 2012)
Manage catalogs.	Use product catalogs to identify the products that you want to offer in your stores.	<a href="#">Key tasks: Create retail product catalogs</a> (TechNet content for AX 2012)
Manage prices.	Set up and use price groups, which are the central link between prices and discounts, and channels, catalogs, affiliations, and loyalty programs.	<a href="#">Setting up prices using price groups</a> (TechNet content for AX 2012) <a href="#">Setting up taxes</a> (TechNet content for AX 2012)
Manage discounts.	Set up and manage price adjustments and four kinds of discounts.	<a href="#">Setting up price adjustments and discounts</a> (TechNet content for AX 2012)
Manage shipping charges.	Set up and manage the shipping charges that are specific to the online store.	<a href="#">Set up shipping charges for online stores</a> (TechNet content for AX 2012)
Manage modes of delivery.	Manage the modes of delivery that the online store offers.	<a href="#">Set up modes of delivery</a> (TechNet content for AX 2012)

## Set up data exchange between Commerce and the online store

TASK	DETAILS	TOPICS
Set up channel integration profiles.	Profiles enable the components to communicate with each other. Set up profiles before you configure data exchange settings.	<a href="#">Set up a Real-time Service profile</a> (TechNet content for AX 2012) <a href="#">Set up a channel profile</a> (TechNet content for AX 2012)

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey.](#)

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

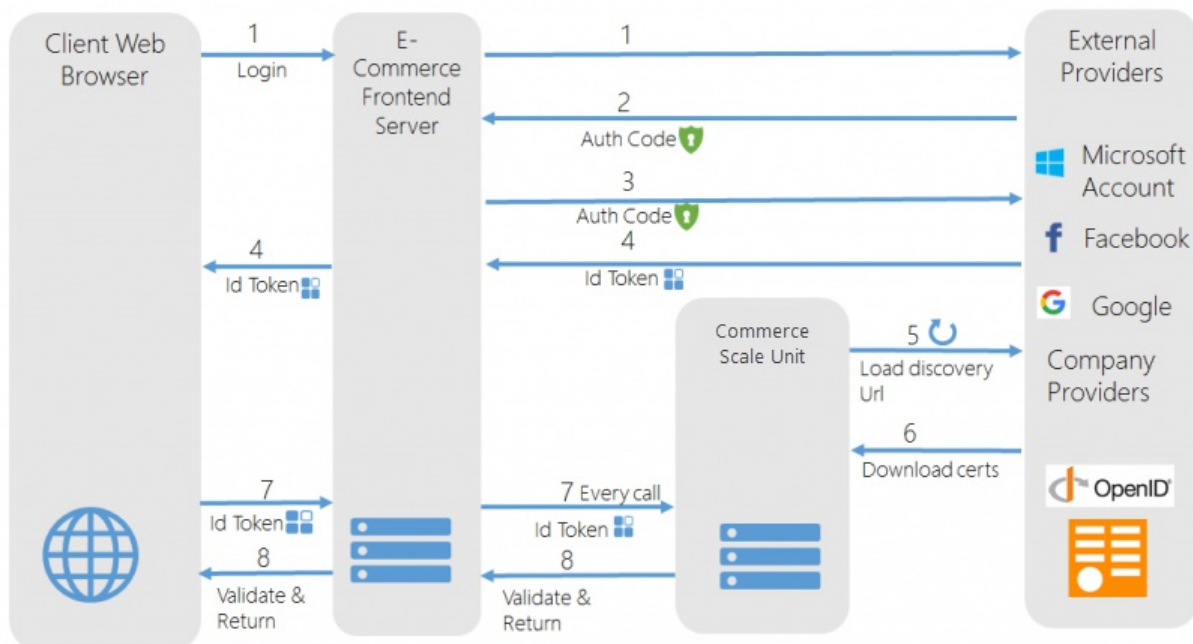


# Configure authentication providers

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides an overview of the process for configuring a new OpenID authentication provider.

The E-Commerce platform uses industry-standard [OpenID Connect](#) as the mechanism for authentication. This article covers the pages that you use to register the OpenID providers that are used in an online store. Commerce Scale Unit uses OpenID Connect as the mechanism to support authenticated customers. OpenID Connect is a universally accepted standard that acts as simple and evolved identity provider on top of OAuth 2.0. Commerce Scale Unit can be integrated with both ready-to-use OpenID providers through the Microsoft Azure Access Control service and other independently available providers. In addition, any custom providers that support OpenID connect can be integrated and registered. The following illustration shows the step-by-step handshake that occurs between the Commerce Scale Unit and the E-Commerce front-end server to pass the authentication token for subsequent calls.



Here is a walkthrough of the process for registering OpenID providers so that they can be used in Commerce Scale Unit.

1. From the Retail and Commerce IT workspace, go to **Commerce shared parameters > OpenID providers**. You can use the **OpenID providers** page to register additional providers. For every provider that you support, enter the details of the OpenID provider and the details of the relying parties. Commerce Scale Unit uses this information to request and use an authentication token for subsequent calls.
2. Run distribution schedule 1110.
3. For the test online store, edit the web.config file so that it specifies the correct redirect URL and domain, as shown in the following example. If you're using a third-party online store, this information can be stored as required.

```
redirectUrl=https://usnconeboxax1ecom.cloud.onebox.dynamics.com/en/Pages/OauthV2Redirect/OauthV2Redirect.aspx
```

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# e-Commerce platform software development kit (SDK)

2/18/2021 • 3 minutes to read • [Edit Online](#)

This topic describes the e-Commerce Platform SDK. The e-Commerce Platform software development kit (SDK) consists of the following components:

- Framework
- Controls
- Publishing (only supported in version 1661 and higher: [Support for Service to Service authentication in Retail Server](#))
- Sample ASP.NET website

## Use the sample ASP.NET website

### Download the Retail SDK

The Retail SDK is available in development environments, and in hotfix packages in a Retail SDK folder.

- If you get the SDK from a development instance, it is immediately ready for configuration and use. For more information, see [Access instances](#).
- If you get the SDK from a hotfix, it is included in the hotfix package as a zipped folder. Hotfixes are cumulative and include all other fixes.

We recommend that you put the SDK in a source control system such as Visual Studio Online.

### Use the Retail SDK to create the sample ASP.NET website

1. Open Visual Studio in Admin mode. This step is necessary for publishing to the inetpub folder.
2. Open C:\Microsoft Dynamics AX70\RetailSdkOnlineStoreOnlineStore.sln contains all the framework components.
3. The sample online store is available at C:\Microsoft Dynamics AX70\RetailSdkSampleExtensionsOnlineStore.
4. Compile and publish the web store front, from within Visual Studio.
5. Update the path of the RetailStorefrontWebSite from IIS Manager.
  - Note that RetailStorefrontWebSite created by the default setup points to C:\Microsoft Dynamics AX70\Retail Store Front\Package.
  - However, the publishing of the web storefront from RetailSDK will drop the files at C:\inetpub\RetailWeb\Storefront.
  - Hence, the physical path of the RetailStorefrontWebSite must be updated to point to "C:\inetpub\RetailWeb\Storefront" to access web storefront on the same ports as before. Another option would be to create a new website and have that point to the inetpub location.
6. Browse to `http://localhost:55080` or access `https://usnconeboxax1ecom.cloud.onebox.dynamics.com/` to see a test ASP.NET website.

### Enable anonymous access

E-Commerce websites must allow anonymous access to work correctly. Ensure that anonymous access is enabled by adding the following key in the Commerce Scale Unit web.config file under app settings.

```
add key="IsAnonymousEnabled" value="true"
```

## Externally accessing the ASP.NET website

The following configuration changes will be required if either of these conditions applies:

- You are accessing web storefront from within a browser that is not on the same box as the e-Commerce server.
- The e-Commerce server and Commerce Scale Unit are on two different boxes.

You will need to update the "retailServerUrl" inside the web.config file of the RetailStorefrontWebSite. The following two fields will need to be updated to use the machine name instead of local host:

```
retailServerUrl=<http://localhost:35080/RetailServer/V1>  
<add key="RetailServerRoot" value="<http://localhost:35080/RetailServer/V1>" />
```

If you are accessing the web storefront over HTTPS, then you will need to update the above URLs to the HTTPS equivalent.

## Operating unit or channel configuration

The e-Commerce website will operate on an operating unit number(channel) specified in the web.config. To change it, change the OU # below. Note that Fabrikam is "077" in the demo data. You will need to update the "retailServerUrl" inside web.config of the RetailStorefrontWebSite. The following two fields will need to be updated to use the machine name instead of local host:

```
<ecommerceControls productUrlFormat="/Pages/ProductDetails/ProductDetails.aspx?itemId={0}"  
retailServerUrl="http://localhost:35080/RetailServer/V1" operatingUnitNumber="068">  
<add key="OperatingUnitNumber" value="068" />
```

# Configure authentication providers

## Authentication providers that you are using

The e-Commerce platform uses OpenID as the mechanism for authentication. You can register any OpenID provider of your choice to the tables below to make this work. You can then log in to test as needed.

1. Edit the web.config file and change it to the following.

```
redirectUrl=https://usnconeboxax1ecom.cloud.onebox.dynamics.com/en/Pages/OauthV2Redirect/OauthV2Redirect.aspx
```

The subsequent steps should only be done to register additional providers.

2. The **Retail Shared Parameters** -> **Open ID Providers** form can be used to register additional providers.
3. Run distribution schedule 1110.

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Localize Commerce extension resources and label files

2/18/2021 • 4 minutes to read • [Edit Online](#)

This topic explains how to modify labels in the point of sale (POS) user interface (UI), POS messages (error, warning, and information), receipt labels, and error messages for Commerce Scale Unit or Commerce Runtime Services (CRT). You can also add custom error messages for in the same way. However, for new POS extension labels, you should use the localization framework in the POS extension.

## POS labels and messages (error, warning, and information)

This section explains how to modify POS UI labels and POS messages by overriding the default strings.

### Override POS UI labels and messages

You can override the default strings in the POS by using the language text entries on the **Language text** page. Follow these steps to change POS strings.

1. Sign in to Commerce.
2. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Language text**.
3. On the **Language text** page, on the **POS** tab, in the **POS language text** grid, select the **Add** button to add the language ID, text ID, and text for the string that you want to override.

For example, you want to change the label of the **Operator ID** field on the POS sign-in page to **Employee ID** for US English (en-us). In this case, add the following entry in the **POS language text** grid.

LANGUAGE ID	TEXT ID	TEXT
en-us	502	Employee ID

#### NOTE

For information about how to get the text ID for POS strings, see the next section.

4. On the Action Pane, select **Save**.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
6. Select the **Registers (1090)** job, and then select **Run now**.
7. After the data is pushed, sign off, and then sign in to Cloud POS or Modern POS to see the changed labels.

### Get the text ID for POS strings

To get the text ID for a POS string, open the Modern POS/Cloud POS application. Press F12 to launch the developer command tools and select the **Console** tab to open the JavaScript console. Run the `Commerce.Helpers.DeveloperModeHelper.setDeveloperMode(true)`; command in the JavaScript console to turn on the developer mode.

After enabling the developer mode in the JavaScript console, navigate to the **Settings** page in POS, under the **Developer mode**, set **Developer Mode** to **Yes**. Set **Show Strings IDs** to **Yes**. Sign out of the POS, and then sign in again. The POS now shows the strings IDs in front of all the labels and messages.

## Error messages or receipt strings

This section explains how to modify error messages, or receipt strings, by overriding the default strings. It also explains how you can add new, custom error messages, or receipt strings.

### Override error messages or receipt strings

1. Sign in to Commerce.
2. Go to **Retail and Commerce > Channel setup > POS setup > POS profiles > Language text**.
3. On the **Language text** page, click the **Add** button to add the language ID, text ID, and text for the string that you want to override.

For example, when users enter an incorrect user name or password during sign-in, the POS shows the following error message: "We didn't recognize the user name or password. Please try again." For US English, you want to change the message to "Please enter valid user name or password." In this case, add the following entry in the **language text** grid.

LANGUAGE ID	TEXT ID	TEXT
en-us	Microsoft_Dynamics_Commerce_Runtime_InvalidAuthenticationCredentials	Please enter valid user name or password

#### NOTE

For information about how to get the text ID for error messages and receipt strings, see the next section.

4. On the Action Pane, select **Save**.
5. Go to **Retail and Commerce > Retail and Commerce IT > Distribution schedule**.
6. Select the **Registers (1090)** job, and then select **Run now**.

### Get the text ID for messages or receipt strings

1. Go to ...\**RetailSDK\Documents\Resources**.
2. In Visual Studio, open one of the following resource files:
  - **To modify error messages:** `RuntimeExceptionMessages.resx`
  - **To modify receipt strings:** `RuntimeReceiptMessages.resx`

For every message in the resource file, Visual Studio shows a name and a value.

3. In the **Value** column, search for the text that you want to change.
4. Copy the name that corresponds to that value. You enter this name as the text ID in the **language text** grid.

### Add custom error messages or receipt strings

You can also add new error messages or new receipt strings, on the **Language text** page. In this way, you support localization instead of hard-coding everything in the code.

### IMPORTANT

The text ID of all your new messages must start with **Microsoft\_Dynamics\_Commerce\_**.

For example, you want to add a new exception message in US English (en-us) and UK English (en-uk). In this case, add entries that resemble the follow entries on the **Language text** page.

LANGUAGE ID	TEXT ID	TEXT
en-us	Microsoft_Dynamics_Commerce_Custo mld1	My new message in US English
en-uk	Microsoft_Dynamics_Commerce_Custo mld1	My new message in UK English

### NOTE

In this example, **CustomId1** is the text ID for the new message. You can use any text ID that you want, provided that it starts with **Microsoft\_Dynamics\_Commerce\_**.

The following example shows how to use this new message in your CRT extension code.

```
throw new CommerceException("Microsoft_Dynamics_Commerce_CustomId1", ExceptionSeverity.Warning, null,
"Custom error")
    {
        LocalizedMessage = "My new message in US English.",
        LocalizedMessageParameters = new object[] { }
    };
```

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Development and administration for Finance and Operations apps

2/18/2021 • 3 minutes to read • [Edit Online](#)

## NOTE

Effective November 2020:

- Common Data Service has been renamed to Microsoft Dataverse. For more information, see [Power Automate Blog](#).
- Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. For more information, see [Terminology updates](#).

This topic will be updated soon to reflect the latest terminology.

Development and administration for Finance and Operations apps includes:

- Administrator experience and Lifecycle Services
- Developer experience
- Intelligence
- Mobile apps
- Data management and data entities
- Office integration

## Developer experience

The developer experience is based on modern tooling using Visual Studio and .NET components.

- The development tools are decoupled from any running environment, which means that you develop against local, XML-based files, not the online database.
- Microsoft Visual Studio is the development environment. Finance and Operations customizes the Visual Studio environment to provide you with a smooth and familiar experience.
- The X++ compiler generates Common Intermediate Language (CIL) for all features. CIL is the same intermediate language used by other .NET-based (managed) languages, such as the C# programming language.
- You can leverage the browser-based client and design patterns for forms to provide an improved end-user experience.
- The Application Lifecycle Management (ALM) system supports build automation, test automation, and deployment of models to the cloud.

For more information, see [Develop and customize home page](#).

## Administrator experience and Lifecycle Services

Finance, Supply Chain Management, and Commerce are cloud-hosted. As an IT professional, you can use Dynamics Lifecycle Services (LCS) to monitor and tune your environments, deploy features, and stay up to date with recent hotfixes. Within your deployment, you can configure security, and manage when processes run. You can also use LCS when you are called on to support business intelligence and reporting, mobile apps, Office, and other integrations.



## BI & reporting

Finance and Operations provides five distinct reporting experiences. Specialized tools are provided to meet the complex and diverse reporting needs of various functions throughout the organization.

- Operational views – Designed to address the specific needs of a given business persona.
- Business documents – Static documents used to capture and exchange processed business data.
- Analytical tools and visualizations – Personalized presentations of logical calculations that allow the user to explore their data.
- Electronic reporting – Tool used to configure formats for electronic documents.
- Financial reporting – Designed to provide in-depth accounting management tools based on standard views of financial activities across legal entities.

## Mobile apps

The Finance and Operations mobile app empowers your organization to mobilize its business processes. After your IT admin enables the mobile workspaces feature for your organization, users can sign in to the app and immediately begin to run business processes from their mobile devices. The Dynamics 365 for Finance and Operations mobile app includes the following features that can help increase productivity:

- Users can view, edit, and act on business data, even if they have intermittent network connectivity or their mobile devices are offline. When a device reestablishes a network connection, offline data operations are automatically synchronized with Finance and Operations.
- IT admins or developers can build and publish mobile workspaces that have been tailored to their organization. The app uses your existing code assets, so you don't have to re-implement your validation procedures, business logic, or security configuration.
- IT admins or developers can easily design mobile workspaces by using the point-and-click workspace designer that is included with the Finance and Operations web client.
- IT admins or developers can optionally optimize the offline capabilities of workspaces by using the Business logic extensibility framework. Because data continues to be processed while a device is offline, your mobile scenarios remain rich and fluid, even if devices don't have constant network connectivity.

## Data management and data entities

Data from Finance and Operations can easily be integrated with Microsoft and non-Microsoft data sources using Dataverse, Power Apps, and Power BI. For more information, see [Data entities overview](#).

## Office integration

The Microsoft Office integration capabilities provide users with a productive environment that helps them get the job done by using Office products. For more information, see [Office integration overview](#).

## eLearning courses

For online courses and training, check out [Dynamics 365 Finance and Operations on Microsoft Learn](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Finance home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides a list of the help topics and other resources for the financial management features in Microsoft Dynamics 365 Finance.

Select a feature area to learn more about it.

- [Accounts payable](#)
- [Accounts receivable](#)
- [Budgeting](#)
- [Cash and bank management](#)
- [Cost accounting](#)
  
- [Expense management](#)
- [Financial reporting](#)
- [Fixed assets](#)
- [General ledger and Financial reporting](#)
- [Project management and accounting](#)
- [Public sector](#)

## Additional resources

### Blogs

- [Microsoft Dynamics 365 blog](#)
- [Financials blog](#)
- [Microsoft Dynamics Operations Partner Community Blog](#)

### Task guides

Additional help is available as task guides inside Finance and Operations. To access task guides, click the Help button on any page.

### Videos

Check out the how-to videos that are now available on the [Microsoft Dynamics 365 YouTube Channel](#).

### Country/region functionality

Country/region regulations affect tax setup and other areas of financial management. Refer to the [Localization and regulatory features](#) section of our help content to learn about country/region-specific functionality.

### Additional content

Supply chain management functionality covers parts of the procure-to-pay process that includes requisitioning, ordering, receiving, invoicing and paying for the goods and services your organization purchases. Refer to the [Supply Chain Management home page](#) for information about the capabilities for managing purchases, inventory, and manufacturing.

**NOTE**

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).

# Supply Chain Management home page

2/18/2021 • 2 minutes to read • [Edit Online](#)

This topic provides a list of the help topics and other resources in Dynamics 365 Supply Chain Management.

## What's new and in development

Go to the [Dynamics 365 Roadmap](#) to see what new features are released and what new features are in development.

## Core concepts and tasks

Select a feature area to learn more about it.

- [Asset management](#)
- [Cost accounting](#)
- [Cost management](#)
- [Inventory management](#)
- [IoT Intelligence](#)
- [Master planning](#)
- [Procurement and sourcing](#)
- [Product information management](#)
- [Production control](#)
- [Sales and marketing](#)
- [Service management](#)
- [Transportation management](#)
- [Warehouse management](#)

## Dynamics 365 Finance

For information on Dynamics 365 Finance, go to the [Finance home page](#).

## Videos

This short video summarize the new supply chain management features added to Microsoft Dynamics 365 for Finance and Operations version 8.0 (April 2018).

- [Synchronize a work order between Field Service and Finance and Operations](#)

These short videos summarize the new supply chain management features added to Microsoft Dynamics 365 for Finance and Operations, Enterprise edition 7.3 (December 2017).

- [Prospect to cash integration](#)
- [Optimization advisor](#)
- [Use warehouse template to copy configuration](#)

These short videos summarize the new supply chain management features added to Microsoft Dynamics 365 for Finance and Operations, Enterprise edition (July 2017).

- [Get started with Cost accounting](#)

- [Cost control mobile workspace](#)
- [Use Excel for cost analysis](#)
- [Approve purchase orders on a mobile device](#)
- [Visual scheduling with Gantt chart for production and batch orders](#)

The following tech conference recordings discuss supply chain management functionality from previous versions of Finance and Operations. This functionality is now part of Dynamics 365 Supply Chain Management; the same concepts still apply, and the procedures are similar in the current version.

- **Cost management:**
  - [Overview of Cost management](#)
- **Master planning:**
  - [Extend the demand forecasting functionality](#)
  - [Master planning - tips and tricks for troubleshooting performance](#)
  - [Help! MRP is slow!](#)
- **Product information management:**
  - [Product configurator in Microsoft Dynamics AX](#)
- **Warehouse management:**
  - [Get the best out of your warehouse management system](#)
  - [Dynamics AX 2012 R3: Advanced warehouse management - A day in the life of process manufacturing](#)
- **Production control videos:**
  - [Subcontracting operations and activities in manufacturing](#)
- **Transportation management videos:**
  - [Transportation management \(TMS\) in the new Microsoft Dynamics AX](#)

## Blogs

There are many topics about manufacturing and supply chain management on the [Dynamics AX Manufacturing R&D Team Blog](#) and [Supply Chain Management in Dynamics AX R&D Team Blog](#). Most of these were written for the previous version, but the same concepts still apply, and the procedures are similar in the current version.

## White papers

- [Lean manufacturing: Capable to promise and kanban job scheduling](#)
- [BOM calculation by using a costing sheet](#)

## eLearning courses

For online courses and training, check out [Dynamics 365 Supply Chain Management on Microsoft Learn](#).

### NOTE

Can you tell us about your documentation language preferences? [Take a short survey](#).

The survey will take about seven minutes. No personal data is collected ([privacy statement](#)).